# Predicting Retrieval Failures in Conversational Recommendation Systems

Maria Vlachou

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow


September 2024

# Abstract

In recent years, the use of dialogue systems and voice assistants commonly implemented in smart devices have shifted the users' interest towards online shopping. In turn, online shopping platforms are gaining popularity and move towards allowing an interactive dialogue with users that more accurately depicts a real shopping setting. In this regard, the task of Conversational Image Recommendation is the state-of-the-art task for conversational recommendation in the fashion domain, where a user has a specific fashion item in mind, and interacts with the system with natural language feedback on recommended image items, which guides the system in finding the imagined item in the next turn. Such systems are trained and evaluated with user simulators as a plentiful surrogate for human users. A practical problem with CRS performance is that it is primarily evaluated in terms of successes and is therefore assumed to return the item of interest by a pre-defined number of turns. In practice, often the item is not returned by the end of a conversation, therefore leading to conversational failures; this is our particular setting of interest.

In this thesis, we argue that the performance of a Conversational Recommendation System can be predicted to detect when a conversation fails, under different scenarios, across different turns of a conversation. In this regard, Query Performance Prediction (QPP) techniques predict the effectiveness of a ranked list result in response to a query without having access to relevance judgments. We predict the performance of CRS models by treating them dense retrieval processes, where both image retrieved items and textual feedback can be represented with dense embedded representations. In particular, we propose a set of coherence-based dense QPPs specifically designed for single-representation dense retrieval models (ANCE and TCT-ColBERT) and show that the examination of the relations among dense embedded representations already contained in the document list is sufficient to provide effective predictions for dense retrieval models. At the same time, by using a multi-level perspective that jointly considers QPPs and types of queries, we explain why some QPPs are better for certain types of queries, thus explaining discrepancies among different evaluation metrics.

At the next stage, we predict the effectiveness of a ranking of image items in Conversational Image Recommendation models, which are also based on learned embedded representations of images, and where user feedback takes the place of a textual query. Indeed, we create a novel task which we call Conversational Performance Prediction (CPP), which predicts conversation

success at the conversation level and taking into account the multi-turn nature of the task, and can differentiate between success predicted over a short-term and a long-term horizon, thereby predicting current user satisfaction or overall satisfaction of a conversation. First, we examine the set of unsupervised predictors developed for dense retrieval models but applied to state-of-the-art Conversational Image Recommendation models; a GRU-based model, which mainly considers the feedback of the previous turn, and an EGE model that considers the entire dialogue history. Our results show that using correlations is not an optimal evaluation strategy for predicting conversational failures, as, while correlations are low to medium mainly for short-term predictions, a lot of inconsistencies are observed among the performance of different predictors across metrics and datasets (similarly to dense retrieval models). Consequently, we propose a supervised CPP approach, which treats CPP as a binary classification task, which predicts whether a target item is returned by a give turn. In this way, we show that by learning the embedded representations already contained in the CRS models, we can predict the accuracy of a conversation success using the retrieved items of both single and multiple turns.

In addition, state-of-the-art CRS models are trained using user simulators with a single target item in mind, and at the same time, they are assumed to be infinitely patient. These settings do not reflect a real shopping scenario, where a user might change their mind according to what a shopping assistant is suggesting. For this purpose, we enhance the evaluation completeness of CRS models by obtaining real user opinions in a user study using pooling similar to information retrieval tasks, thus identifying alternatives relevance labels for a number of target items, and in turn, inform the user simulator with an extended target space. This increases the completeness of CRS evaluation, and therefore, creates a more realistic prediction setting for CRS, which leads to improved predictions of user preferences. Indeed, when we reevaluate the CRS models using the updated simulator with the identified alternatives as part of the target space, we show that by the single target setting previously used to evaluate CRS models for a maximum amount of 10 turns was underestimating the effectiveness of CRS models.

As a final step, we account for the fact that CRS models assume only one type of recommendation failure, namely the inability of the system to retrieve the target item. In this regard, we introduce the concept of recommendation scenarios, and specifically, we adapt our CPP framework for different types of conversational failures, which are determined by whether the user need is clearly defined and whether the target item is available. Therefore, we propose the removed target scenario (the target is not available in the catalogue), and the alternative scenario (a user has a more flexible need, which can be satisfied by either the original target or any of the identified alternatives in the collected datasets). Consequently, we detect different types of conversational failure, such as when a user cannot find an item, versus when the system's catalogue does not contain the relevant item. By examining the supervised CPP predictors introduced under these two novel scenarios, we find that in both cases, there is a marked difference from the original scenario, and that CPP can indeed be predicted for different recommendation scenarios.

# Contents

# List of Tables

# List of Figures

# Acknowledgements

Time flies, and this bring us to the end of my PhD. My personal journey in the academic world has been full of adventures and against the odds, and I won't complain about it. Instead, I take the time to thank everyone who has contribute in their own way to the completion of this project. First and foremost, I would like to thank my family for the continuous support they have provided. In particular, I would like to thank my parents, Aikaterini Vlachou (Karakonstanti) and Panagiotis Vlachos, for everything they have offered all these years, for setting an example of a nice person, for sacrificing everything for me, and for teaching me that love is acceptance, acceptance is freedom, and freedom is the only way that helps you to pursue your dreams. I have been able to reach the PhD stage because of them and their unique level of support. I, therefore, dedicate this thesis to them; after all, it is theirs, too. I would also like to thank my dear aunt, Varvara Karakonstanti, for her love and dedication all these years, for being the person I can reach anytime, and for inspiring me to be independent. Her contribution to my academic journey cannot be described in words.

Next, I would like to thank everyone that has contributed to this achievement from the University of Glasgow. In particular, during my time here, I was a member of two families: The Terrier Team and the Social AI CDT. As for the first, I would like to thank my supervisor Prof. Craig Macdonald for his valuable guidance and support throughout my research. Coming to the Terrier Team with a background outside of information retrieval, I was lucky enough to benefit from his expertise, which has shaped my learning curve towards a positive direction. Indeed, overtime, our relationship has grown, and I can finally understand his patience, and also why he placed so much importance on evaluation; in fact, he is responsible for the fact that I constantly think about evaluation, too. During our interaction, I have also adopted a number of research practices that I find really helpful.

Furthermore, I would lie to express my gratitude to everyone in the Social AI CDT for allowing me to join the team and live the PhD experience. I am deeply grateful to my director, Prof. Alessandro Vinciarelli, for his consistent support throughout these four years, for being extremely generous and for contributing to my overall well-being. I will never forget his unique guidance in the PhD process, the advice he has provided, and his patience with me. In addition, I would like to thank my director Prof. Monika Harvey for the support she has provided to all of us, for her generosity, and for her positive energy towards the entire cohort. I would also like

to thank our third director Prof. Stacy Marcella for the amazing course he taught us in the first year of the CDT.

I would also like to thank all of my colleagues from the Terrier Team for the time we have spent and the experiences we've had. My gratitude extends to Iadh Ounis, Debasis Ganguly, Sean Macavaney and Jake Lever for providing yearly feedback on my project, Richard McCreadie, Graham McDonald, Zaiqiao Meng, Yaxiong Wu, Xiao Wang, Zixuan Yi, Edward Richards, Jack Mckechnie, Sarawoot Kongyoung, Javier Sanz-Cruzado Puig, Thomas Jänich, Andreas Chari, Andrew Parry, Lubingzhi Guo, and many more.

Apart from the Terrier Team, I am extremely grateful to my fellow students in the Social AI CDT. Of course, I remember the amazing time I have spent with the students of my cohort with who I shared the first-year courses, and the unique experiences we have shared during the pandemic. I thank each one of them: Jacqueline Borgstedt, Christopher Chandler, Radu Chirila, Robin Bretin, Andreas Drakopoulos, Thomas Goodge, Casper Hyllested, Gordon Rennie, Sean Westwood, Morgan Bailey, Tobias Thejll-Madsen, and Serera Dimitri. Also, I am thankful to students from other cohorts of the CDT, with who I have shared valuable discussions and moments of support, such as Shaul Ashkenazi, Juliane Kloidt, Rawan Zreik-Srour, Amelie Voges, and many more. Finally, I would like to thank the CDT staff, and in particular Jared de Bruin for his constant support and valuable advice, Monika Maitles, Alison Purdie-Gore, Saskia Sieprath and others.

Since the PhD is a journey of mental strength, I cannot ignore the enormous amount of support and true friendship I have received from my best friends, Ritsa Voulgari and Paraskevi Kaliamoutou. I thank them for being patient when I asked them to leave because I had to work on a paper, for accepting my life choices, and for providing advice when things went wrong. Also, I thank the people who have come to my life in the past year that remind me that I need to keep fighting for my dreams.

Furthermore, I would like to thank all my ballet teachers for contributing to my persistence and attention to detail, to the fact that I am not easily satisfied with the work I produce, and to viewing every research paper as a work of art. I thank them for having shaped me in a way that proves to be helpful for a path in research. Finally, I would like to thank my guardian angel, for always saving me when I stop feeling safe.

# Declaration

With the exception of Chapters 1 and 2, which contain introductory material, all work in this thesis was carried out by the author unless otherwise explicitly stated.

# Chapter 1

# Introduction

In recent years, the use of dialogue systems or voice assistants, such as Amazon Alexa, Apple Siri or Google Assistant (Argal et al., 2018; Brill et al., 2019; Dalton et al., 2018) (often implemented on smart devices) has become more prevalent information search. At the same time, online shopping platforms are becoming increasingly popular and move away from simply displaying items with textual descriptions, thus allowing an interaction with users that mimics a real shopping setting. For example, a user might have a specific fashion item in mind, such as a shoe, a dress, or a shirt, and interacts with a dialogue system in order to find it. The process runs as follows: the system starts by providing a random first recommendation by displaying an image; based on this recommendation, the user provides natural language feedback, which aims at guiding the system in finding the imagined item more easily in the next turn recommendation (Guo et al., 2018; Wu et al., 2021a,b). This process continues until the user's desired item is reached. However, in practice, some users often may not find their items of interest even after interacting with the system for many turns. In this regard, there are a number of challenges that compose the overall problem of *retrieval failure* by a conversational system: (i) there is no mechanism that predicts when a conversation would fail, and which factors contribute to successfully retrieving the correct item, and (ii) there is no distinction between the different reasons why an item might not be returned, for example whether the system is ineffective or because the item is unavailable or simply because the system cannot incorporate the particular aspects of a user need (for example, take into account what else they would prefer instead).

Indeed, we are moving from the more traditional ad-hoc information retrieval paradigm (a search system returning a list of documents from a collection in response to the user's information need (Manning, 2009)) towards a range of *Conversational Information Seeking (CIS)* tasks, which refer to a system that helps users to satisfy their information needs by engaging in an interactive conversation (Zamani et al., 2023). In particular, Conversational Recommendation Systems (CRS) (Christakopoulou et al., 2016; Li et al., 2018; Sun and Zhang, 2018; Zhang et al., 2018; Zou and Kanoulas, 2019) assist users with finding items and with decision making (Sun and Zhang, 2018; Zou and Kanoulas, 2019) by engaging in a dialogue with users. CRS differ

both from (a) traditional recommender systems and (b) conversational search. In particular, with respect to (a), earlier forms of *Recommender Systems (RS)* assist users in finding items of interest in cases of information overload (Ricci et al., 2015) and aid in user exploration (Chen, 2021), but mainly on the basis of user feedback in terms of ratings, clicks or reviews. On the other hand, CRS help with dynamic preference elicitation (Christakopoulou et al., 2016; Li et al., 2018; Sun and Zhang, 2018; Zhang et al., 2018), as they allow users to provide natural-language feedback in the form of *critiquing* (Tou et al., 1982). Critiquing allows users to refine the recommendations in an iterative way towards the user's desired item(s) in each interaction turn by updating the model of the users' preferences according to the user feedback in the previous turn(s) (Chen and Pu, 2012; Yuan and Lam, 2021). Therefore, critiquing-based CRS increase the effectiveness of recommendations compared to traditional RS (Chen and Pu, 2007, 2012; McCarthy et al., 2004). With respect to (b), CRS also differ from *Conversational Search (CS)*, which refers to interactively searching for information with a conversational system using natural language conversations (Zamani et al., 2023). Still, while both CS and CRS are ranking tasks using natural language conversations, CRS involves understanding users' preferences and providing suggestions, which requires more complex mechanisms such as keeping track of the user feedback and system actions (Jannach et al., 2021).

In general, CRS assist users with achieving a number of goals (Jannach et al., 2021). Relevant to the above example from the fashion domain, Guo et al. (2018) introduced *Conversational Image Recommendation*, a multi-turn task that aims to help users with online shopping and returns *candidate* image items at every interaction turn until the user's *target* item is reached. The first CRS model used for this task was based on a Gated Recurrent Unit (GRU) mechanism (Cho, 2014), which mainly considers which mainly considers how the representation of the dialogue state should be updated at each turn. More recently, the task was enriched by also considering the entire dialogue history (Wu et al., 2021b) or displaying a list of candidates (Yu et al., 2019). In all cases, training and evaluation is done by simulating real users with a user simulator that uses natural language sentences describing the relative differences of the target to the candidate image item. While the effectiveness of CRS is widely studied, much less attention has been shown to the case of retrieval failures, i.e., when the system does not return the target item. However, predicting retrieval failure in CRS is important, as it would lead to a timely identification of the lack of understanding of user needs and would indicate the factors that contribute to a more effective future CRS performance at the various stages of a dialogue. In this case, both user feedback and the recommended result list determine the quality of recommendation. To predict the signs of conversational failure, we need to examine which factors impact system performance.

In this regard, the task of *Query Performance Prediction (QPP)* (Carmel and Yom-Tov, 2010; Cronen-Townsend et al., 2002), originally used for ad-hoc retrieval in search engines, predicts the effectiveness of a ranked list result in response to a query without having access to *relevance*

*judgments*. Indeed, thsi is the paradigm we follow in this thesis. For this purpose, the per query value of *query performance predictors (QPPs)* is correlated with the corresponding value of a given ranking effectiveness evaluation metric. QPPs can be either pre-retrieval (examine characteristics of the queries or the corpus before retrieval) (Hauff et al., 2008; He and Ounis, 2006) or post-retrieval, which examine the content of the retrieved document list (Arabzadeh et al., 2021a,b; Cronen-Townsend et al., 2002; Datta et al., 2022b; Roitman et al., 2017b; Shtok et al., 2009, 2010, 2016; Zamani et al., 2018). Since they consider the retrieved documents, post-retrieval predictors are considered more accurate than pre-retrieval ones (Hauff et al., 2008). Over two decades, a variety of post-retrieval unsupervised predictors have examined either the scores distribution (Roitman et al., 2017b; Shtok et al., 2009) or the coherence of the sparse representations (Arabzadeh et al., 2021a; Diaz, 2007) of the document list. More recently, supervised predictors (Arabzadeh et al., 2021b; Datta et al., 2022b; Hashemi et al., 2019) have employed the fine-tuning of BERT-based (Devlin et al., 2019) pre-trained multi-vector representations to predict mainly sparse model rankings, such as BM25 (Robertson et al., 1995). Still, none of these predictors were examined with respect to their performance on a multi-turn and multi-modal ranking task such as Conversational Image Recommendation. In our view, QPP is a promising approach to predict conversational failures if we properly account for the task difficulties.

## 1.1 Motivation

As mentioned, CRS performance is primarily evaluated in terms of successes (returning an item of interest by a pre-defined number of interaction turns) (Christakopoulou et al., 2016; Jannach et al., 2021; Ren et al., 2022; Zangerle and Bauer, 2022; Zou and Kanoulas, 2019; Zou et al., 2020). In contrast, much less attention is placed on how and when system failures happen (item not returned by the end of a conversation). The same holds for our CRS sub-task of interest, namely Conversational Image Recommendation (Guo et al., 2018; Wu et al., 2021a), where a system is assumed to be successful by a pre-defined number of evaluation turns. Instead, to consider real-life scenarios more accurately, in this thesis, we address the issue of CRS failure by identifying the indicators that relate to an effective recommendation. Specifically, we detect and predict conversational failures at various stages of a dialogue. Indeed, we consider its multi-turn setting, the user feedback at each turn, and the reduced returned list (often displaying only the top item). To this end, only few attempts have examined QPP in a conversational setting, and this was mainly CS. For example, research has adapted existing score-based predictors to the question level (the top-item as the answer) to determine the answer quality of a conversational assistant (Roitman et al., 2019), focused on the query ambiguity for determining whether a clarifying question is needed (Arabzadeh et al., 2022), or considered a geometric interpretation of the query contents in a conversations with few turns (Faggioli et al., 2023a).

Still, predicting the performance of Conversational Image Recommendation differs significantly, not only from predicting passage retrieval, but also from rankings in Conversational Search. Specifically, CRS do not contain relevance judgment information that usually comes with information retrieval rankings. Therefore, to predict conversation success, we need to rely on the ranking of the recommendation list at each turn, which reflects the result of user feedback on the previous turn. In addition, predicting a ranking of image items differs from text-based retrieval, where QPP is normally used, and employing external pre-trained multi-vector retrieval models such as BERT (Devlin et al., 2019) cannot be generalised to image items. For this purpose, we focus our attention on single-representation dense retrieval models (Lin et al., 2020; Xiong et al., 2020), which separately encode queries and documents and retrieve items based on nearest neighbour search. In this way, we can more easily generalise to the embedded representations of a multi-modal task, which is composed by text-based user feedback and image-based recommendation lists. To be more precise, in this thesis, after examining dense embedded representation based in its original setting and using dense retrieval models, we propose a variety of both unsupervised and supervised predictors that are based on learning the embedded representations already contained in CRS models to predict conversational failures. Also, while QPP is evaluated at the query level, we propose an evaluation at the conversation level. To summarise, while research has widely studied CRS performance and at the same time QPP has been studied as an information retrieval task in a single-turn setting or with independent interaction turns, we fill in the gap of studying predictions in a multi-turn and multi-modal conversational recommendation setting by proposing a new evaluation methodology of QPP specifically designed for CRS or as we call it, Conversational Performance Prediction (CPP). In other words, this thesis lies at the intersection of CRS and QPP.

Furthermore, the common assumption in the conventional evaluation of CRS systems including Conversational Image Recommendation systems, is that the target item exists in the catalogue and must be returned. However, in a real-life shopping scenario, this might not always be true. For example, a lot of times, when searching for a fashion item, it might be sold out and therefore, does does exist in the item catalogue. In other cases, a user might have a more flexible need, which is equally satisfied with an item similar to the original target (in terms of a given criterion such as colour, shape, etc.). Still, these options are currently not incorporated within the evaluation methodology of the various state-of-the-art models (Guo et al., 2018; Wu et al., 2021a,b). First, the evaluation of CRS is limited to single target items; still, this may not reflect the ability of the CRS to return other items that might also be relevant to the user. To address this, we improve the evaluation methodology by providing datasets and the corresponding user simulators with better completeness, inspired by TREC pooling in ad-hoc retrieval. Second, current evaluation methodologies cannot predict when a system fails to identify an item or it fails to inform the user that the item does not exist. For this purpose, we introduce the concept of recommendation scenarios. In this regard, to strengthen our predictions for the different cases of

retrieval failure, we introduce two novel scenarios: one that considers items not available in the catalogue, and another one that incorporate the pooling judgments for completeness to address cases of alternative items that are also relevant. Using both of these additional scenarios, we extend our conversational prediction framework. To summarise, we fill in the gap of studying CRS performance under different scenarios.

## 1.2 Thesis Statement

The statement of this thesis is that the performance of a Conversational Recommendation System can be predicted to detect when a conversation fails, under different scenarios, across different turns of a conversation. Initially, we can predict the effectiveness of a ranking of textual items for a textual query, by examining the coherence of the top-retrieved items based on their dense embedded representations. Similarly, we can predict the effectiveness of a ranking of items in a Conversational Recommendation Systems (CRS), which are also based on learned embedded representation of images, where user feedback takes the place of a textual query. Indeed, by introducing a framework of Conversational Performance Prediction (CPP), we can predict the degree of success of a conversation by a CRS - such success can be predicted over a short or long time horizon, thereby predicting current user satisfaction or overall satisfaction of a conversation. Furthermore, by obtaining user opinions about the relevance of items, we improve the completeness of the evaluation mechanism by identifying alternatives recommendations for existing target items, which could be used to both inform the user simulator and therefore improve the overall evaluation of CRS systems. Finally, using these alternatives datasets, and by predicting conversational performance under different Recommendation Scenarios, we detect different types of conversational failure, such as when a user cannot find an item, versus when the system's catalogue does not contain the relevant item.

## 1.3 Contributions

The contributions of this thesis can be summarised as follows:

1. We propose a set of coherence-based dense query performance predictors (QPPs) that are specifically designed for single-representation dense retrieval models and adopt an evaluation methodology that explains discrepancies between correlation results among different evaluation metrics.

Existing QPPs were mainly used to predict rankings of sparse retrieval models,and therefore, cannot be generalised to dense embedding-based lists including those of images rather than documents. In Chapter 3, we show that the examination of the relations among dense embedded representations of the document list is sufficient to provide effective predictions for

single-representation dense retrieval models, namely ANCE (Xiong et al., 2020) and TCT-ColBERT (Lin et al., 2020), the type of models that can more easily be representative of a multi-modal ranking model. In addition, by using a multi-level perspective that jointly considers QPPs and types of queries, we explain why some QPPs are better for certain types of queries for MAP@100, while they are more robust when correlated with NDCG@10 for dense retrieval models.

2. We create a novel task which we call Conversational Performance Prediction (CPP) task, which predicts conversation success at the conversation level, and can predict at what stage of a conversation a retrieval failure is likely to happen, thus extending the original QPP task to conversational recommendation.

While the QPP task is widely studied, no work has addressed the issue of predicting conversation success using specific indicators. Therefore, in Chapter 4, we develop a new evaluation framework which we call Conversational Performance Prediction (CPP), and show how we can predict conversation failures at different prediction horizons. First, we examine a range of both score-based and embedding-based unsupervised predictors at the conversation level and show that using correlations is not an optimal evaluation strategy for predicting conversational failures. Consequently, we propose a supervised CPP approach, which treats CPP as a binary classification task and show that by learning the embedded representations already contained in the CRS models we can predict the accuracy of a conversation success using the retrieved items of both single and multiple turns.

3. We obtain real user opinions about the relevance of items, thus identifying alternatives relevance labels for a number of target items, and in turn, inform the user simulator with an extended target space. This increases the completeness of CRS evaluation, and therefore, creates a more realistic prediction setting for CRS, which leads to improved predictions of user preferences.

State-of-the-art CRS models are trained using user simulators with a single target item in mind, and are infinitely patient. These settings do not correspond to a real user shopping scenario. Therefore, in Chapter 5, we use crowd-sourcing to collect relevance labels for a number of identified target items using pooling, thereby creating relevance judgments similar to information retrieval tasks. Then, we reevaluate the CRS models using the updated simulator with the identified alternatives as part of the target space and show that by using a single target for an unlimited amount of turns was underestimating the effectiveness of CRS models.

4. We adapt our CPP framework for different types of conversational failures, which are determined by whether the user need is clearly defined and whether the target item is available.

We introduce the concept of recommendation scenarios: First, we consider the case when the target is not available in the catalogue (removed target scenario) and then the case where a user has a more flexible (not clearly defined) user need, which can be satisfied by either the original target or any of the identified alternatives in Chapter 5. In particular, we examine the supervised CPP predictors introduced in Chapter 4 under these two novel scenarios, and find that in both cases, there is a marked difference from the original scenario, and this effect is different according to the rank cutoff of the examined ranking. This is the chapter that connects Chapter 4 and Chapter 5.

## 1.4 Origins of Material

Part of the material presented in this thesis is based on papers published during this PhD programme. Specifically, for some chapters, we are based on the following papers:

- In (Vlachou and Macdonald, 2024a), we propose a set of and further adapt some other coherence-based query performance predictors exclusively for the task of single-representation dense retrieval, and showing how this performance can vary across different evaluation metrics. The reason for this is because these predictors are easily generalisable to our multi-modal recommendation task. This work was published at the ACM ICTIR 2024 conference and contributes to our Chapter 3.

- In (Vlachou and Macdonald, 2022), we introduce our Conversational Performance Prediction (CPP) evaluation framework, which adapts score-based predictors to predict Conversational Image Recommendation rankings both at the single-turn and the consecutive-turn level. This work, which corresponds to the early stage of our evaluation methodology, was published in the ACM RecSys 2022 KaRS Workshop and contributes to the first part of Chapter 4.

- In (Vlachou and Macdonald, 2024b), we introduce the concept of alternative options to a given target item and we collect real user opinions about what they would select as an alternative in case their desired item does not exist. We further re-train the CRS models based on the newly obtained "relevance judgments". This work was published on arXiv and is currently under review, and is the basis for our Chapter 5.

## 1.5 Thesis Outline

The rest of this thesis is structured as follows:

- Chapter 2 provides background information about the different Conversational Information Seeking (CIS) tasks, and emphasise on the relevant literature on the state-of-the-art

Conversational Image Recommendation models, their evaluation methodology, and the related datasets and user simulators. In addition, we provide an overview of the state-of-the-art QPP methods and the different evaluation metrics.

- Chapter 3 presents our dense coherence-based predictors specifically designed for dense retrieval that consider top to bottom rank relationships of the embedded representations already produced by these models. Also, we present a multilevel approach for studying QPP in different evaluation metrics by considering the contribution of the different types of queries (Bolotova et al., 2022).

- Chapter 4 presents our novel Conversational Performance Prediction (CPP) framework, which predicts CRS performance at the conversational level, extending over both short-term and long-term horizons. Next, we focus on examining CPP using a smaller amount of queries, similar to the QPP setup (Carmel and Yom-Tov, 2010; Cronen-Townsend et al., 2002). Finally, we extend our evaluation framework to a binary classification task, which we call supervised CPP, by creating a correspondence with the different groups of QPP predictors, and propose a supervised predictor that gradually learns a compressed representation of the retrieved items of previous turns.

- Chapter 5 presents our Meta-Simulator, an updated user simulator for Conversational Image Recommendation models (Guo et al., 2018; Wu et al., 2021a) that uses the new target space created by incorporating the range of identified alternatives for each original target item that we found in a user study. Consequently, we retrain the CRS models by considering each alternative as equally relevant as the target, and report the results by comparing our Meta-Simulator with the base user simulator.

- Chapter 6 presents our two novel recommendation scenarios, namely the Missing Target (target item is not available in the catalogue) and the Alternatives (each identified alternative option from Chapter 5 is equally satisfying for a user that has a more flexible user need) scenarios. We extend our supervised CPP framework under the different scenarios and introduce the corresponding types of retrieval failure, namely catalogue failure and alternatives failure, both of which differ from a regular system failure.

- Finally, in Chapter 7, we summarise our contributions and provide concluding remarks, and end with suggestions that follow as a direction following the results and insights presented in this thesis.

# Chapter 2

# Background and Related Work

As mentioned in Chapter 1, the focus of this Thesis is on Conversational Recommendation, which belongs to the family of Conversational Information Seeking tasks (Zamani et al., 2023). Broadly defined, information search tasks can be categorised as informational, navigational, and transactional (Broder, 2002; Jansen et al., 2008). In other words, a user's intent is not always informational in nature (i.e, trying to acquire information from a web page); it can also be navigational (reach a particular site) or transactional (perform some kind of activity on the web) (Broder, 2002; Jansen et al., 2008). We are particularly interested in the transactional intent. Specifically, the activity we are interested in is online shopping and the fashion domain. In particular, we focus our attention on *Conversational Fashion Image Recommendation* in an interactive setting. The task can be best described in Figure 2.1. A user has a specific information need operationalised with an imagined image item, such as a shoe. The system first provides a random initial recommendation, on which the user provides natural language feedback after turn 1 that guide the system in finding items relatively improved than the current recommendation. At turn 2, the next item is closer to the user need than the previous. In practice, such systems are trained using *user simulators* that act as real users to provide feedback on the recommended item at each turn in order to provide a sufficient amount of data (Guo et al., 2018; Wu et al., 2021a).

This example shows a successful recommendation setting, where a the system returns the item of interest at turn 4. In practice, this is quite a realistic setting with a few interaction turns. However, in many cases, users do not find their desired items even after a large number of interaction turns. This is exactly the type of scenario we are interested in. In particular, we focus on failed dialogues in Conversational Image Recommendation in online shopping. For this purpose, we aim at predicting failure both at the short and the long term of a conversation. We do this through the lens of Query Performance Prediction (QPP) (Carmel and Yom-Tov, 2010; Cronen-Townsend et al., 2002), originally proposed for search engines, which we adapt to a conversational recommendation setting. In this regard, we need to take into account its multi-turn nature of the task and the dependence of one turn on another.

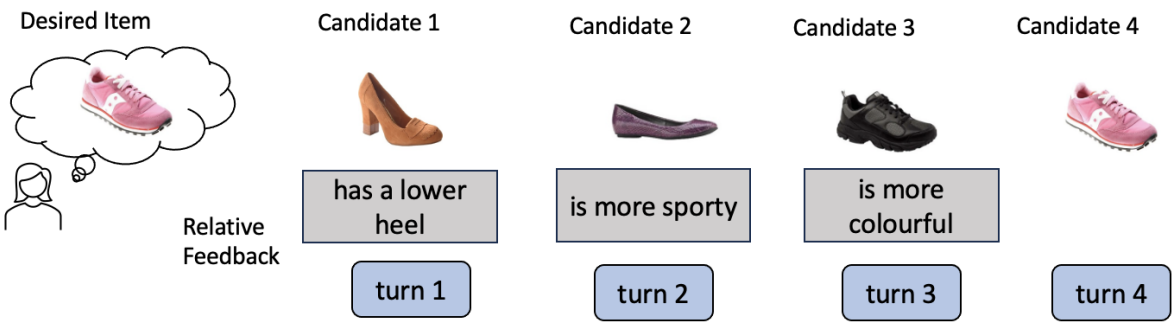



Figure 2.1: Schematic representation of the Conversational Fashion Image recommendation task. The desired item is shown on the left as the top of a ranking, and at each turn, the user receives a candidate item to provide natural language feedback on.

More specifically, in this chapter, we start with an overview of the different Information Seeking Tasks in Section 2.1, followed by an overview of the existing state-of-the-art Conversational Recommendation models in Section 2.2, and continue with more specific information about our task of interest in Sections 2.2.1 and 2.2.2. Then, we proceed with some background information referring to the second of our main themes, namely query performance prediction in Section 2.3, and the evaluation methods in Section 2.4, specifically the various evaluation measures of these tasks in Section 2.4.1 and the corresponding user simulators and existing datasets in Section 2.4.2. After that, we provide some concluding remarks and a set of limitations in Section 2.5.

## 2.1 Information Seeking Tasks

When users search for information on the web, they are driven by a so-called *information need* (Shneiderman et al., 1997). Often, they do not simply search for information on a web page, but they rather look to navigate to another page or for online shopping or another transaction (Broder, 2002; Jansen et al., 2008). Over the years, a variety of *Information Seeking (IS)* tasks have been proposed, which aim to satisfy the different types of information needs. The common line of these tasks is that a system returns a *ranked list* of results that users are likely to find *relevant* in response to their *queries*. In other words, the ranking of IS tasks is based on the relevance of each item to the information need expressed with a query. Below, we present a number of IS tasks with examples. More specifically, in Section 2.1.1, we present the task of ad-hoc retrieval, and differentiate between sparse and dense retrieval models, and in Section 2.1.2, we present a range of *Conversational Information Seeking (CIS)* tasks, namely Conversational Search, Conversational Question Answering, and Conversational Recommendation.

### 2.1.1 Ad-hoc Retrieval

Traditionally, search engines relied on the return of relevant documents to a set of keywords, a task called *ad-hoc retrieval* or *document retrieval*. This task is best described by a score function $s(q,d)$ as seen in Figure 2.2, which computes the relevance between the two input elements, a query $q$ resulting from a user information need, and a document $d$ from a document *collection* or *corpus* whose representation is obtained with a process called *indexing*, based on the similarity of their embedded representations. The estimation of this relevance score results in an output of a ranked list of retrieved documents appearing in descending order. Depending on the representation method of the queries and documents, retrieval systems can be sparse or dense. For the purpose of this thesis, we will not examine ad-hoc retrieval in much detail. The extent to which we use it (in Chapter 3) is to first examine the QPP task in its original setting, in order to examine with how compares with our conversational setting (in Chapters 4 and 5). In this section, we provide an overview of some commonly used sparse and dense retrieval models that are used in the following sections.

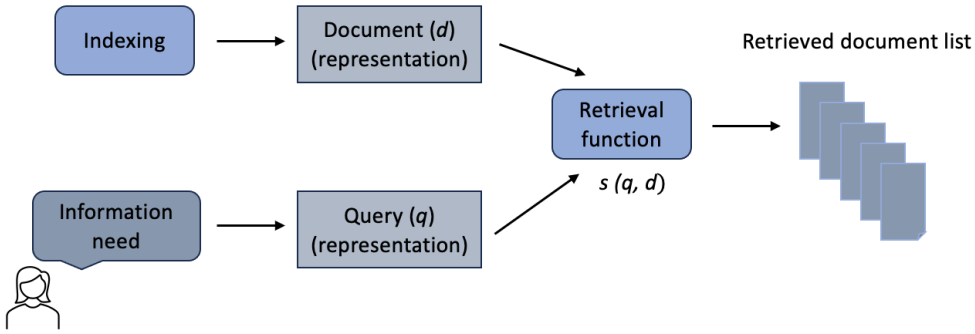



Figure 2.2: Schematic representation of the ad-hoc retrieval task. The retrieval function is determined by query and document representations.

#### Sparse Retrieval models

*Sparse retrieval* methods encode queries and documents into sparse vector representations by creating a vocabulary of all unique terms contained in the collection of all documents. In this representation, the dimensionality of the vector representation $v$ corresponds to the sum of all unique terms contained in both the queries and documents. Each vector of either the query or a document in the collection is binary represented for the presence of each of the vocabulary terms. Specifically, we define $V_q$ as a query binary vector representation, where $V_q = (\vec{q_1}, \vec{q_2}, ..., \vec{q_{|v|}})$ and $\vec{q_i}, i = 1, ..., v$ is a set of ordered index terms in the query representation, and $V_d = (\vec{d_1}, \vec{d_2}, ..., \vec{d_{|v|}})$ is a document vector representation, with $\vec{d_i}, i = 1, ..., v$ corresponding to a set of ordered index terms in the document representation, respectively. Then, the score function that measures the similarity estimation between the query and document representations corresponds to their

cosine similarity or more formally:

$$s(q,d) = cosine(V_q, V_d) \tag{2.1}$$

This is the simplest form of a sparse retrieval models and is called the *vector space model* (Salton et al., 1975). One problem with vector space models is that they ignore term-dependence relationships; for example, certain non-informative tokens need to be ignored from the vocabulary, as they do not add information to the similarity function (Scholer et al., 2002).

For this reason, a number of weighting schemes have been proposed. For example, the importance of a term can be expressed with the term frequency (*tf*, higher weight to terms that appear more frequently), or the inverse document frequency (*idf*, measures how rare a term is across the collection and gives higher weight to the terms that occur rarely). Idf is more formally expressed as:

$$idf(t,d) = log(\frac{N+1}{N_i+1}) \tag{2.2}$$

where $N_i$ is the number of documents that contain term $t$ and $N$ is the total number of documents in the collection (+1 is a smooting factor). The resulting weighting scheme corresponds to the *TF-IDF* model (Sparck Jones, 1972), which, as described, combines the count of index term occurrences, and where the score function is formally:

$$S_{TF.IDF}(q,d) = \sum_{t \in q \cap d} f(\eta(q,t), \eta(d,t)) = \sum_{t \in q \cap d} tf(t,d) \cdot idf(t,d) \tag{2.3}$$

where $tf.(t,d)$ is the frequency of term $t$ in document $d$ and $idf(t,d)$ is the inverse document frequency.

While vector space models use the sparse representations directly, other sparse retrieval models are based on probability theory. The most representative example of probabilistic retrieval models is BM25 (Jones et al., 2000; Robertson et al., 1995). BM25 incorporates the query and document term weights into the scoring function and nowadays it is still a very competitive ranking model, is used as a baseline for all examined new ranking models and is also used as a first stage retriever for re-ranking models. In this case, the scoring function can be written as:

$$\begin{aligned} S_{BM25}(q,d) &= \sum_{t \in q \cap d} f(\eta(q,t), \eta(d,t)) \\ &= \sum_{t \in q \cap d} idf(t,d) \cdot \eta(q,d) \cdot \eta(t,d) \\ &= \sum_{t \in q \cap d} idf(t,d) \cdot \frac{tf(t,q)(1+k_2)}{k_2 + tf(t,q)} \cdot \frac{tf(t,d)(k_1+1)}{tf(t,d) + k_1(1-b+b\frac{dl}{avgdl})} \\ &= \sum_{t \in q \cap d} log(\frac{N-N_i+0.5}{N_i+0.5}) \cdot \frac{tf(t,q)(1+k_2)}{k_2 + tf(t,q)} \cdot \frac{tf(t,d)(k_1+1)}{tf(t,d) + k_1(1-b+b\frac{dl}{avgdl})} \end{aligned} \tag{2.4}$$

where $N_i$ is the number of documents containing the token $t_i$, $k_1$ and $k_2$ control the scale the $tf$, and $b$ is the normalisation of document length $dl$. In this thesis, we use BM25 as a baseline setting for QPP compared to dense retrieval QPP approaches.

### Dense Retrieval models

In recent years, a novel form of self-attention deep learning models have been proposed called *Transformers* (Vaswani et al., 2017), initially applied on natural language processing tasks. The transformer architecture follows an encoder-decoder block structure to process the data. As a first step, the encoder maps an input sequence to a continuous representation for each input element. Then, the decoder uses the embedded representations to generate an output sequence. In this regard, a number of the *pre-trained language models (PLMs)* have been proposed. Most related to our interest, the *Bidirectional Encoder Representations from Transformers (BERT)* (Devlin et al., 2019) model pre-trains deep bidirectional representations by jointly conditioning on both left and right context in all layers and is then fine-tuned with an additional output layer to a wide range of tasks. Its conception is based on a masked language model (MLM) (Taylor, 1953), which randomly selects input tokens to mask, and then predicts the original vocabulary id of the masked word based only on its context. The result is contextualised embeddings, which can be used for a number of downstream tasks.

Indeed, using the transformer architecture, deep learning architectures have been fine-tuned for information retrieval tasks with several modifications to their embedded representations. In particular, BERT can be used as a re-ranker for a set of retrieved documents from a sparse retriever, usually BM25. Importantly, a due to BERT's increased computational cost, for example, compared to sparse retrieval models as explained above, *ColBERT* (Khattab and Zaharia, 2020), a ranking model based on contextualized late interaction over BERT, was proposed. In particular, ColBERT proposes a novel late interaction paradigm for relevance estimation between a query $q$ and a document $d$, where queries and documents are separately encoded into two sets of contextual embeddings, and relevance is evaluated using cheap and computations with a *MaxSim* operator as:

$$S_{MaxSim}(q,d) = \sum_{i=1}^{|q|} \max_{\{j=1,...,|d|\}} \phi_{q_i}^T \phi_{d_j} \tag{2.5}$$

where $|q|$ is a set of query embeddings, $|d|$ is a set of document embeddings, $\phi_q = \{\phi_{q_1}, ..., \phi_{q_{|q|}}\} = Encoder_Q(q)$, and $\phi_d = \{\phi_{d_1}, ..., \phi_{d_{|d|}}\} = Encoder_D(d)$.

ColBERT uses a multi-vector dense embedded representation for queries and documents. In cases where a more time-efficient approach is needed, another type of retrieval model was proposed called *single-representation dense retrieval*. In this approach, the multi-vector representation is passed into a single-vector representation usually via knowledge distillation. One example of this is the TCT-ColBERT model (Lin et al., 2020), Specifically, TCT-ColBERT distills the knowledge from ColBERT's MaxSim operator to compute relevance scores into a simple

dot product, thus enabling single-step ANN search. This is achieved via tight coupling between a teacher model and a student model, producing the following relevance function:

$$S_{PoolDOt}(q,d) = \langle Pool(E_q), Pool(E_d) \rangle \tag{2.6}$$

where the terms represent two pooled embeddings, and the *Pool* operator is the average pooling over token embeddings. Another related single-representation model is *Approximate Nearest Neighbour Negative Contrastive Learning (ANCE)* (Xiong et al., 2020), a learning mechanism that "selects hard training negatives globally from the entire corpus, using an asynchronously updated ANN index". In this way, it overcomes some issues causing dense retrieval to be outperformed by sparse retrieval models. Indeed, by using this asynchronous trainining, ANCE eliminates the problem caused by other dense retrieval models that use hard negative samples identified by BM25 returned items, such as *Dense Passage Retriever (DPR)* (Karpukhin et al., 2020). DPR also uses ANN search like ANCE, but is less effective than ANCE, as it does not update the dense index during hard negative sampling. In general, with dense retrieval, a system finds the documents whose neural embeddings lie closer to the corresponding query embeddings. A schematic representation of single-representation dense retrieval models is given in Figure 2.3.

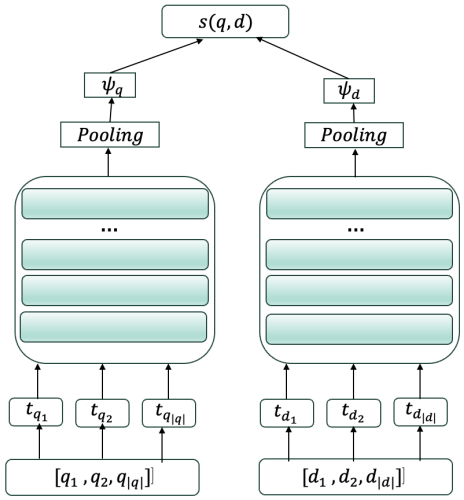



Figure 2.3: Schematic representation of the single-representation dense retrieval task. The retrieval function is enabled after pooling the dense embedded representations to a single-vector representation, thus allowing search of nearest neighbours.

### 2.1.2 Conversational Information Seeking

The recent development and increasing popularity of smart devices is leading users to switch from traditional search engines to more customised and interactive information seeking platforms. Nowadays, dialogue systems and voice assistants, such as Amazon Alexa, Apple Siri or Google Assistant (Argal et al., 2018; Brill et al., 2019; Dalton et al., 2018) are becoming

prevalent when people prefer to engage in a conversation. Indeed, people use these assistants in various *Conversational Information Seeking (CIS)* tasks; these tasks refer to a system that helps users to satisfy their information needs by engaging in conversations (Zamani et al., 2022). Both inputs and outputs of CIS systems can be of multiple sources such as natural language text, images, clicks, voice etc (Deldjoo et al., 2021; Hauptmann et al., 2020; Lei et al., 2020a). Despite the various definitions of CIS, researchers agree on the following main requirements for a system to be considered as CIS: (i) the system is pro-actively involved with supporting the user to satisfy their information needs, and therefore, it requires *mixed-initiative* by both the user and the system (both sides initiate the conversation and can request for more information by asking questions) (Andolina et al., 2018; Radlinski and Craswell, 2017; Trippas et al., 2018), and (ii) the dialogue develops over more than one utterance for each conversation participant (user and system), and therefore, it requires multi-turn interactions, which can be over one or more sessions and are enhanced by asking clarifying questions (Aliannejadi et al., 2019). Below, we provide some examples of the most common CIS, namely Conversational Search, Conversational Recommendation, and Conversational Question Answering.

### Conversational Search

While Information Seeking and searching for information has always been considered as an interactive process (Croft and Thompson, 1987; Oddy, 1977), the progress in machine learning and natural language processing has allowed users to express their intent in natural language form (Zamani et al., 2023), and it has led to the development of more advanced CIS such as Conversational Search. In particular, *Conversational Search (CS)* is the type of CIS which refers to interactively searching for information with a conversational system using natural language conversations (Zamani et al., 2023). As for what constitutes a CS system, over the last few years, a number of definitions have been proposed that generally describe a system that interactively retrieves information between a user and an agent which allows speech and natural language properties (Anand et al., 2020; Radlinski and Craswell, 2017). Still, some differences exists in terms of the criteria they use to define such systems. For instance, Radlinski and Craswell (2017) defined a conversational search system in terms of the properties that need to be met. Specifically, a CS system is a system for retrieving information that permits interactive dialog or a *mixed-initiative* between a user and an agent, where the agent's actions are determined according to a model of current user needs. From a different perspective, Anand et al. (2020) defined CS systems in terms of how they differ from other IR systems or disciplines. For example, it is an interactive IR system, a retrieval-enhanced chatbot or a "dialog system with retrieval capabilities".

Radlinski and Craswell (2017) presented a framework for CS where the interactive dialog process can be initiated by either the system, or the user. The system can select between different actions, which involve providing a partial or full description of one or more items, or requesting

the user to provide their information need, rating, or natural language critique at a given turn. In turn, the user actions or responses can be the initial description of their information need, a rating, a preference or a natural language critique at a given turn. A particular advantage of CS systems is that they ask questions to the user, which leads to more accurate understanding of user needs and higher confidence of the system in its results (Aliannejadi et al., 2019; Zhang et al., 2018). Once the user provides their response to a given question, the system considers not only the user's initial request, but also the content of the response in order to both provide a ranking list of items and to generate new questions in the search process (Zhang et al., 2018). More formally, given a search topic a user wants to learn more about, a conversation $c \in C$ can be defined as a list of user-system interactions with each interaction turn being composed of a query $q$ and an answer $a$ (Lipani et al., 2021) as follows:

$$c = [(q_1, a_1), ..., (q_k, a_k)] \tag{2.7}$$

where each pair $(q_i, a_i)$ is a pair of a query and user response at a conversation turn $k$.

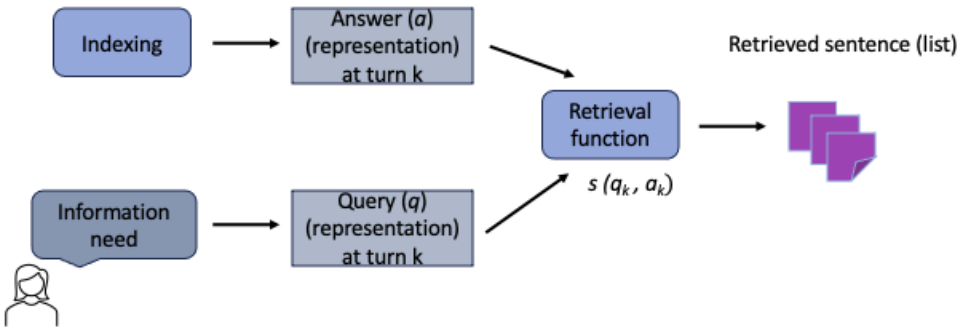



Figure 2.4: Schematic representation of the Conversational Search task. The resulting multi-turn ranking of passages is shown with a different colour to demonstrate the difference in length.

In contrast to ad-hoc retrieval, CS enables the user to provide their query in natural language form, which in turn leads to a more natural style. For example, when looking at Figure 2.2, we see entire long passages being retrieved in a single round in response to a keyword-based query. On the other hand, Figure 2.4 shows the CS procedure, where the retrieved information (depicted with a different colour and size to showcase the difference in type and length) is divided into pieces and is part of an overall turn-taking dialog process, which is more intuitive and by giving users the opportunity to refine their query, it creates a user need model of preference elicitation (Zamani et al., 2023). In other words, the retrieved units are short sentences or images instead of passages, and the ranked list of items is usually shorter consisting of one or just a few items depending on the platform.

### Conversational Question Answering

In general, Question Answering (*QA*) is a form of information seeking where the user need is expressed with a question in natural language form (Zamani et al., 2023). Indeed, unlike traditional retrieval systems that return lists of full documents, QA retrieves short pieces of information to answer users' queries in the form of text snippets that contain the exact answer (Gao et al., 2018; Voorhees et al., 1999), for example, the resulting answer can be a short passage, a sentence or a phrase (Lu et al., 2019). The progress in conversational assistants has also influenced the recent developments in QA leading to a new task, namely Conversational Question Answering (*Conversational QA* or *ConvQA*). Specifically, ConvQA is a sub-type of CIS, and its main difference from traditional QA is that it allows users to express their need with more than one questions in a conversation. This further implies that systems should be able to handle complex linguistic characteristics such as anaphoras (reference to previous conversation turns) (Vakulenko et al., 2021). In that sense, ConvQA is similar to CS described above, but with a narrower focus (Zamani et al., 2023). Due to the varying nature of questions asked and the conversational nature of the task, ConvQA is sometimes indistinguishable from CS.

### Conversational Recommendation: From Traditional to Dialog-based Recommender Systems

Typically, *Recommender Systems (RS)* refer to applications that assist users in finding items of interest in cases of information overload (Ricci et al., 2015) and in user exploration (Chen, 2021). Also, in the context of e-commerce settings, RS help business providers with promoting their service (Jannach and Jugovac, 2019). The classical context of recommendation algorithms is a one-shot interaction process, where the system keeps track of user data over time and when the user enters the system, it provides a set of tailored recommendations. In this regard, RS recommend items on the basis of user feedback in terms of ratings, clicks or reviews. For example, an earlier line of research centers around the concept of *collaborative filtering (CF)* (Herlocker et al., 2000; Konstan et al., 1997; Schafer et al., 2007; Su and Khoshgoftaar, 2009; Ungar and Foster, 1998). CF techniques are either neighbourhood-based (for example, inferring a user's preference for an item based on similar ratings of "neighboring" items by the same user) or based on latent factor models, which explain the ratings by scoring both items and users on a number of factors that predict how much a user likes an item based on specific factors). As an evolution to CF, *matrix factorisation (MF)* algorithms (Koren et al., 2009; Mehta and Rana, 2017) represent items and users with latent vectors and combine those representations in a user-item matrix of sparse ratings (a given user is likely to have rated a small percentage of items). The inner product of user-item interactions approximates the user's interest in a given item. Finally, more recently, neural network-based models have been proposed (Kang and McAuley, 2018; Sun et al., 2019), which use self-attention and complex network structures such as Gated Recurrent Units (GRUs) (Cho, 2014) to model user behavior sequences. In particular, *sequential*

*recommendation* (Hidasi and Karatzoglou, 2018; Hidasi et al., 2015; Kang and McAuley, 2018) models sequential model employ an attention mechanism to use only few actions at each time step to identify relevant items from a user action history, and use those to predict the next item.

Most of the above examples, and especially the older RS methods (Herlocker et al., 2000; Konstan et al., 1997; Koren et al., 2009; Mehta and Rana, 2017; Schafer et al., 2007; Su and Khoshgoftaar, 2009; Ungar and Foster, 1998), have a common aspect, which refers to their static interpretation of user preferences. This means that they assume an existing database of user-item ratings or implicit feedback up to a certain time point in an offline setting, and once the user is logged in the service, a recommendation can be provided based on a predicted preference. In contrast to this approach, Therefore, another related CIS task, namely *Conversational Recommendation*, provides a more dynamic interpretation and satisfaction of user needs. As mentioned above, in recent years, online shopping platforms such as Amazon are becoming dominant when users look for items of interest. As a result, a variety of Conversational Recommendation Systems (CRS) have been proposed, which assist users with finding items and with decision making (Sun and Zhang, 2018; Zou and Kanoulas, 2019). Unlike traditional RS, *Conversational Recommendation Systems (CRS)* allow for more complex recommendation settings, since they assist users in a number of task-oriented goals in the context of a dialogue (Jannach et al., 2021). CRSs can be useful in a number of cases. For example, while users sometimes know their preferences when they visit a platform or a system, they might still construct their preferences as they enter and find out about the options (Wang and Benbasat, 2013). Also, during the interaction with the system, they are informed about the available options (Wärnestål, 2005).

In general, CRSs share some important common aspects that distinguish them from traditional RS. In this regard, the CRS definitions that have been proposed are converging. For instance, Jannach et al. (2021, p. 105) define CRS as "*software systems that support users in achieving recommendation-related goals through a multi-turn dialogue*", while Gao et al. (2021, p. 101) emphasised that CRS can "*elicit the dynamic preferences of users and take actions based on their current needs through real-time multi-turn interactions*". Therefore, an important aspect of CRS is their multi-turn nature (more than one utterance for each side, user and system). Due to the interactivity with users, CRS help with dynamic preference elicitation (Christakopoulou et al., 2016; Li et al., 2018; Sun and Zhang, 2018; Zhang et al., 2018) in CRS is also enhanced by the fact that modern systems allow users to express their preferences through natural language feedback. Feedback in this context is called critiquing (Tou et al., 1982). The interactive nature of critiquing-based CRS can increase the effectiveness of recommendations in a series of interactions (Chen and Pu, 2007, 2012; McCarthy et al., 2004). Another critical aspect of CRSs is that they are goal-oriented, meaning that they assist in the completion of specific tasks by recommending products (Jannach et al., 2021).

As described by Jannach et al. (2021), the typical architecture of CRS includes the following

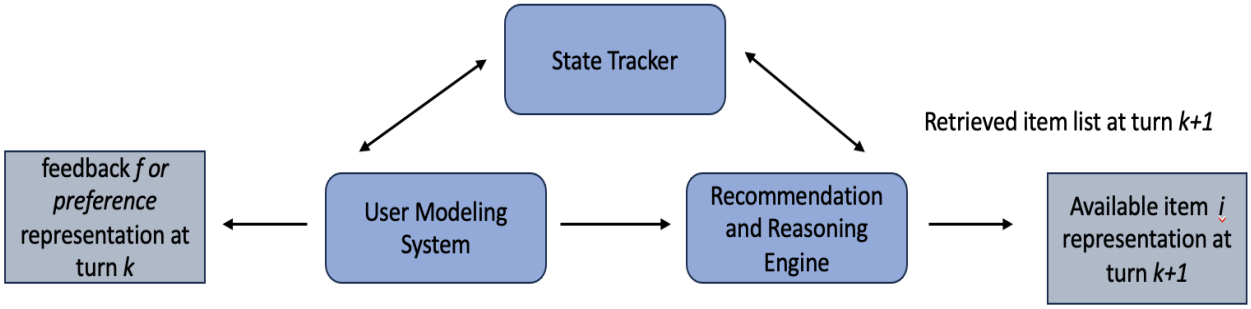



Figure 2.5: Schematic representation of the Conversational Recommendation System architecture (adapted from Jannach et al. (2021)). The state tracker controls the interaction between the use modeling system and the recommendation engine, while the database with all available items can be accessed. The user profile is updated at each turn.

modules: First, we have the *dialog manager* or *state tracker*, which processes the received user actions, updates the user profile during the dialog and decides on the next action. The two other main components refer to the user and system side, respectively. Specifically, the *user modeling system* controls the user profile that is composed of their preferences, whether these are long-term or not, and how they are updated during the dialog. Finally, the *recommendation and reasoning engine* retrieves recommended items from the item database, while it can also produce explanations for the recommendation it provides. All three components may have access to some background knowledge regarding the specific domains. The architecture is described schematically in Figure 2.5. In general, the multi-turn nature of CRS assumes some kind of memory functionality during the dialog, and therefore, storing past interactions and historical data is important in CRS systems (Jannach et al., 2021).

CS and CRS have a number of similarities. Importantly, both tasks aim to rank items based on their graded relevance and consequently provide users with relevant items according to the resulting ranking, either through a query (search) or user preference (recommendation) (Belkin and Croft, 1992). Furthermore, both systems will interact with the user through natural language conversations. Still, while in CS the interaction is based on written or spoken language, in CRS other modalities are possible such as images. Also, unlike CS, the purpose of CRS involves understanding users' preferences and providing suggestions, which requires more complex mechanisms such as keeping track of the user feedback and system actions (Jannach et al., 2021).

To provide a more complete description of the task, we need to refer to specific recommendation algorithms. Therefore, in the following section, we present the main types of CRS that have been proposed in the recent years, followed by some concrete examples of CRS models, and we continue with a presentation of the CRS paradigm that we use in this thesis, namely Conversational Image Recommendation.

Table 2.1: Summary of Information Seeking (IS) Tasks with a description of input and output elements, number of turns and context of use for each task.

| Task | Input | Output | # Turns | Context |
|---|---|---|---|---|
| Ad-hoc retrieval | query | passage | 1 | search engines |
| Conversational Search | query | sentence-level passage | >1 | digital assistants |
| Conversational QA | query | sentence-level passage | >1 | digital assistants |
| Conversational Recommendation | history, feedback | item(s) (text or image) of interest | >1 | digital assistants |

## 2.2 Conversational Recommendation Models

In recent years, various CRS models have been proposed, most of which are based on textual embedded representations of both user feedback and system-suggested items (for example, system descriptions about a recommended movie and questions to users, or a user explanation about their preference). Text-based CRSs can be roughly categorised into *attribute-based* and *topic-guided*. Attribute-based CRSs ask about the presence of certain attributes in the desired item (Christakopoulou et al., 2018; Luo et al., 2020; Sun and Zhang, 2018; Zhang et al., 2018), usually in a fixed number of questions until a recommendation is made at the last turn of the dialog (Lei et al., 2020a,b; Zou et al., 2020). On the other hand, topic-guided CRSs interact with users through natural language conversations, which enables more accurate responses and recommendations based on the dialogue semantics instead of attributes (Chen et al., 2019; Liu et al., 2020; Ma et al., 2020; Tu et al., 2022; Zhou et al., 2020). This is achieved by following a *topic path* (Ren et al., 2022), which is a pattern of preference elicitation guided by the system centered around a topic of interest. Given a topic path, short-term (historical data) and long-term (natural language feedback) user preferences can be modeled separately (Ren et al., 2022), and the system accordingly selects between a clarification question or a recommendation. In some cases, a hybrid approach is taken where a given algorithm determines when it is appropriate to ask a clarifying question and recommendations can be made more than once in a dialog. In this section, we give an overview of a number of widely used recently proposed CRS, followed by the state-of-the-art CRS models of our setting of interest, namely Conversational Image Recommendation. Finally, we briefly introduce the concept of user simulators in conversational systems and explain how we use such simulators for the purpose of this thesis.

### 2.2.1 Text-based CRS models

For the purpose of this thesis, we refer to *text-based CRS models* as the type of CRS models that use retrieved items as text on top of the text-based feedback. This is to distinguish these models from our task of interest (described in detail below in Section 2.2.2), which retrieves image items at each turn based on natural language feedback. The majority of text-based CRS use a deep learning approach, and normally they involve complex architectures to describe each of the CRS components (see Figure 2.5). Still, these models differ not only in the specific architecture

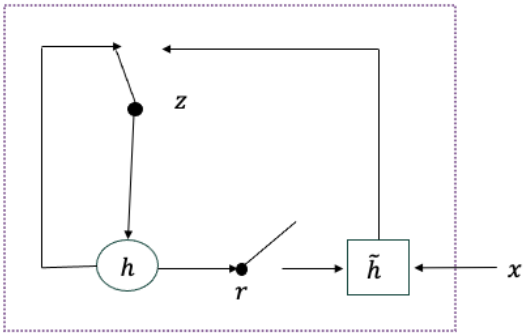



Figure 2.6: Schematic representation of the Gated Recurrent Unit (GRU) hidden activation function. The update gate $z$ decides whether the hidden state $h$ is updated with a new hidden state $\tilde{h}$, while the reset gate $r$ decides whether the previous hidden state is ignored. Adapted from Cho (2014).

modifications, but also on their focus on obtaining and updating users' preferences, asking clarifying questions, when and how many times an item is recommended during a dialog etc. Below, we present the main CRS models that are related to our approach and are further mentioned in the remaining chapters of this thesis. Specifically, we start with the GRU model (Cho, 2014), originally proposed for natural language processing, that has inspired a number of CRS models, and then we move on to present some examples that have incorporated GRU functionalities in some of their components.

### Gated Recurrent Unit (GRU)

The *Gated Recurrent Unit (GRU)* model (Cho, 2014) was originally proposed for statistical machine translation (SMT) and used conditional probabilities of phrase pairs. In general, for an RNN network, an output sequence of symbols is given by $p(x) = \prod_{t=1}^{T} p(x_t|x_{t-1},...,x_1)$, which is the conditional distribution of an input sequence $x = (x_1,...,x_T)$. Going further, GRU proposes an encoder-decoder RNN architecture that first encodes a variable-length sequence into a fixed-length vector representation and then decodes the vector representation back into a variable-length sequence; essentially, it learns the conditional distribution over a sequence conditioned on another sequence or more formally:

$$p(y_1,...,y_{T'}|x_1,...,x_T) \tag{2.8}$$

where $T$ and $T'$ are the input and output sequence, respectively. This is achieved by using a hidden state $h_{\langle t \rangle}$ which reads each symbol of input $x$ at each timestep and provides a summary $c$ of the sequence. In particular, the decoder generates the output sequence by predicting the next symbol $y_t$ given $h_{\langle t \rangle}$ as

$$h_{\langle t \rangle} = f(h_{\langle t-1 \rangle}, y_{t-1}, c) \tag{2.9}$$

which means that the hidden state is conditioned on the previous symbol and on the summary of the input sequence. The conditional of the next symbol is given by:

$$p(y_t|y_{t-1}, y_{t-2}, ..., y_1, c) = g(h_{\langle t \rangle}, y_{t-1}, c) \tag{2.10}$$

where $g$ is another activation function. The main contribution of GRU is the new type of hidden unit which is depicted in Figure 2.6, describing the activation of the $j-th$ hidden unit as follows:

$$h_j^{\langle t \rangle} = z_j h_j^{\langle t-1 \rangle} + (1 - z_j)\tilde{h}_j^{\langle t \rangle}) \tag{2.11}$$

In other words, the proposed hidden unit is jointly determined by a *reset gate* $r_j = \sigma([W_r x]_j + [U_r h_{\langle t-1 \rangle}]_j)$ and an *update gate* $z_j = \sigma([W_z x]_j + [U_z h_{\langle t-1 \rangle}]_j)$. Then, the updated hidden state is:

$$\tilde{h}_j^{\langle t \rangle} = \phi([W_x]_j + [U(r \otimes h_{\langle t-1 \rangle})]_j) \tag{2.12}$$

Therefore, if the reset gate approaches 0, the hidden state ignores the previous hidden state and is more influenced by or "resets" with the current input $x$, thereby dropping any unnecessary information. At the same time, the update gate corresponds to the "memory" of the previous hidden state, and therefore, it determines how much past information passes to the current hidden state. In this way, the two parallel gates provide each hidden unit the opportunity to reflect more short-term (reset gate) or longer-term (update gate) dependencies. This property of GRU is very useful for CRS models, since it corresponds to the long and short-term user preference modeling. This is particularly useful in a multi-turn conversational setting, where it is important to distinguish between recommending an item based on the feedback of the previous turn or whether information of all past turns should be considered or even whether the long-term history should contribute to preference estimation. Consequently, GRU has been adapted for a number of query suggestion (Sordoni et al., 2015) and CRS models (Guo et al., 2018; Li et al., 2018; Ren et al., 2022). In what follows, we present a few examples of CRS models that incorporate the GRU functionality.

### Context-aware and topic-guided CRS

**REDIAL** One early example of CRS that introduced the research setting of *conversational movie recommendation* was the *REcommendations through DIALog (REDIAL)* model (Li et al., 2018). REDIAL proposed a CRS model that acts like an agent chatting with a partner in a dialogue. Following the style of a friendly discussion, it is assumed to to lead a pleasant experience and ideas for movies to watch. The development of such a conversational agent involves a two-party conversation of a *recommender* and a *recommendation seeker*, where the seeker is expected to chat with the recommender about their movie tastes with an aim to get recommendations in a cold-start setting. The input-output structure of REDIAL uses the basic concept GRU (Cho,

2014), and specifically the more recent approach of the hierarchical recurrent encoder-decoder (HRED) as proposed in Sordoni et al. (2015), while it is enhanced by further elements to ensure it is adapted to a recommendation setting. Specifically, the model architecture is composed of:

- A hierarchical recurrent encoder (HRED) (Serban et al., 2016; Sordoni et al., 2015; Subramanian et al., 2018) that encodes the general purpose sentence representations (GenSen) from a bidirectional GRU (Cho, 2014) that are pre-trained in the encoder obtained from Subramanian et al. (2018). In particular, each utterance $U_m$ is modeled as a sequence of $N_m$ words as $U_m = (w_m, 1, ..., w_m, N_m)$ such that each dialog is encoded as a set of utterances $D = ((U_1, s_1), ..., (U_M, s_M))$ (with dialogue steps where $m = 1, ..., M$) with roles (seeker or recomender) denoted with $s \in -1, 1$. Given an input sequence $i_1, ..., i_T$, and learned parameters $W_{**}$ and $b_*$, HRED computes the reset gate $r_t$, update gates $z_t$, new gates $\tilde{h}_t$, and forward hidden state $h_t$: $r_t = \sigma(W_{ir} i_t + W_{hr} \vec{h}_{t-1} + b_r)$, $z_t = \sigma(W_{iz} i_t + W_{hz} \vec{h}_{t-1} + b_z)$, $\tilde{h}_t = \tanh(W_{i\tilde{h}} i_t + b_{i\tilde{h}} + r_t \otimes (W_{h\tilde{h}} \vec{h}_{t-1} + b_h \tilde{h}))$, and $\vec{h}_t = (1 - z_t) \otimes \tilde{h}_t + z_t \otimes \vec{h}_{t-1}$. Utterances are passed to the sentence encoder bidirectional GRU, which leads to conversational representations at each turn.

- An RNN component for Movie Sentiment Analysis: The model predicts for both the seeker and the recommender three labels about whether a movie was suggested (binary), and seen and liked by the seeker (three-class categorical). This is achieved by modifying the utterance encoder to take movie entities into account and adding a dimension to the hidden state in case a movie is mentioned. Applying activation functions to the last utterance representations, it obtains predicted probabilities for the each category.

- An Autoencoder Recommender, which acts as the dialog component. The model has no past information about the seeker's preferences; those are built during the dialog. This is done by predicting their ratings from a partially-observed user-movie matrix: Representing each user as a vector, it projects it in a smaller space and retrieved it again in its full version using a denoising autoencoder (Sedhain et al., 2015; Vincent et al., 2008).

- A Decoder with a Movie Recommendation Switching mechanism: Given $|V'|$ movies and at a given dialogue turn $m$, the sentiment analysis predicts whether a seeker liked a particular movie. The prediction creates an input $r_{m-1} \in \mathbb{R}^{|V'|}$ which produces a rating vector $\hat{r}_{m-1} \in \mathbb{R}^{|V'|}$. This combined with the context of previous utterances from the hierarchical encoder $h_{m-1}$ are used by the decoder to predict the next utterance by the recommender using a GRU with a switch to select between word or movie tokens.

Training such a system is challenging, as a large amount of data is needed to train this complex neural network structure. In this regard, a two-party conversational corpus was collected with crowd-sourcing, where one worker acted as the seeker and the other as the recommender with specific instructions to discuss about movies and mention a number of them during the dialogue.

At the end, further data was collected about whether each movie was mentioned, seen or liked by each participant.

**TG-DERIAL** While REDIAL (Li et al., 2018) provided the basis for conversational movie recommendation, it lacks the ability to semantically guide the conversation towards a goal recommendation scenario. Therefore, Zou et al. (2020) introduced the concept of *topic-guided* conversational movie recommendation. In particular, it is assumed that a user $u$ is associated with a profile $P_u$, which corresponds to a set of sentences regarding a user's interests, and an interaction history $I_u$, historical utterances $s_1, ..., s_{k-1}$, where $s_k$ is the utterance at the $k-th$ turn, and the corresponding topic sequence $t_1, ..., t_{k-1}$. Given this context, the model aims to: (i) predict the next topic $t_k$ that is as close as possible to the target topic (topic prediction), (ii) recommend the a movie item $i_k$ (item recommendation), (iii) generate a response $s_k$ about the topic (response generation).

Again, a new dataset was collected entitled *Topic Guided Recommendations through Dialogue (TG-REDIAL)* using the same style of two-party recommendation dialogues as REDIAL, but this time with topic threads, which guide the seeker from a non-recommendation to a recommendation scenario through a sequence of topics in a more friendly chit-chat conversation style. Also, instead of collecting full dialogues, they followed a semi-automatic controllable annotation method to link user ids with real users from a website.

**UPCR** Another example of CRS model is *User Preference Conversation Recommender (UPCR)* (Ren et al., 2022). The assumption behind this model is that topic tracking is not sufficient to capture users' preferences in a dialogue. Instead, the authors propose a more detailed process for recognising and maintaining user's preferences at different representation levels. Specifically, to account for the fact that CRS only track the user's short-term feedback (during the dialogue), the limited annotations, and the complex semantic relations among items, they introduce an end-to-end variational reasoning approach to separately model long-term preferences and short-term preferences as latent variables with topical priors. Notably, the model uses an encoder-decoder representation architecture to encode text sequences and generate response sequences for both short-term and long-term input representations. Importantly, a policy network is used to predict topics that lead to either a clarification question or recommend an item.

UPCR used both the REDIAL and the TG-REDIAL datasets for the purpose of topic-guided recommendation and produced more accurate results compared to both REDIAL and TG-REDIAL corresponding CRS models.

### 2.2.2 Conversational Image Recommendation

Apart from text-based CRS, another line of research emphasises on retrieving images at each interaction turn during a multi-turn dialogue in the fashion domain. This research setting aims

to simulate an environment of online shopping with a digital assistant or a seller, with a goal to satisfy the shopper as fast as possible and fulfil their needs. This setting can be described as Conversational Image Recommendation and is based on previous research on image retrieval. In this section, we describe the nature of the task, and continue with a number of representative models, training, and evaluation settings.

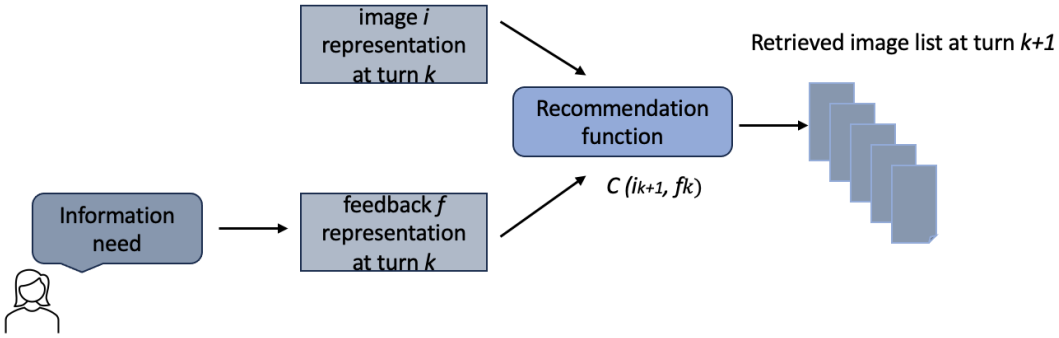



Figure 2.7: Schematic representation of the Conversational Image Recommendation task for the fashion domain. The resulting ranking of images at turn $k+1$ is a result of the feedback and image representation of the candidate item of the previous turn.

### Dialog-based Interactive image Retrieval

Inspired by previous research on image search, visual dialogues and reinforcement learning methods, Guo et al. (2018) proposed a new CRS task, which they called *dialog-based interactive image retrieval*. Motivated by the neural network-based advances in retrieval systems in e-commerce (Huang et al., 2015; Liu et al., 2016) and in web search (Gordo et al., 2016; Jégou et al., 2011), they created a system that accounts for the inconsistencies between feature representations and semantic concepts. Importantly, the user can provide iterative feedback to the system that leads to improvements in CRS performance. Indeed, Guo et al. (2018) based the model's feedback on older systems that allow users to provide feedback on recommended items based on their relevance (how similar or dissimilar they are to the target item) in a binary manner (Rui et al., 1998), or provide relative attribute feedback which compared candidate and target images with a set of fixed attributes (Kovashka et al., 2012). In contrast, the task provides a richer form of feedback than these attribute-based systems which can be provided in natural language, thus allowing the user to more directly express their interest by commenting on the conceptual differences between the retrieved item and what they need. For this purpose, they formulated the task as a reinforcement learning problem that optimises the rank of the target image item, with input as natural language feedback and output as a ranked list of items. Schematically, the task is depicted in Figure 2.10. A conversation $C()$ at each turn $k+1$ is assumed to be determined by the image representation of the candidate item at turn $k$ and then feedback $f_k$ received by the user at turn $k$. The received feedback produces a new list of ranked image items at each turn, while the user only sees the top one. The task is applied on real-world data from the

fashion domain, and in particular shoe retrieval. More generally, this is our task of interest, and we will refer to this task as Conversational Image Recommendation to describe all CRS models used in this section and across the thesis chapters. Below, we describe the basic architecture of the task.

**Model Architecture** The detailed architecture describing the framework of dialog-based interactive image retrieval model is presented in Figure 2.8. It corresponds to a *Dialog Manager* agent which interacts with a user in a multi-turn dialogue. At each turn $t$, the dialog manager presents a *candidate* image $a_t$ to the user drawn from a retrieval database $I = I_{i=0}^{N}$. Based on this image, the user provides a natural language utterance feedback $f_t$, which describes the differences of $a_t$ to the user's desired or *target* image. Based on this feedback and the dialog history $H = a_1, f_1, ..., a_t, f_t$ up to turn $t$, the dialog manager retrieves a new candidate image $a_{t+1}$ from the database to present to the user.

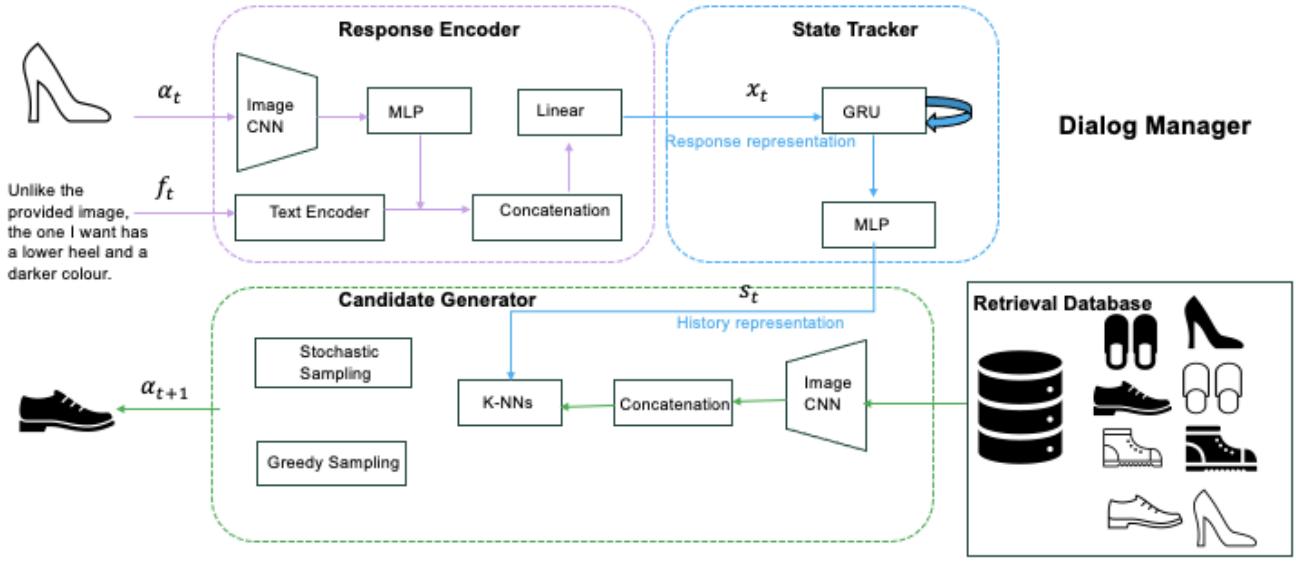



Figure 2.8: Schematic representation of the end-to-end framework for dialog-based interactive image retrieval. Since this model uses a GRU in its State Tracker, we refer to it as GRU in the context of Conversational Image Recommendation. Adapted from Guo et al. (2018).

The main components of Dialog Manager are a *Response Encoder*, which creates an embedding of both $a_t$ and $f_t$ and joins them into a unified input representation $x_t \in \mathbb{R}^D$, a *State Tracker*, which aggregates $x_t$ with the dialog history to produce an updated vector representation $s_t \in \mathbb{R}^D$, and a *Candidate Generator*, which uses $s_t$ to produce a new candidate image $a_{t+1}$. For the task to be successful, $a_t$ is assumed to be similar to $s_t$. More specifically, each component acts as follows:

**Response Encoder** As mentioned above, the Response Encoder encodes the visual-semantic information $a_t, f_t$ at turn $t$ into $x_t \in \mathbb{R}^D$. This is executed with 3 steps: (i) Encode the candidate

image with a deep convolutional neural network (CNN) and subsequently, a linear transformation

$$x_t^{im} = ImgEnc(\alpha_t) \in \mathbb{R}^D \tag{2.13}$$

where the CNN in this case is an ImageNet pre-trained ResNet-101 (He et al., 2016). (ii) Encode the user feedback sentences as one-hot vectors and represent them with a linear projection and a CNN as

$$x_t^{txt} = TxtEnc(f_t) \in \mathbb{R}^D \tag{2.14}$$

(iii) Concatenate image and text vector representations with a linear transformation to obtain a response representation for a given turn

$$x_t = W(x_t^{im} \oplus x_t^{txt}) \tag{2.15}$$

where $\oplus$ is the concatenation function and $W$ is the linear projection.

**State Tracker** The State Tracker receives the input representation $x_t$ from the Response Encoder, combines it with the historical information representation from previous turn, and produces an updated aggregated representation vector $s_t$. This is achieved with a gated recurrent unit (GRU) (Cho, 2014). More formally, its update function can be described as:

$$\begin{cases} g_t, h_t = GRU(x_t, h_{t-1}) \\ s_t = W^s g_t \end{cases} \tag{2.16}$$

where $h_{t-1} \in \mathbb{R}^D$ is the hidden state, $g_t \in \mathbb{R}^D$ is the GRU output, $h_t$ is the updated hidden state, $W^s \in \mathbb{R}^{DxD}$ is a linear projection, and $s_t \in \mathbb{R}^D$ is the updated historical information representation based on information from the current dialog turn. The authors state that the formulation of the State Tracker leads to a memory-based design that sequentially takes into account information from user feedback in order to identify new candidate items.

**Candidate Generator** The purpose of this component is to select a candidate image $\alpha_{t+1}$ to present to the user. In this regard, given the representation of all images in the retrieval database ${x_t^{im}}_{i=0}^{N}$, where $x_t^{im} = ImgEnc(I_i)$ the generator computes a sampling probability that minimises the distance between $s_t$ and each image representation $x_i^{im}$. For this purpose, the model uses a softmax distribution over the top nearest neighbours of $s_t$. To sample an image from the sampling distribution, they use either a stochastic approach during training time or a greedy approach during inference time.

Based on the GRU model used by the State Tracker, for the rest of this thesis, when we speak about the base model in Conversational Image Recommendation, we will refer to it as GRU.

### Training and Evaluating Conversational Image Recommendation Systems

A number CRS models use a reinforcement learning approach to train and evaluate their models. In this section, we first explain why a RL approach is needed, introduce the reader to the concept of user simulation used for this purpose, and then describe the state-of-the-art approaches that were recently proposed as a solution to simulate user behaviour for training and evaluating reinforcement learning-based CRS, and in particular for the task of our interest, namely Conversational Image Recommendation.

**User Simulation in Reinforcement Learning Approaches** In recent years, a number of CRS models have been proposed, which normally use a multi-turn structure that resembles a real conversation. In this regard, the model tries to optimise a dialogue with either a supervised learning (SL) or a reinforcement learning (RL) approach. More specifically, in SL approaches, a policy is used to follow a real user expert actions, which, in turn, requires obtaining data from experts to annotate machine conversations. Most importantly, by following a pre-defined plan of states, some state spaces will be missed during the exploration of dialogue training data (Li et al., 2016). This is the reason why in recent CRS systems, such as the GRU for Image Recommendation described above, a RL policy is preferred, as it allows a system to learn based on reward signals by optimising a policy through its interaction with users (Li et al., 2016). In particular, GRU models the ranking percentile as the environment reward during the learning process with an aim to maximise the sum of discounted rewards as: $\max_\pi \upsilon^\pi = \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t | \pi_\theta]$, where $r_t \in \mathbb{R}$ is the reward for the ranking percentile of the target image item at turn $t$, $T$ is the number of dialogue turns, $\theta$ corresponds to the network parameters, $\pi_\theta$ symbolises the policy, and $\gamma$ is a discount factor for the trade-off between short-term and long-term rewards (Guo et al., 2018).

Still, training a CRS with RL often uses mainly optimises for long-term rewards (Shi et al., 2019), i.e. retrieving the correct item in later turns. Still, such models require an exploration of the action space, which requires access to large amount of training data from real users (Li et al., 2016; Shi et al., 2019). To evaluate CRS, user simulators are used as a surrogate of human users to human behaviour (Li et al., 2016; Shi et al., 2019). For text-based conversational systems, a user simulator was proposed following an agenda-based simulation framework (Schatzmann and Young, 2009), in which a stacked representation of user states encodes the dialogue history and the user goal, and user state updates can be modeled as sequences of push and pop operations with stacks. For image-based CRS, *relative captioning* is an example simulating real user natural language feedback trained on human-annotated dialogues, which we explain below.

**Relative Captioning and Dataset Creation** As mentioned, in RL-based CRS, an extensive amount of data exploration with the environment is needed, and relying on existing dialogue corpora does not solve this problem. In the case of the Conversational Image Recommenda-

tion task, a user simulator is employed called the *relative captioner*, which generates natural language feedback based on the relative visual differences between a candidate item at turn $t$ and the user's target item (Guo et al., 2018). Note that the generated feedback is independent of previous feedback or retrieved images from earlier turns other than the immediate turn. For the purpose of training the relative captioner with simulated data, a dataset was collected with crowd-sourcing via Amazon Mechanical Turk, where participants were asked to describe for each candidate-target image pairs, what the differences are. The problem describes a shopping scenario between a customer and a shopping assistant. The process of the relative captioning dataset collection is described in Figure 2.9. To prompt annotators for relative feedback, a sentence prefix was given as an instruction for responding to a given image pair. This example refers to footwear items from the fashion domain. Note that *relative captioning* is different from *discriminative captioning* previously used (Vedantam et al., 2017) to only describe the target item. Following the collection of the first dataset Guo et al. (2018), which was based on the Shoes dataset (Berg et al., 2010), a more dedicated relative captioning dataset was collected, again in the fashion domain, with multiple clothing fashion categories, namely Dresses, Shirts, and Tops & Tees (Wu et al., 2020). The resulting dataset collection is called FashionIQ and consists of thousands of captions collected for the purpose of training the user simulator of a GRU Conversational Image Recommendation model.

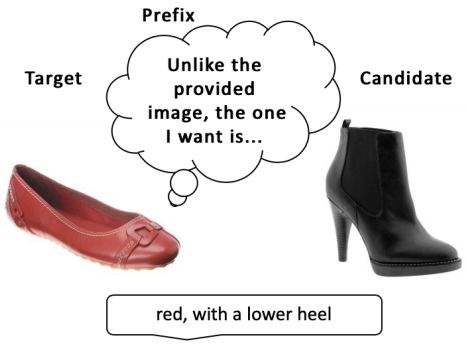



Figure 2.9: Schematic representation of the relative captioning setting. Using a *prefix*, the simulator generates natural language feedback or critiques, which describe the relative visual differences of the target and candidate item at a given turn.

More generally, the process of obtaining training data for the user simulator and their use for training a CRS is schematically depicted in Figure 2.10. As a first step, training data is collected with crowd-sourcing. Then, the collected data is used to train the dedicated CRS component which corresponds to the user modeling system as described in figure 2.5, namely the user simulator. The resulting simulated data simulate the user role in a dialog, which means that each critique that is produced as user feedback actually comes from the simulator to inform the system about user preferences, and could be further used to potentially respond to clarifying questions.

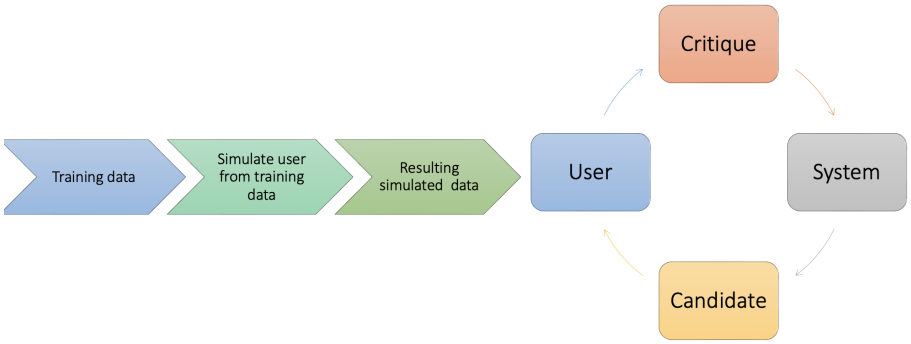



Figure 2.10: Schematic representation of an interaction in Conversational Recommendation Systems.

**Estimator - Generator - Evaluator (EGE) model**

Since the introduction of GRU for image recommendation (Guo et al., 2018), research has attempted to modify the main structure and specific elements of its training policy. For example, Yu et al. (2019) proposed a variation in the number of recommended items; they displayed the top-ranked list to the user instead of only the top item. In particular, they present a list of items to the user at each turn, and the user can provide feedback on items of choice. In this way, they encourage exploration and collect more diverse feedback information on items. Another approach proposed by Wu et al. (2021b) is the Estimator - Generator - Evaluator (EGE) model, which models interactive recommendation as a partially observable Markov decision process (POMDP). As depicted in Figure 2.11, the EGE model consists of three main components, an Estimator that tracks and estimates user preferences, a Generator that recommends candidate items based on estimated states, and an Evaluator which is used to judge the quality of an estimated state at each tme point.

The initial setting of Conversational Image Recommendation with GRU (Guo et al., 2018) is most effective for evaluating short-term interactions. In contrast, the evaluator component of EGE, together with its ability to use both historical feedback and prior recommendations, enhances its performance for long-term interaction satisfaction. In particular, EGE learns a policy that depends on observations but also on action histories (historical feedback and recommendations), and conditions its actions on the entire history. For that reason, and compared to GRU (Cho, 2014; Guo et al., 2018), it maximises longer-term rewards. This functionality has led EGE to improved performance over both GRU variants (Wu et al., 2021b). Across this thesis, GRU and EGE will form the key CRS models that we compare using the Shoes (Berg et al., 2010; Guo et al., 2018) and FashionIQ (Wu et al., 2021a) datasets.

### 2.2.3 Limitations of existing CRS Evaluation Settings

All of the above-mentioned CRS settings (as detailed in Sections 2.2.1 and 2.2.2) share a common ground; they assume that a user's target item is available in the retrieval database or *cat-*

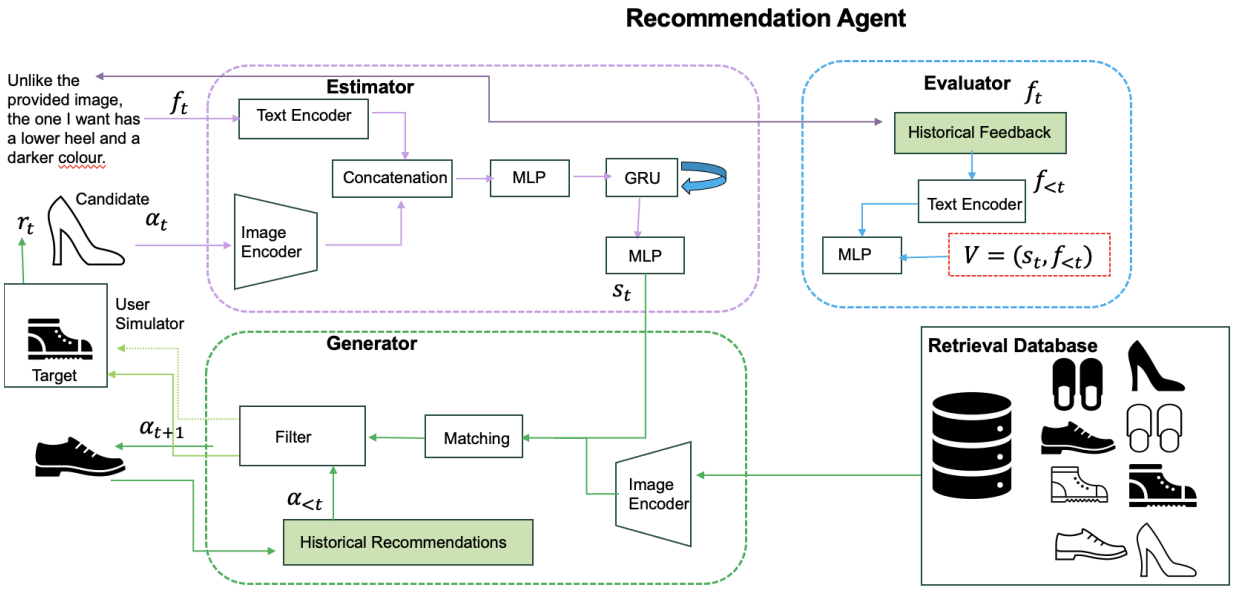



Figure 2.11: Schematic representation of the EGE model. Adapted from Wu et al. (2021b).

*alogue*, and consequently, if a system does not return the user's target by a given rank and at a given turn, this signals the inability of the system to satisfy the user. We call this type of failure *system failure*. This assumption has a number of limitations.

- **Limitation 1a):** In general, the purpose of recommender systems is to facilitate users' exploratory behaviour (Bursztyn et al., 2021), moving beyond what they have already seen or bought. In the case of Matrix Factorisation or Sequential Recommendation algorithms (see Section 2.1.2), user preferences are estimated based on a given user-item prior representation. However, for our task of interest (Conversational Image Recommendation) and CRS in general, such information is obtained during the interaction with the system. Therefore, a system trying to find a single item that is already known by the user contradicts the recommendation intuition.

- **Limitation 1b):** In addition, focusing on a single target item without having any more options to choose from highly restricts system performance. In this way, there is a chance that the system keeps repeating the same recommendations, thus influencing the distribution of items being recommended.

- **Limitation 2a):** In some cases, an item requested by the user is not contained in the item database or catalogue. If a system does not account for this, system failure will be assigned as a reason to a problem that is best described as *catalogue failure*. Specifically, if the database is missing a target item, it will not return it, but this should not be mistaken for a CRS failure. For the purpose of this thesis, we will introduce a new recommendation scenario that corresponds to a missing target item.

- **Limitation 2b):** In addition, the given context of Conversational Image Recommendation does not allow more generalised user satisfaction where the user has a vague information

need that could be satisfied when the system returns the user's target item or another item similar to the original target based on a certain criterion. For the purpose of this thesis, we will introduce a new recommendation scenario that represents a more generalised user need, where the user is more flexible in their choices, can change their mind, and request alternative items.

- **Limitation 3:** A further problem with CRS systems is that they do not try to predict the likelihood of recommendation failures, focusing instead on reporting effectiveness metrics for system performance. In particular, they assume that they should continue the recommendation process regardless of the likely condition of the user (i.e., for an infinite number of feedback turns). For this purpose, we develop a prediction framework that quantifies the extent to which a given target item will be returned by a given turn, and use multiple source information to produce those predictions, together with the corresponding evaluation measures. In this regard, we take into account the multi-turn information in the setting.

We have presented the limitations in the evaluation of CRSs in state-of-the-art models. In the next section, we present the main approaches of the prediction framework we rely on throughout this thesis to build our own prediction framework for CRS, namely Query Performance Prediction (Carmel and Yom-Tov, 2010).

## 2.3 Predicting Query Performance

As mentioned, CRS systems lack an overall failure prediction methodology. In order to predict why a conversation with a CRS might fail, we need to identify indicators that show when the user is unable to find the target item during the interaction. In this regard, we rely on existing work from *Query Performance Prediction (QPP)*. QPP was originally proposed for search engines and predicts the effectiveness of a search result list in response to a query, without having access to relevance judgments (Carmel and Yom-Tov, 2010). In general, there are two types of QPP predictors; pre-retrieval, and post-retrieval. *Pre-retrieval* predictors are used to estimate the performance of queries before the retrieval stage, and therefore, can be considered independent of the search ranking model and the ranked list of results produced by the model (Hauff et al., 2008). This means that pre-retrieval predictors base their predictions on properties of query-terms or corpus-based statistics (Cronen-Townsend et al., 2002; Hauff et al., 2008; He and Ounis, 2004; Mothe and Tanguy, 2005; Scholer and Garcia, 2009; Zhao et al., 2008). Table 2.1 shows the QPP predictors originally proposed for sparse retrieval models such as BM25 and Query Likelihood. Apart from the distinction netween pre-retrieval and post-retrieval predictors, QPPs are further divided into *unsupervised* and *supervised* predictors. Notably, for earlier predictors developed for sparse retrieval models, unsupervised predictors were exclusively used.

With the recent development of pre-trained language models (PLMs) (Devlin et al., 2019; Khattab and Zaharia, 2020; Lin et al., 2020; Xiong et al., 2020), more advanced supervised predictors were proposed that take advantage of the dense embedded representations contained in PLMs. Still, these predictors tend to be correlated mainly with sparse retrieval models. A few attempts have been made to quantify their behaviour or more advanced models, but the conclusions are not consistent (Faggioli et al., 2023b). This thesis is mainly focused on the examination of post-retrieval predictors, since the examination of the contents of the result list can provide richer information that helps in predicting a future ranking than pre-retrieval predictors (Hauff et al., 2008), and also because this allows us to extrapolate to an image-based recommendation list in our task of interest, where examining the content of the result list in one turn can help to make predictions for the result list of the following turn(s). The set of QPPs is presented in Table 2.2. In this section, we provide an overview of the existing pre-retrieval predictors (Section 2.3.1), continue by describing a range of post-retrieval predictors in Section 2.3.2, including unsupervised and more recent supervised predictors, while also briefly mention some early attempts to adapt QPP to a conversational setting, which is our main focus for CRS prediction. Finally, we present a number of limitations in existing research on QPP in Section 2.3.3.

### 2.3.1 Pre-retrieval Query Performance Predictors

*Pre-retrieval* predictors are used to estimate the performance of queries before the retrieval stage, and therefore, are independent of the search performed and the ranked list of results (Hauff et al., 2008). This means that pre-retrieval predictors base their predictions on properties of query-terms or corpus-based statistics (Cronen-Townsend et al., 2002; Hauff et al., 2008; He and Ounis, 2004; Mothe and Tanguy, 2005; Scholer and Garcia, 2009; Zhao et al., 2008). Examples of pre-retrieval predictors that describe the statistical properties of the query terms or the corpus include the *query length* (number of non-stop words in the query), the *query scope*, the *standard deviation of the inverse document frequency* of the query terms, i.e., $\sigma_{idf}$, and two related predictors that measure the relative presence of terms in the query and the collection; the simplified query clarity score *(SCS)*, which measures the occurrence of a query term in the query relatively to its occurrence in the collection, and the *average inverse collection term frequency* (AvICTF), which relates to measuring the divergence of a collection model from a query model (He and Ounis, 2006). Another class of pre-retrieval predictors refers to linguistic features of the queries, such as *syntactic complexity* (distance between syntactically linked words) and *word polysemy* (number of semantic classes a word belongs to) (Mothe and Tanguy, 2005). More recently, a few pre-retrieval predictors were proposed that use the query representations. For example, a group of unsupervised neural pre-retrieval predictors (Arabzadeh et al., 2020; Roy et al., 2019) propose, for example, geometric semantic similarities of query terms, which indicate query specificity or contextual similarity and are based on pre-trained neural embeddings. Still, these predictors are not directly applicable to our task of interets, as they are based

on pre-trained neural embeddings from multi-representation dense retrieval models, and therefore, cannot be applied on a task that uses image-based embeddings.

### 2.3.2 Post-retrieval Query Performance Predictors

*Post-retrieval* predictors, on the other hand, focus on the list of the top-ranked returned documents, and therefore use the relevance scores of the returned items. In this section, we first present the different categories of unsupervised post-retrieval predictors, and then describe a number of supervised post-retrieval predictors.

#### Unsupervised Post-retrieval QPPs

Over the last two decades, a wide variety of unsupervised QPPs have been proposed, including statistical properties, semantic content, and the difference of the result list documents from the corpus. A first group of post-retrieval predictors refers to the difference of the result list from the corpus, or the *focus of the result list*. For example, the *Clarity method* (Cronen-Townsend et al., 2002) measures the focus of the resulting ranking with respect to the corpus using the KL-divergence between their language models, while the *Weighted Information Gain (WIG)* corresponds to the difference between the average retrieval score of the result list and of that of the corpus (Zhou and Croft, 2007). A second group includes the distribution of the retrieval scores of the top-ranked items. Such predictors include *Normalized Query Commitment (NQC)* (Shtok et al., 2009) (the standard deviation of the retrieval scores in the result list). The standard deviation is considered to be negatively correlated with the amount of query drift (the non-related information in the result list) (Mitra et al., 1998). Based on NQC, Roitman et al. (2017b) developed a robust estimator version, which is based on estimating the standard deviation from multiple generated samples of the original result list using bootsrapping. This more robust estimator was found to enhance query performance compared to NQC. Also, the modeling of retrieval scores is another example, since the top-ranked items could be modeled as a certain mixture of distributions corresponding to relevant and non-relevant items (Cummins, 2014). Finally, a simple way to predict query performance is to use the maximum score of the retrieved document list (Roitman et al., 2017a).

A third group of unsupervised post-retrieval predictors examines the coherence of the top-ranked items' embedded representations, which contain TF-IDF vectors. One such related predictor is *autocorrelation* (Diaz, 2007), which assumes that spatially related documents receive similar scores. In addition, a set of *network metrics* (Arabzadeh et al., 2021a) examine the neighbour degree and density of a given retrieved document, and this was found to enhance QPP performance when interpolated with score-based predictors. A low correlation between scores of topically-close documents is assumed to imply a poor retrieval performance. Additionally, another set of coherence-based predictors creates a graph of the most similar documents among

the top-ranked documents (Arabzadeh et al., 2021a), based on their TF-IDF representations. For example, Weighted Average Neighbour Degree (WAND) and Weighted Density (WD) enhanced the performance of score-based predictors by using linear interpolation.

A fourth group of post-retrieval predictors refers to the relation of the top-ranked retrieval scores with a particular reference list, which points to external retrieved documents lists that are found to be either effective or ineffective (Shtok et al., 2016); the stronger the relation with these external lists, the more indication we have about query performance. One example refers to the *utility estimation framework (UEF)* (Shtok et al., 2010), which estimates the utility of a given ranking with respect to how much it represents an underlying information need (Lafferty and Zhai, 2001). The utility is estimated by the expected similarity between a given document ranking and those induced by estimates of relevance language models (these rankings are assumed to be representative of the information need) (Lavrenko and Croft, 2017). A similar predictor to the UEF approach is *query feedback (QF)* (Zhou and Croft, 2007), which measures the overlap of top items between the result list and a reference list retrieved from the corpus using a language model induced from the result list. Autocorrelation (Diaz, 2007) could also fall under this category. For example, if use as a reference to the original result list either a perturbed version of the scores diffused in space or an averaged value from multiple retrievals. A more generalised approach for estimating the effectiveness of a ranking is the assumption that high association with *pseudo-effective* reference lists and low association with *pseudo-ineffective* lists improves effectiveness (Shtok et al., 2016). Lastly, another type of predictor using reference items is the *rank-biased overlap (RBO)*, which measures the expected average overlap between two rankings (Webber et al., 2010).

Finally, only few attempts of post-retrieval predictors have proposed for conversational systems. In this regard, some recent work on QPP in a conversational environment has only addressed Conversational Search. For example, recent work examines the effectiveness of top-retrieved documents for deciding to generate clarifying questions, and specifically some extracted features, such as noun phrases or named entities (Sekulić et al., 2022). Indeed, clarifications are useful for both the user and the system (Aliannejadi et al., 2019; Kiesel et al., 2018; Zamani et al., 2020). More recently, Faggioli et al. (2023b) proposed a QPP evaluation framework for Conversational Search. In particular, they suggested that QPP in conversational search systems should be evaluated in different settings, based on a single-utterance, the previous utterance or the entire dialogue. Still, unsupervised QPP for conversational recommendation has not been addressed in the literature.

In all cases, we can express a QPP function more formally as:

$$\hat{M} \leftarrow \mu(q, D_q, C) \tag{2.17}$$

where $q$ is a query, $C$ is a document corpus, $D_q$ is a list of retrieved documents, $\mu$ is a query performance predictor which produces a metric $M$. In other words, predicting query performance

is a function of the query, the retrieved document list, the corpus, and how the QPP measure is produced.

#### Supervised Post-retrieval QPPs

Supervised predictors use, in general, more complex indicators than unsupervised predictors to predict query performance. Indeed, they might use multiple sources of information or other QPPs. For example, one of the first supervised QPP predictors was Neural-QPP (Zamani et al., 2018), which used a multi-component supervised predictor as the output of existing unsupervised QPP predictors with weak supervision. Also, based on Deep-QPP (Datta et al., 2022a), which used information from semantic interactions between query terms and terms of the top-documents retrieved with it, Datta et al. (2023) go one step further to combine it with a second source of information from learning-to-rank features, which were found as good indicators for query performance (Chifu et al., 2018).

Another group of supervised post-retrieval QPP predictors mainly fine-tune the BERT model's (Devlin et al., 2019) embedded representations in multiple ways. For example, BERT-QPP (Arabzadeh et al., 2021b) fine-tunes BERT by adding an additional cross-encoder or bi-encoder network layer in order to produce a final relevance score per query. NQA-QPP (Hashemi et al., 2019) developed a method for question answering by providing a multi-source supervised score. Extending this, qpp-BERTpl (Datta et al., 2022b) moves to a list-wise approach that moves beyond simply estimating the relevance of the top document. As for supervised predictors that consider a more conversational context, NQA-QPP (Hashemi et al., 2019) developed a method for question answering by providing a multi-source supervised score by also using BERT embeddings in a question answering setup. Apart from BERT-based predictors, Roitman et al. (2019) examined a constrained retrieval setting, such as the interaction with a conversational assistant, where the assistant needs to decide whether the provided answer could be accepted. The authors built a classifier that determines the answer quality by adapting some existing QPPs to the answer level (using the score of the top item, which is provided as the answer).

Again, while a few supervised predictors have been applied to Conversational Search, none of the existing works have addressed CRSs. In addition, while these predictors predict performance at the query level, they do not predict at the conversation level, taking into account how the information is acquired through the sequence of turns.

### 2.3.3 Limitations of existing QPP research

From the above, we observe two limitations of current QPP approaches. First, they have been applied to sparse retrievers (e.g. BM25), which are outperformed by dense retrieval models (e.g. TCT-ColBERT). Indeed, even supervised predictors are applied mainly to BM25, and when applied to more advanced retrievers, their performance fails. More specifically, the limitations

of existing QPPs can be summarised as follows:

- **Limitation 3:** We believe that the problem is that BERT-based QPP predictors do not use the same model and type of representations between QPPs and retrieval model. In addition:
- **Limitation 4a):** While some early attempts have been made to adjust to a conversational setting, they do not take into account the multi-turn nature of the task.
- **Limitation 4b):** At the same time, while these attempts were made on Conversational Search, no one has addressed QPP in a multi-turn recommendation setting

In this thesis, we ail to quantify a system failure by predicting its performance across multiple turns. Note that some of this early attempts to adjust to a conversational search setting are part of a work that was conducted shortly after we first proposed our own conversational framework, and therefore, we do not consider it strictly as background, but rather as parallel work. In the next section, we briefly mention QPP in a conversational context. We are concerned with creating a prediction framework for failed conversations in a recommendation setting.

## 2.4 Evaluation Methods

In Section 2.1, we have describe a range of information seeking tasks, extending from ad-hoc retrieval models for web search to more conversational tasks such as conversational search and recommendation. All these tasks share a common intuition; all of them refer to *ranking task*, where a given set of documents or items are retrieved in a ranked list based on their relevance to a user query or natural language feedback. For the purpose of evaluating ranking tasks, a variety of *evaluation metrics* have been proposed, that aim to address certain aspects of a system's objective performance. In addition, for each of the tasks that we use, there are a number of dedicated datasets on which the tasks are evaluated on. In parallel, for the purpose of the thesis statement (Section 1.2), new datasets were collected, aiming at improving either the nature of the task or the functionality of a system. In this section, we first describe the different evaluation metrics used both for assessing system performance and the strength of an association along the following chapters, and we continue with a summary of all the datasets mentioned across the thesis, both the openly available and the collected datasets.

### 2.4.1 Evaluation Metrics

To evaluate the performance of a system, we can use two main criteria: effectiveness and efficiency. On one hand, *effectiveness* of a system evaluates the ability of a system to retrieve the relevant documents in high rank positions in response to a user's query or CRS feedback, respectively. On the other hand, *efficiency* corresponds to the reduced system time to return

the retrieved document list. For the purpose of this thesis, we will focus on effectiveness of a system, whether it concerns a search engine or a conversational recommendation system. Evaluation with efficiency metrics out of the scope of this project; to a certain extent, we assess how much time it takes to return a target item to a simulated user in Conversational Image Recommendation by counting the number of turns taken to retrieve it.

### Effectiveness Metrics

Before presenting the effectiveness metrics, we introduce some notation. In particular, given a user query $q$, the returned documents in response to this query are returned as a ranked list $R_q$. In practice, not all documents are returned; instead, the top $k$ documents are retrieved and returned as a $R_k(q)$ list. In addition, the set of relevant documents in terms of the degree to which they satisfy a user query can be denoted as $Rel(q)$. In general, the retrieved documents that are also relevant can be found at the intersection of $R_k(q)$ and $Rel(q)$. Based on this, the two main effectiveness evaluation metrics for ranking systems are Precision and Recall (Cleverdon et al., 1966). First, *Precision* is defined as:

$$P(q,k) = \frac{|Rel(q) \cap R_k(q)|}{R_k(q)} \tag{2.18}$$

which measures the proportion of documents (items) out of the list of retrieved documents with a cutoff $k$ are relevant to the query. On the other hand, *Recall* is calculated as:

$$R(q,k) = \frac{|Rel(q) \cap R_k(q)|}{Rel(q)} \tag{2.19}$$

which measures the proportion of documents (items) out of the list of relevant documents with a cutoff $k$ are indeed retrieved in response to the query. Recall-based metrics have a greater focus on retrieving a larger number of relevant documents by risking to retrieve some more irrelevant ones. Still, both precision and recall are top-heavy metrics, in the sense that they focus on the top of a ranking, and therefore, the change in order of other ranks is not influential (Robertson, 2008). For that reason, *Average Precision (AP)* (Harman, 1995) can alternatively be used, since it takes the rank order into account. AP is defined as:

$$AP(q,k) = \frac{|\sum_{i=1}^{k} P(q,i) Rel_i|}{Rel(q)} \tag{2.20}$$

where for each document $i$, $P(q,i)$ is the precision calculated as Equation (2.18), and $Rel_i$ is the binary relevance judgment (1 = relevant, 0 = non-relevant) to the query. Still, this assesses the performance of a single query. Instead, it might be useful to assess the system effectiveness based on a query set $Q$. When averaging over AP for each query in $Q$, *Mean Average Precision*

*(MAP)* (Craswell and Hawking, 2002) can be calculated as:

$$MAP(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} AP(q_j,k) \tag{2.21}$$

As mentioned above, one concern with AP and MAP is that the relevance of documents to a query can be judged is binary (0 or 1). Sometimes, the relevance and the length of the result list can be influenced by specific queries. As a solution for this problem, the *discounted cumulative gain (DCG)* (Järvelin and Kekäläinen, 2002) was proposed, which uses the concept of *graded relevance* extending from non-relevant to highly relevant. Specifically, DCG is defined as:

$$DCG(q,k) = \sum_{i=1}^{k} \frac{2^{Rel_i}-1}{log_2(i+1)} \tag{2.22}$$

where $Rel_i$ denotes the graded relevance of document $i$ in the ranked list of documents (items) and $k$ is the rank cutoff. As mentioned, users tend to examine the higly relevant documents in the top ranks and gradually ignore the documents in lower ranks (Robertson, 2008). Therefore, DCG includes a logarithmic discount factor $log_2$ that assigns higher weights to the relevance of documents that are ranked later in the list and lower weights to documents that are ranked higher. In particular, the logarithmic penalty ensures that when moving down the list, you divide each item's gain by a growing number, (inverse logarithm of the position number). In other words, the discount helps in the diminishing value of relevant items further down the ranking. A further problem that might be caused by DCG is that the different queries have a different length of relevant result list. To account for this, a normalising factor is added in order to enable the comparison across multiple queries (Järvelin and Kekäläinen, 2002). Specifically, the normalised discounted cumulative gain (nDCG) is defined as:

$$nDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} \frac{DCG(q_i,k)}{IDCG(q_j,k)} \tag{2.23}$$

where $IDCG(q_j,k)$ is perfect ranking of a query and is further defined as:

$$IDCG(q,k) = \sum_{i=1}^{REL_k} \frac{2^{Rel_i}-1}{log_2(i+1)} \tag{2.24}$$

where $REL_k$ is the list of relevant documents ordered by their relevance in the corpus up to $k$. Finally, there are cases where a single document might be relevant for a query, or simply a user might be biased to click on the top-ranked result (Craswell et al., 2008). In this regard, *Reciprocal Rank* was proposed, or in other words, the reciprocal of the rank of the first relevant document in the result list. To average over all queries in a query set $Q$, the *Mean Reciprocal*

*Rank (MRR)* is used. Specifically, MRR is defined as:

$$MRR = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{rank_{q_j}} \tag{2.25}$$

where $rank_{q_j}$ is the rank of the top relevant document for query $q_j$ in query set $Q$. Finally, in some cases, it is important to simply know whether the desired item or document is returned by the system. In this case, *Success Rate (SR)* can be used.

For the rest of this thesis, we will be using the above-mentioned metrics for both CRS and ad-hoc retrieval evaluation In particular, for the case of the Conversational Image Recommendation task, we will use nDCG@10, MRR@10 and SR at different cutoffs. In this case, SR indicates whether the target item was returned by the system above or at rank $k$. To examine whether it was the one shown to the user, we are interested in SR@1; if we examine the ranked list, other cutoffs can be used, such as SR@10. For the case of experimental results on QPP (described above), we will be using MAP@100, nDCG@10, and MRR@10. Note that for QPP results, we will not be using the effectiveness metrics independently, but with regard to how well they correlate with a QPP predictor. Also, note that in the context of Conversational Image Recommendation, a query $q_j$ corresponds to a conversation with a target image item $j$ in a relative captioning dataset.

### Correlation for QPP

More specifically, the correlation (or strength of association) is measured between two quantities; the per query effectiveness measure and the per query QPP predictor value. For this purpose, three correlation measures are used. First, the *Pearson's r* correlation (Pearson, 1896) is a measure of the linear relationship between two numeric variables. Let $X$ be a random variable where $x_i$ denotes the per query QPP predictor value, and $Y$ another random variable corresponding to the per query effectiveness measure value $y_i$. Then, Pearson's $r$ is defined as:

$$r = \frac{\sum_{i=1}^{|Q|}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{|Q|}(x_i-\overline{x})\sum_{i=1}^{|Q|}(y_i-\overline{y})}} \tag{2.26}$$

where $\overline{*}$ denotes the average value. Other correlation measures are calculated after $X$ and $Y$ have been ranked transformed to values between 1 and $Q$ (De Winter et al., 2016). For example, Spearman's $\rho$ (Spearman, 1987) random variables and into ranked variables and assesses the monotonic relations based on the rank of the observations (Temizhan et al., 2022). Specifically, $\rho$ transforms $X_i$ and $Y_i$ into ranked variables $r_{xi}$ and $r_{yi}$. Then, for ordinal data (such as two ranked lists measured on an ordinal scale), the scores of one are assumed to be monotonically related to the other with no ties. Denoting $d = r_{xi} - r_{yi}$, both variables receive a rank of $i = 1, ..., Q$, and

$\rho$ can be calculated as:

$$\rho = 1 - \frac{6\sum d_i^2}{Q(Q^2-1)} \tag{2.27}$$

Assessing the strength of association between ordinal data can also be done with Kendall's $\tau$ correlation (Kendall, 1938). In particular, $\tau$ is the least strict correlation measure, as it is distribution-free and makes less assumptions than $r$ and $\rho$. In particular, denoting $C$ as the the number of concordant pairs (how many larger ranks are below a given rank of a ranked list), $D$ as the number of discordant pairs (how many smaller ranks are below a given rank in a ranked list), Kendall's $\tau$ can be calculated as:

$$\tau = \frac{C-D}{Q(Q-1)/2} = \frac{2(C-D)}{Q(Q-1)} \tag{2.28}$$

In all cases, the correlation values range between -1 and 1. Across the thesis, we will be using all three correlation measures. In addition, we will show why assessing the relationship between different quantities should not be exhausted on correlation results.

### 2.4.2 Evaluation Datasets

In this section, we introduce and provide an overview of the datasets used in this thesis. In particular, we separately describe the datasets for the two main tasks we use, namely Conversational Image Recommendation for a number of goals and Query Performance Prediction (QPP) for dense retrieval. For the former, which corresponds to the majority of works in this thesis, we use some recently collected multi-modal datasets of the relative captioning setting. These refer to image items and natural language feedback phrases or sentences, together with some metadata with relevant information about the items. A summary of all datasets along with their corresponding task is presented in Table 2.3. Specifically, the first dataset collected specifically for the purpose of dialog-based interactive image retrieval, which was based on the Shoes dataset (Berg et al., 2010). Following that, Wu et al. (2020) collected the FashionIQ dataset, which aimed to provide a resource for developing dialog-based interactive image retrieval models. Three fashion item categories were selected: Dresses, Shirts, and Tops & Tees.

Specifically, each dataset contains multiple data sources: (i) triples for training and testing the user simulators, where each row has the form of $\langle target, candidate, caption \rangle$, where each relative caption describes the visual differences between the target and candidate images. These were obtained by showing candidate-target pairs to real users with crowd-sourcing. (ii) images of the fashion products that can be used for training and testing recommendation models. (iii) side information (textual descriptions and product meta-data, attribute labels). To select images, the authors used a product review dataset (He and McAuley, 2016) and used the link to the product website contained in the dataset, which in turn allowed to obtain the product information. For example, from the textual information, they used fashion attributes from the title, the

product summary, and product descriptions. While this information is useful, it is not directly used for training and evaluation of a CRS. For the purpose of this thesis, we do not examine side information further.

While these datasets provide useful information for assessing CRS performance in multiple ways, some questions remain open, which we will detail in the following chapters. For the purpose of answering questions to the thesis statement, we collected our own datasets. First, we used the existing relative captioning datasets to produce relevance labels for certain image items. In particular, we used crowd-sourcing and asked participants what items, out of a set of presented images, they would prefer instead of the presented assumed target item they would like to buy. Participants' preferences were noted as relevant in the items they found as sufficient alternatives, and the rest as non-relevant. Therefore, we note for each target, the corresponding candidate and the alternative "targets", which are then used to train an updated user simulator that accepts alternative options. The data collection process is described in more detail in Chapter 5.

As a secondary task, and for the purpose of providing insights to our main task of interest, we conduct QPP experiments using search datasets. For this purpose, we use the corresponding state-of-the-art datasets that are used for dense retrieval models. In particular, we use the *MAchine Reading COmprehension (MSMARCO)* passage ranking corpus dataset (Nguyen et al., 2016), which contains 8.8 million passages extracted from document web pages, where for each training query in the collection, there are on average 1.06 judged relevant passages. To distinguish relevant from non-relevant passages, triplets are used for each training instance with positive (relevant) and negative (non-relevant) passages for each query in the training set. As for query sets, we use two state-of-the-art neural retrieval datasets, namely *TREC Deep Learning Track 2019 and 2020*. In particular, TREC 2019 Deep Learning track (Craswell et al., 2020) contains 43 test queries with an average of 153.4 relevance judgements per query, while the TREC 2020 Deep Learning track (Craswell et al., 2021) contains 54 test queries with 39.26 relevance judgements per query on average.

## 2.5 Conclusions

In this chapter, we have presented the background knowledge related to Conversational Recommender Systems (CRS) and dialog-based interactive image recommendation, as well as predicting query performance from multiple sources.

First, we presented an overview of the various Information Seeking tasks in Section 2.1, ranging from ad-hoc retrieval to conversational search and recommendation, and showed that their common ground is that they are all ranking tasks. We elaborated more on the different CRS models in Section 2.2, the background information for the influential Gated Recurrent Unit (GRU) model in Section 2.2.1, and our setting of interest, namely Conversational Image Recommendation in Section 2.2.2. At the end of Section 2.2, we presented a summary of limitations

of existing CRS models. Then, in Section 2.3, we described the existing query performance prediction (QPP) methods and measures, which will be useful to predict failure in CRS performance. We ended this section by presenting the limitations in current CRS approaches, pointing out that no one has addressed it in CRS and a multi-turn setting. We continued with the included datasets and evaluation metrics used across the thesis in Section 2.4, where we detail both the openly available datasets we use in this thesis and our own collected datasets during the PhD programme.

This thesis focuses on developing a framework for predicting and improving the various types of retrieval failure in Conversation Recommendation Systems in the fashion domain. For this purpose, we examine QPP using state-of-the-art retrieval models in ad-hoc retrieval. In particular, we will examine the coherence of the top-retried items to discover semantic relations that are responsible for improved query performance. We will attempt to use QPPs aligned with the corresponding retrieval models in Chapter 3. This addresses **Limitation 3** (see Section 2.3.3), according to which *BERT-based QPP predictors do not use the same model and type of representations between QPPs and retrieval model*. After having carefully studied QPP in its original context, in Chapter 4, we develop a Conversational Performance Prediction (CPP) framework that takes into account the multi-turn nature of the task, thereby addressing **Limitation 4a)** (*While some early attempts have been made to adjust to a conversational setting, they do not take into account the multi-turn nature of the task.*). We will also and extend our evaluation methodology to a recommendation setting, thus addressing **Limitation 4b)** (*While these attempts were made on Conversational Search, no one has addressed QPP in a multi-turn recommendation setting*).

Then, in Chapter 5, we address **Limitation 1b)** (*Focusing on a single target item without having any more options to choose from highly restricts system performance. In this way, there is a chance that the system keeps repeating the same recommendations, thus influencing the distribution of items being recommended.*) by extending the user preference elicitation. We do this by conducting a user study and ask users about alternative preferences to given target items, and then inform the simulator and system performance estimation. In this way, we also extend the intuition of recommendation scenarios also from a user perspective and examine the user a more flexible shopper that changes their mind in a realistic everyday context. In addition, Chapter 5 addresses **Limitation 1a)** (*A system trying to find a single item that is already known by the user contradicts the recommendation intuition*), thus aiding users' exploratory behaviour, by allowing them to reconstruct their preferences during the interaction and opt for an alternative item that is close to what they see as a current suggestion. Finally, in Chapter 6, we introduce our novel recommendation scenarios, including a missing target scenario, and an alternative user preference scenario. In this way, we address **Limitation 1a)** (*A system trying to find a single item that is already known by the user contradicts the recommendation intuition*, and extend our CPP evaluation setting to the scenarios. To conclude, overall, in Chapters 5 and 6,

we address **Limitation 2a)** (*In some cases, an item requested by the user is not contained in the item database or catalogue. If a system does not account for this, system failure will be assigned as a reason to a problem that is best described as* catalogue failure) and **Limitation 2b)** (*The given context of Conversational Image Recommendation does not allow more generalised user satisfaction where the user has a vague information need that could be satisfied when the system returns the user's target item or another item similar to the original target based on a certain criterion.*), since we propose two novel recommendation scenarios and account for various types of recommendation failure.

Table 2.2: Existing Pre- and post-retrieval Query Performance Predictors, including current state-of-the-art QPPs.

| Type | Predictor | Description |
|---|---|---|
| Pre-retrieval | | |
| statistical | query length (Hauff et al., 2008) | number of non-stop words in the query |
| | query scope (Hauff et al., 2008) | relates to the ambiguity of a query |
| | $\sigma_{idf}$ (Hauff et al., 2008) | standard deviation of the inverse document frequency of the query terms |
| | SCS (Hauff et al., 2008) | occurrence of a query term in the query relatively to its occurrence in the collection |
| | AvICTF (Hauff et al., 2008) | divergence of a collection model from a query model (before retrieval) |
| linguistic | syntactic complexity (Mothe and Tanguy, 2005) | distance between syntactically linked words |
| | word polysemy (Mothe and Tanguy, 2005) | number of semantic classes a word belongs to |
| neural | neural specificity (Arabzadeh et al., 2020) | geometric relations between terms in the embedding space, capturing term semantics |
| | $P_{Clarity}$ (Roy et al., 2019) | ambiguity of each query term by estimating the number of 'senses' of each word |
| Post-retrieval | | |
| focus | Clarity (Cronen-Townsend et al., 2002) | KL-divergence between ranking and corpus language models |
| | WIG (Zhou and Croft, 2007) | difference between the average retrieval score of the result list and that of the corpus |
| score-based | NQC (Shtok et al., 2009) | standard deviation of the retrieval scores in the result list |
| | mixture of distributions (Cummins, 2014) | retrieval scores as mixture of relevant and non-relevant items |
| | RSD (Roitman et al., 2017b) | bootstrap-based robust standard deviation |
| | MAX (Roitman et al., 2017b) | maximum score |
| coherence | Autocorrelation (Diaz, 2005) | KL-divergence between ranking and corpus language models |
| | WAND, WD (Arabzadeh et al., 2021a) | difference between the average retrieval score of the result list and that of the corpus |
| reference list | UEF (Shtok et al., 2010) | utility of ranking based on similarity with relevance language models |
| | QF (Zhou and Croft, 2007) | overlap of top items between result list and list from the corpus using language models induced from result list |
| | autocorrelation (Diaz, 2007) | original result list with either a perturbed version of the scores diffused in space or an averaged value from multiple retrievals |
| | pseudo-(in)effective lists (Shtok et al., 2016) | ranking effectiveness based on pseudo lists |
| | RBO Webber et al. (2010) | expected average item overlap between two rankings |
| supervised | Neural-QPP (Zamani et al., 2018) | multi-component supervised predictor as the output of existing unsupervised QPP predictors with weak supervision |
| | BERT-QPP (Arabzadeh et al., 2021b) | BERT fine-tuning with an additional cross-encoder or bi-encoder network layer to produce a relevance score |
| | qpp-BERTpl (Datta et al., 2022b) | BERT fine-tuning with a list-wise approach, training supervised model in chunks of documents |
| | NQA-QPP (Hashemi et al., 2019) | multi-source supervised score using BERT embeddings for QA |
| | Deep-QPP Datta et al. (2022a) | information from semantic interactions between query terms and terms of the top-documents retrieved with it |

Table 2.3: Summary statistics of the relative captioning datasets used for training and evaluation of Conversational Image Recommendation systems. Each dataset contains a number of target-candidate pairs together with a caption, and a number of image items.

| | Shoes | | Dresses | | Shirts | | Tops & Tees | |
|---|---|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test | Train | Test |
| Triples | 10,751 | - | 11,970 | 4,034 | 11,976 | 4,096 | 12,054 | 3,924 |
| Images | 10,000 | 4,658 | 7,182 | 2,454 | 8,555 | 2,966 | 8,387 | 2,808 |

# Chapter 3

# Coherence-based Query Performance Prediction

In Chapter 1, we introduced our task and setting of interest, which is centered on Conversational Recommendation in the fashion domain. This setting mimics an online shopping scenario, where a customer interacts with a shopping assistant and provides feedback about how each recommended item is relevant to their desired item. Still, as we described in Section 1.1, existing evaluation methodologies in fashion CRS do not account for whether or the extent to which a system fails. In addition, current systems do not provide an explanation about specific indicators of performance, and thus, do not aid failure prediction (see also Section 2.2.3). For that reason, it is important to provide a general framework of CRS performance, together with an evaluation methodology that describes the factors that are responsible for that performance. In this regard, we are inspired by a previously proposed methodology initially used for search tasks, namely Query Performance Prediction (QPP) (Carmel and Yom-Tov, 2010). More specifically, in this chapter, we experimentally test the first hypothesis of the thesis statement: *Initially, we can predict the effectiveness of a ranking of textual items for a textual query, by examining the coherence of the top-retrieved items based on their dense embedded representations.* This addresses **Limitation 3** *BERT-based QPP predictors do not use the same model and type of representations between QPPs and retrieval model*. Indeed, as mentioned in Section 2.1, Conversational Recommendation belongs to the family of ranking tasks, which implies that insights drawn from QPP in search tasks can also guide the prediction of CRSs by developing the corresponding indicators of performance. In particular, Conversational Image Recommendation, a subset of CRS tasks, is also based on dense retrieval, in the sense that both text-based critiques and image-based recommendation lists (both text and images) can be represented at the embedding space. Therefore, drawing inspiration from a task originally developed for search systems using only text, we inform recommendation systems that contain both text and images. In this way, we predict the multi-turn rankings of another form of dense retrieval. Specifically, we do this by proposing QPPs specifically designed for dense retrieval, examining the relations of the

embeddings of the top-retrieved items, and show how those can be indicative of query performance. After testing our proposed predictors in their original search setting, we evaluate their usefulness for CRS in Chapter 4.

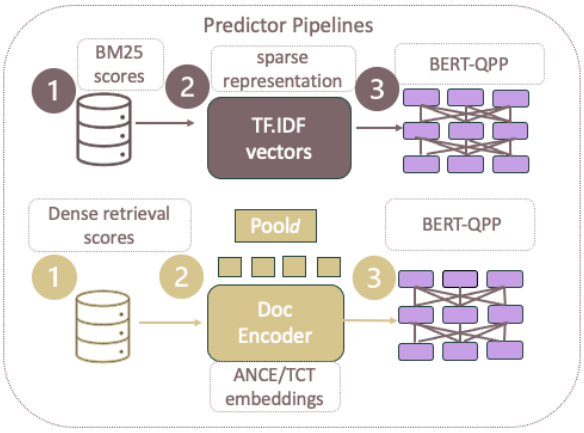



Figure 3.1: Schematic representation of recent QPP pipelines, together with our proposed approach (Step 2, bottom). Top: A BM25 ranking consisting of TF.IDF vector representations (Step 2) (Arabzadeh et al., 2021a; Diaz, 2007), and fine-tuning BERT-based models on top of existing rankings (Step 3) (Arabzadeh et al., 2021b; Datta et al., 2022b; Hashemi et al., 2019; Zamani et al., 2018). Bottom: Dense retrieval ranking with dense embedded representations. Numbers denote each step in the pipeline.

As we mentioned in Section 2.3, *Query Performance Prediction (QPP)* aims to predict the effectiveness of a search result in response to a query without having access to relevance judgments (Carmel and Yom-Tov, 2010). Indeed, retrieval effectiveness in search engines can vary across different queries (Harman and Buckley, 2004; Voorhees et al., 2003). Being able to accurately predict the likely effectiveness of a search engine for a given query may facilitate interventions, such as asking the user to reformulate the query (Belkin et al., 2001; Lioma and Ounis, 2008; Rieh et al., 2006; Wang et al., 2020). In the last two decades, a number of *query performance predictors* have been proposed, which can be grouped in two main categories: *Pre-retrieval* predictors (introduced in Section 2.3.1) estimate query performance using only linguistic or statistical information contained in the queries or the corpus (Hauff et al., 2008; He and Ounis, 2004; Mothe and Tanguy, 2005; Scholer and Garcia, 2009; Zhao et al., 2008). On the other hand, *post-retrieval* predictors (introduced in Section 2.3.2) use the relevance scores or contents of the top returned documents, by measuring, for example, the focus of the result list compared to the corpus (Cronen-Townsend et al., 2002; Zhou and Croft, 2007), or the distribution of the scores of the top-ranked documents (Cummins et al., 2011; Pérez-Iglesias and Araujo, 2010; Roitman et al., 2017b; Shtok et al., 2009; Tao and Wu, 2014). Predictors based on NQC (Shtok et al., 2012) (the standard deviation of relevance scores) have been found to be surprisingly accurate. A further group of predictors that are of particular interest for our purpose, as outlined in the thesis statement (Section 1.2) examine the pairwise similarities among the retrieved documents (Arabzadeh et al., 2021a; Diaz, 2007). Still, a problem with these pre-

dictors (as described in Section 2.3.3) is that thus far, they have been applied using traditional bag-of-words representations. While examining the coherence between returned documents is useful, as we show, these representations are not suitable for predicting the query performance of more advanced retrieval methods.

Indeed, in Section 2.1.1, we mentioned that more recently, pre-trained language models (PLMs) have introduced neural network architectures that encode the embeddings of queries and documents (Devlin et al., 2019; Khattab and Zaharia, 2020; Lin et al., 2020; Xiong et al., 2020), and have led to increased retrieval effectiveness. Often, a BERT-based model is trained for use as a reranker of the result retrieved by (e.g.) BM25 (Robertson and Walker, 1994) - such *cross-encoders* include BERT_CLS (Nogueira and Cho, 2019) and monoT5 (Nogueira et al., 2020). On the other hand, *dense retrieval approaches* (Karpukhin et al., 2020; Xiong et al., 2020) are increasingly popular, whereby embedding-based representations of documents are indexed, and those with the similar embeddings to the query are identified through nearest-neighbour search (e.g. ANCE (Xiong et al., 2020), TCT-ColBERT (Lin et al., 2020), see Section 2.1.1). Compared to reranking setups, dense retrieval is attractive as recall is not limited by the initial BM25 retrieval approach, and improvements in the PLM can improve all aspects of the retrieval effectiveness. Therefore, dense retrieval models inspire us to develop predictors that are effective for predicting their rankings.

In parallel, neural architectures have also been adopted as methods for predicting query difficulty. As briefly mentioned in Section 2.3.2, these post-retrieval methods are *supervised*, and use refined neural architectures in order to produce a final performance estimate (Arabzadeh et al., 2021b; Datta et al., 2022b; Hashemi et al., 2019; Zamani et al., 2018). For instance, BERT-QPP (Arabzadeh et al., 2021b) fine-tunes BERT (Devlin et al., 2019) embeddings for QPP by estimating the relevance of the top-ranked document retrieved for each query. However, its performance is lower or outperformed by unsupervised predictors when using advanced retrieval methods and the TREC Deep Learning datasets Faggioli et al. (2023b). In our view, the problem lies in the mismatch of representations between predictor and ranking, which is best described in Figure 3.1. On top, we see the pipeline resulting from a BM25 ranking, and, at the bottom, a ranking from a dense retrieval system (Karpukhin et al., 2020; Xiong et al., 2020). While BERT-based QPP techniques can be used to predict the effectiveness of BM25 (Arabzadeh et al., 2021b; Datta et al., 2022b; Hashemi et al., 2019; Zamani et al., 2018), single-representation dense retrieval models already contain representations that can accurately predict their corresponding ranking, thus eliminating the need to apply step 3 (BERT-QPP). Instead, to create predictors applicable for dense retrieval, we could use the existing embedded representations of dense retrieval models, as shown in step 2. Indeed, by considering patterns among the embeddings of the retrieved documents, we can update existing unsupervised predictors from traditional sparse (Arabzadeh et al., 2021a; Diaz, 2007) to dense representation-based.

At the same time, the selection of evaluation measure can have an impact on the conclusions of QPP experimental results. In Section 2.4.1, we introduced the main evaluation metrics used

to measure the effectiveness of a ranking system; MAP, NDCG, and MRR. We explained that a common practice in QPP experiments is to correlate a list of QPP predictor values for each query with the corresponding list of evaluation metric values. In our view, the reported metric is a result of researchers' choices. In other words, due to space or time limitations, a researcher might opt for reporting the correlations with a subset of evaluation metrics or simply one of them. This limits the conclusions drawn from these experiments. This observation becomes more prominent if we consider, for example, that unsupervised QPP predictors such as NQC (Shtok et al., 2009) were primarily optimised for MAP at deeper cutoffs (100 or 1000); on the other hand, more recent supervised predictors were either optimised for RR@10 (Arabzadeh et al., 2021b; Hashemi et al., 2019) or used both NDCG@10 and RR@10 (Datta et al., 2022b) providing comparable results between the two measures, but in both cases, results for MAP were missing. As a result, it is impossible to provide insights that are fully generalisable, as missing to report either of them can lead to biased results and incomplete conclusions. In this regard, researcher's choices should be acknowledged and accounted for when drawing insights about the scope of QPP results. Moreover, we believe that designing experimental studies should be aligned with the idea that the different measures are not interchangeable, and that proposed predictors could be complemented with the case where the predictor fails, together with the explanation of the reasons why this happens.

One explanation for why a QPP predictor fails could be that query performance is further mediated by query categorisation. To this point, only a few works have examined how QPP varies with query categories (Carmel et al., 2006; Faggioli et al., 2021a). Indeed, knowing which queries are more difficult to answer may inform us about how to develop more refined predictors. Recently, a query taxonomy was proposed (Bolotova et al., 2022), where the identified question categories were placed in a labelled dataset together with a classifier that enables researchers to apply this categorisation to other datasets. In this work, certain question categories were found to be more difficult to answer compared to others, in particular the questions belonging to Debate, Experience, and Reason categories. In addition, to complement the non-factoid types of questions, a group of factoid questions was extracted and added to the resulting dataset together with a "not-a-question" type, indicating queries submitted to web searches without a question intent. Since this dataset and classifier are relatively new, there has been no attempt to examine how it affects the prediction of query performance. Still, the original study used datasets from TREC and MSMARCO to check the distribution of questions, which directly relates with QPP research especially for advanced retrieval models. Therefore, in this Chapter, we also quantify the extent to which query categories are responsible for the unstable performance of QPPs across different evaluation measures.

In short, our contributions for this Chapter can be summarised as follows:

- We predict the effectiveness of rankings created by single-representation dense retrieval models (ANCE & TCT-ColBERT), emphasising the differences with sparse retrieval mod-

els more traditionally used in QPP research.

- We propose a number of embedding variants of existing unsupervised coherence-based predictors that employ neural embedding representations and our own extension *pairRatio*, an unsupervised predictor which uses pairwise relations of embedding vectors. In this way, we create predictors specifically designed for dense retrieval.
- We study existing predictors and our own proposed predictors to two state-of-the-art single-representation dense retrieval models, namely ANCE (Xiong et al., 2020) and TCT-ColBERT (Lin et al., 2020), as well as BM25 and show that changing the representations increases performance significantly not just for dense but also sparse retrieval.
- We conduct an extensive study by using all three evaluation metrics currently used for QPP, and highlight when each predictor behaves under each measure.
- By also comparing with supervised predictors, we show that applying a BERT-based model for dense QPP is an unnecessary step in the pipeline that decreases QPP performance.
- Going deeper, we select the most representative and best performing predictors to study the importance of differences among predictors and query types (resulting from applying the recent query classifier (Bolotova et al., 2022)) on query performance.
- Consequently, we apply multilevel statistical models (Curran et al., 1997; Field et al., 2012; Maxwell et al., 2017; Singer and Willett, 2003) in QPP to quantify the relationship between query categorisation and the unstable QPPs. In our analyses, we measure the performance of different QPPs in relation to the total QPP variation that can be attributed to the categorisation or as we term *query types*. At the same time, we detect a unique sensitivity of dense retrieval methods, which are affected by query type (up to 35% increase in query performance variations due to query categorisation) and exhibit larger differences between predictors, a pattern which is not apparent in sparse retrieval.
- We obtain insights from a number of coherence-based predictors on state-of-the-art retrieval models and datasets to inform our task of interest for conversational recommendation, taking advantage of the embedding-based nature of contemporary CRS models (Guo et al., 2018; Wu et al., 2021a).

The findings of this Chapter can be summarised as follows: (i) Using coherence-based unsupervised predictors can sufficiently predict dense retrieval models for many of the examined contexts, and although we are inspired to predict dense retrieval, they are also effective for BM25. (ii) Our proposed predictors provide the highest correlations for the more precision-oriented NDCG@10 for all retrieval models, while NDCG@10 and MRR@10 provides similar

results. (iii) In our experiments on the TREC Deep Learning Track datasets, we demonstrate improved accuracy upon dense retrieval using the dense versions of coherence predictors (up to 92% compared to sparse variants for TCT-ColBERT and 188% for ANCE). (iv) Our multi-level perspective proposes a solution to correlation instabilities between measures, by showing how the interplay with query types differently influences each of the measures. In other words, we provide an analytical point that can explain any predictor, and show how our proposed predictors mainly optimise the measure that is less influenced by query variations. Using existing distribution-based evaluation QPP measures and a particular type of linear mixed model, we find that query types further significantly influence query performance (and are up to 35% responsible for the unstable performance of QPP predictors), and that this sensitivity is unique to dense retrieval models. (v) In particular, we find that in the cases where our predictors perform lower than score-based predictors, this is partially due to the sensitivity of MAP@100 to query types. Our novel analysis provides new insights into dense QPP that can explain potential unstable performance of existing predictors and outlines the unique characteristics of different query types on dense retrieval models.

The structure of the rest of this Chapter is as follows: We present related work in Section 3.1, and present our new extended predictors in Section 3.2. Then, we follow with traditional correlation analysis of QPP predictors in Sections 3.3 and 3.4, continue with an extended linear mixed model analysis to test for query type in Section 3.5, and conclude with some final remarks in Section 3.6.

## 3.1 Related Work on Existing QPP Predictors

In this Chapter, we focus our attention on post-retrieval QPPs, since, as we mentioned in Section 2.3.2, they are in general more accurate than pre-retrieval QPPs (Hauff et al., 2008). Contrary to Section 2.3.2, in this section, we place our focus more on the QPPs that can be more easily applied on dense retrieval models and elaborate on them, while explaining how they differ from our proposed predictors. There are two main reason why we eliminate pre-retrieval predictors from our focus. First, existing unsupervised neural pre-retrieval predictors (Arabzadeh et al., 2020; Roy et al., 2019; Saleminezhad et al., 2024) propose, for example, geometric semantic similarities of query terms, which indicate query specificity or contextual similarity and are based on pre-trained neural embeddings. Since these predictors examine queries at the token-level, they are not applicable to single-representation dense retrieval. Second, information based on queries can, in general, provide quite limited information with respect to the effectiveness of the ranking.

Apart from pre-retrieval predictors, we further eliminate a number of post-retrieval QPPs from our analysis and scope, which mainly refer to term-based relations. In particular, we refer to earlier post-retrieval predictors (already introduced in Section 2.3.2) that examined the focus on the result list induced by language models (probability distributions of all single

terms) (Cronen-Townsend et al., 2002). For example, *Clarity* (Cronen-Townsend et al., 2002) measures the divergence of the language model of top-ranked documents from the one of the corpus(irrelevant list) - the higher the divergence, the better the performance. *Utility Estimation Framework (UEF)* (Shtok et al., 2010) uses pseudo-effective reference lists induced by term probability-based language models and estimates their relevance using predictors such as NQC (see below in Section 3.1.1). Both of these rely upon term probabilities, and are, therefore, not feasible for extending our predictions to dense retrieval. Also, *Query Feedback (QF)* Zhou and Croft (2007) refers to the overlap of the returned documents with those obtained after applying pseudo-relevance feedback - yet, pseudo relevance feedback approaches for dense retrieval are still in their infancy (Wang et al., 2021; Yu et al., 2021), so we do not consider QF further.

In the remainder of this section, we discuss the main types of query performance predictors that could be applied to dense retrieval, specifically score-based unsupervised predictors (Section 3.1.1) and document representation-based predictors (Section 3.1.2).

### 3.1.1 Score-based QPP

Score-based predictors encode certain assumptions about how the scores should be distributed for high or low-performing queries. Expanding on the overview provided in Section 2.3.2, in this section, we detail more on the different variants of the standard deviation and show how this is an easy way to predict query performance. Firstly, a simple predictor might be the *Maximum Score* among the retrieved documents (Roitman et al., 2017a) - the higher the maximum score, the more confident the retrieval system is that it has found a document that matches well the query. For a more detailed description of score-based predictors, please refer to Section 2.3.2. The most commonly applied score-based predictor is *Normalised Query Commitment (NQC)* (Shtok et al., 2009), based on the standard deviation of the retrieval scores. Several variations of NQC have been proposed that further enhance its performance (Cummins et al., 2011; Pérez-Iglesias and Araujo, 2010; Roitman et al., 2017b; Tao and Wu, 2014). Out of all variations, the most relevant for our purpose is *Robust Standard Deviation estimator (RSD)* (Roitman et al., 2017b), which estimates a more robust version of variance with bootstrapping and therefore, extends NQC results to multiple contexts (each with a bootstrap sample) representing a population of scores (Roitman et al., 2017b). Score-based predictors, which correspond to step 1 in Figure 3.1, are easily applicable to dense retrieval, since scores are computed by each retrieval method.

### 3.1.2 Document Representation-based QPP

Predictors based on document representations (Arabzadeh et al., 2020, 2021a,b; Datta et al., 2022b; Diaz, 2007; Faggioli et al., 2023a; Hashemi et al., 2019; Roy et al., 2019; Saleminezhad et al., 2024) capture semantic relations either between queries, documents, or their interaction (Devlin et al., 2019; Lin et al., 2020), which makes them particularly important for examination in a dense retrieval setting - we discuss unsupervised and supervised predictors below.

### Unsupervised Coherence Predictors

In general, effective unsupervised predictors that consider document representations are preferable, since they require less computation than supervised predictors. Expanding on the brief introduction of coherence-based predictors in Section 2.3.2, in this section, we detail the ones that can explain semantic relations among retrieved documents and explain why we think they need to be updated for dense retrieval, before we demonstrate how we update them in Section 3.2. One predictor that examines the lexical representations of documents is *spatial autocorrelation* (Diaz, 2007), which considers the spatial proximity of lexical document representations, by using their pairwise TF.IDF-based similarities to produce a new set of scores "diffused in space". The final predictor is obtained by correlating the original scores with the diffused scores. Indeed, a low correlation between scores of topically-close documents is assumed to imply a poor retrieval performance.

Importantly, another main family of relevant coherence-based predictors creates a graph of the most similar documents among the top-ranked documents (Arabzadeh et al., 2021a), based on their TF-IDF representations. In Section 2.3.2, we introduced Weighted Average Neighbour Degree (WAND) and Weighted Density (WD), which enhanced the performance of score-based predictors after linear interpolation. One limitation with these predictors (applied step 2 in Figure 3.1, top) is that they were proposed for sparse document representations and have not previously been applied to dense embedded representations.

### Supervised & Neural Predictors

In general, supervised models for QPP can be attractive due to the varying sources of indicators for inferring query performance (Roitman et al., 2017a). At the same time, they are computationally complex compared to unsupervised predictors. For example, Neural-QPP (Zamani et al., 2018) is a multi-component supervised predictor as the output of existing unsupervised QPP predictors with weak supervision - we can think of this as a neural supervised aggregation predictor. Unlike our more general introduction of supervised predictors in Section 2.3.2, in this section, however, we focus more on transformer-based supervised predictors in order to compare their performance on dense retrieval models with simpler unsupervised predictors and to show how we instead use embeddings already contained in dense retrieval models. As an example of such supervised predictors, more recently, BERT-QPP (Arabzadeh et al., 2021b) fine-tunes a BERT model for the QPP task by adding cross-encoder or bi-encoder layers that estimate an effectiveness measure (e.g. NDCG) based on the contents of the top returned document in response to the query. While BERT-QPP can also be applied to the dense retrieval rankings, it uses a different model to that used by the dense retrieval approach itself. Out of the two BERT-QPP variants, the bi-encoder version is closer to the intuition of single-representation dense retrieval. Finally, qppBERT-PL (Datta et al., 2022b) adds an LSTM network on top of the BERT representation to model both document contents and the progression of estimated relevance in the ranking. Compared to BERT-QPP, this approach has promise as it considers more information than just the top-ranked document.

To summarise, existing predictors have either focused on sparse document representations or retrieval scores on the unsupervised side, or have introduced neural pre-trained architectures to create more complex supervised predictors. However, no work has addressed unsupervised predictors using dense embedded representations, as are readily available in dense retrieval configuration. Instead, we argue that by using simple predictors that consider document representation resulting from dense models (step 2 of Figure 3.1, bottom), we can accurately predict effectiveness without the need for supervised cross-encoder-based methods (step 3). In the next section, we detail existing predictors that can be applied to dense retrieval.

## 3.2 Coherence Predictors for Dense Retrieval

In this section, we first describe some existing sparse coherence-based predictors in Section 3.2.1, and then show how these can be adapted to be better suited for dense retrieval settings in Section 3.2.2.

### 3.2.1 Sparse Coherence-based Methods

Sparse coherence-based predictors include spatial autocorrelation (Diaz, 2007) and a number of network metrics (Arabzadeh et al., 2021a). Below, we see their definitions as originally proposed.

#### Spatial Autocorrelation (AC)

First, consider $d$ to be a document's TF.IDF vector. Then, the inner product of two documents at ranks $i$ and $j$ is given by $sim(d_i, d_j)$. We can obtain a pairwise similarity matrix among $k$ top-ranked documents as follows:

$$W = \begin{bmatrix} sim(d_{11}) & sim(d_{12}) & ... & sim(d_{1k}) \\ ... & ... & ... & ... \\ sim(d_{k1}) & sim(d_{k2}) & ... & sim(d_{kk}) \end{bmatrix} \tag{3.1}$$

where $k$ is the cutoff number of the top-k documents. For brevity of notation, let $sim(d_{ij}) = sim(d_i, d_j)$. Projecting (multiplying) each element of the matrix $W_{ij}$ on the vector of the original retrieved scores, $Score(\vec{d})$, we can obtain a new list of *diffused* scores as:

$$Score(\tilde{d}) = W \cdot Score(d) \tag{3.2}$$

Thereafter, an estimate of the spatial autocorrelation (AC) (Diaz, 2007) is obtained by using the Pearson correlation between the two vectors:

$$AC = corr(Score(\tilde{d}), Score(d)) \tag{3.3}$$

which quantifies the relation between the initial and diffused scores. Indeed, as mentioned above, a low correlation between the original retrieval scores (i.e. $Score(d)$) and those weighted by their topical similarity (the diffused scores, $Score(\tilde{d})$) was found to imply poor retrieval performance (Diaz, 2007).

### Network Metrics

As mentioned above, the matrix $W$ represents all pairwise similarities between the top-retrieved documents. This matrix is equivalent to a fully connected network, where each node $\mathbf{V_G}$ corresponds to the $d$ TF.IDF vector, and each edge $\mathbf{E_G}$ corresponds to each entry $sim(d_{ij})$ (Arabzadeh et al., 2021a), or more formally $\mathbf{G}(q, D_q^{(k)}) = \{\mathbf{V_G}, \mathbf{E_G}, W\}$. In this regard, to avoid all edges being considered equal without attention to the edge weight, the network is further pruned via thresholding (Christophides et al., 2015), where the similarities higher than the mean similarity value are selected as neighbours[1].

Consequently, we have the following definitions, which correspond to some recently proposed network metrics (Arabzadeh et al., 2021a) for QPP:

$$AverageNeighbourDegree(AND) = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{1}{|N_{d_i}|}\sum_{j\in N_{d_i}}|N_{d_j}|\right) \tag{3.4}$$

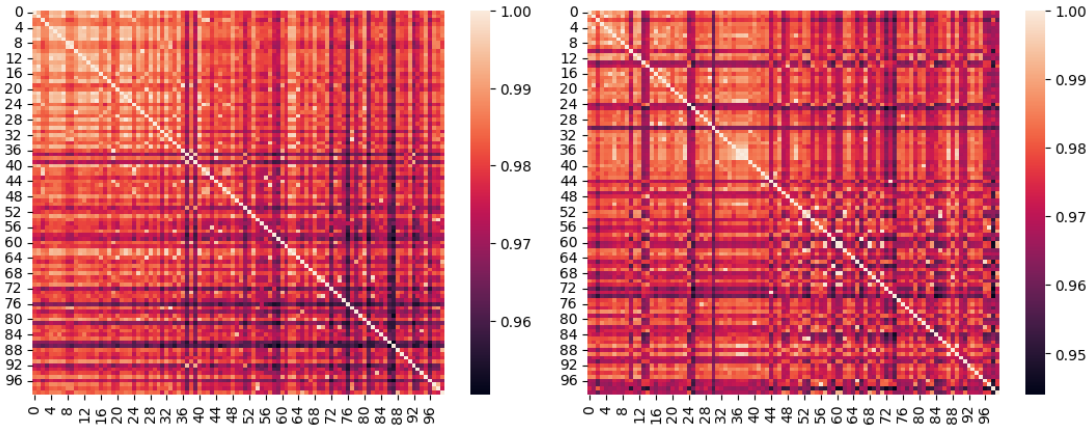
where $N_{d_i}$ is the neighbourhood of document $d_i$. Typically, Equation (3.4) is applied on the pruned graph that only contains edges between the most similar documents, and hence corresponds to the more accurate *Weighted AND* (WAND) measure (Arabzadeh et al., 2021a).

Another way to think about coherence is to count the observed edges or similarities over the set of all possible edges. This results in the Density measure, as follows:

$$Density(D) = \frac{2|\mathbf{E_G}|}{|\mathbf{V_G}|(|\mathbf{V_G}|-1)} \tag{3.5}$$

In short, a higher neighbourhood degree and a higher density of a graph network indicates a more coherent set of top-retrieved results. The general intuition behind these measures is that the presence of coherence, as reflected by highly similar documents in a top-retrieved set indicates the ability of the retrieval method to distinguish relevant from non-relevant documents, and therefore, return the relevant ones at the top of the list.

[1] In this Chapter, we use the definitions for two of the metrics described in the original paper.



(a) Effective Query (b) Ineffective Query

Figure 3.2: Heatmap of pairwise similarity matrix of the top-100 TCT-ColBERT document embeddings for returned for the best (query id 104861 with NDCG@10=1) and worst performing queries (query id 489204 with NDCG@10=0.189) from the TREC DL 19 queryset.

### 3.2.2 Dense Coherence-based Methods

We now derive the embedding representation variants of the above predictors in order to make them suitable for the prediction of neural dense retrievers. We first create the variants for embedding-based AC and network metrics, and then introduce a new variant that extends AC by considering rank groupings.

#### AC-embs

Let $\phi_d$ and $\theta_q$ respectively represent the dense embedded representation of a document and a query. Firstly, we adapt autocorrelation, such that instead of TF.IDF vectors we consider the embedded document representations. Let the inner product of two documents at ranks $i$ and $j$ (with embeddings $\phi_i$ and $\phi_j$) be written $sim(\phi_{dij})$, then we can define the pairwise similarities of the top $k$ ranked documents as:

$$W^{\phi} = \begin{bmatrix} sim(\phi_{d11}) & sim(\phi_{d12}) & \dots & sim(\phi_{d1k}) \\ \dots & \dots & \dots & \dots \\ sim(\phi_{dk1}) & sim(\phi_{dk2}) & \dots & sim(\phi_{dkk}) \end{bmatrix} \tag{3.6}$$

We can then apply autocorrelation (denoted as AC above) as per Equations (3.2) & (3.3). We denote this as *AC-embs*.

#### Network-embs

Similarly, and as we showed that the similarity matrix is equivalent to a fully connected network set of edges, we can apply WAND and WD as per Equations (3.4) & (3.5), denoted as *WAND-embs* and *WD-embs*, respectively.

**pairRatio**

We now introduce an extension of AC-embs inspired by visually exploring embedding relations. Specifically, in Figure 3.2, we visualise the pairwise similarity matrix ($W_\phi$) obtained using TCT-ColBERT (Lin et al., 2020) embeddings for the top-100 passages for the one high and one low performing query in the TREC Deep Learning Track 2019 queryset. For the best performing query, there is higher pairwise similarity among documents of top ranks (top left corner, indicated by a group of lighter shading), and lower correlation for lower ranks (darker shading). On the other hand, for the worst query, elements of darker shading appear at high ranks, indicating that the top-ranked documents may not be as coherent). In addition, there is less dark shading in low ranks compared to the best query. These observations inspire us to explore the trend of average top vs. bottom rank pairwise similarities of top-ranked embeddings.

Specifically, let $W^\phi_{\tau_1..\tau_2}$ denote the (diagonal) subset of $W^\phi$ between ranks $\tau_1$ and $\tau_2$. Then, for a given rank threshold $\tau$, we can measure the ratio between the mean pairwise similarity above and below rank $\tau$, i.e. $W^\phi_{0..\tau}$ and $W^\phi_{\tau..k}$ as follows:

$$pairRatio(W^\phi) = (\overline{W^\phi_{1..\tau_i}}) \cdot (\overline{W^\phi_{\tau_j..k}})^{-1} \tag{3.7}$$

where $\overline{W^\phi}$ denotes the mean of the given matrix, $\tau_i$ corresponds to the end of the upper matrix, and $\tau_j$ symbolises the start of the lower matrix (we use the two cutoff points as separate hyperparameters). We called this predictor *pairRatio*. Unlike WAND and WD, we consider the magnitude of this contrast as indicative of query performance. We believe that, since this relates to the retrieval method itself, it should be indicative of query performance especially for advanced retrieval methods.

Still, the similarity matrix $W^\phi$ can only provide information about the relative similarity of documents. Introducing some information about the document scores would increase performance prediction accuracy, since it relates to the absolute ranking of each document. Let $A$ be an adjusted matrix, where each entry, for a document pair $i$ and $j$ is multiplied by the final similarity of the query to each of the documents:

$$A_{ij} = W_{ij} \cdot (\phi_i \cdot \theta_q) \cdot (\phi_j \cdot \theta_q)$$

$A$ better encodes similarity of the query among the pairwise document similarities. pairRatio (Equation (3.7)) can then be applied upon $A$, which we denote as adjusted pairRatio, or *A-pairRatio*.

In short, we are interested in the effectiveness of these predictors based on dense document representations and how they perform in relation to their sparse versions. Table 3.1 summarises the limitations of each QPP predictor type in relation to their relevance for predicting single-representation dense retrieval models, as described in Section 3.1, as well as the advantages of

our proposed predictors, as described in this Section. Based on these observations, and connecting back to Figure 3.1, the pipeline variations of existing post-retrieval predictors can be described as:

$$Pipe_{\text{score-based}} = Retrieval_{BM25orDense} \gg \text{Correlation} \tag{3.8}$$

$$Pipe_{\text{sparse-coherence}} = Retrieval_{BM25} \gg \text{TD.IDF coherence} \gg \text{Correlation} \tag{3.9}$$

$$Pipe_{\text{dense-coherence}} = Retrieval_{BM25} \gg \text{Emb-based coherence} \gg \text{Correlation} \tag{3.10}$$

$$Pipe_{\text{supervised}} = Retrieval_{BM25} \gg \text{Fine-tuning}_{BERT} \gg \text{QPP estimation} \gg \text{Correlation} \tag{3.11}$$

We see that Equation (3.8) contains the least steps, since the scores are already computer by each retrieval model (taking an average or a standard deviation does not require further computational cost). Then, Equations (3.9) and (3.10) indicate that sparse and dense coherence-based predictors only require the representation pattern calculation before the final correlation. Finally, Equation (3.11) shows that supervised predictors require an additional fine-tuning step before calculating the final predictor, thus requiring the highest computational cost. We test the performance of our proposed dense coherence-based predictors compared to score-based and supervised predictors in Section 3.4 using the above equations for the computation of each predictor type..

Table 3.1: Summary of limitations of existing QPP predictors of multiple types in relation to steps in QPP pipelines and dense retrieval models and the proposed solutions brought by our proposed predictors.

| predictor type | step | limitations/advantages |
|---|---|---|
| Pre-retrieval | 1 | do not provide semantic information relevant for single-representation dense retrieval or provide it only at the token level |
| Score-based | 1 | do not incorporate semantic information as found in dense retrieval models |
| sparse coherence-based | 2 | do not support representations that match dsense retrieval models, dense retrieval model structure requires the extensions their of intuitions |
| dense coherence-based | 2 | match representations of dense retrieval models, facilitate extended intuitions with semantic information, eliminate the need for step 3 |
| supervised | 3 | computationally expensive (more steps in the pipeline), representation mismatch with dense retrieval model representations, worse performance on advanced models |

## 3.3 Experimental Setup

In this section, we begin with a traditional evaluation of QPPs with correlations. To achieve this, our experiments address the following research questions:

**RQ3.1** How do unsupervised coherence-based predictors compare to unsupervised score-based predictors in dense and sparse retrieval?

**RQ3.2** How do unsupervised predictors perform compared to supervised predictors in dense and sparse retrieval?

To address these research questions, our setup is as follows:

**Datasets:** We use the MSMARCO passage ranking corpus, and apply the TREC Deep Learning track 2019 and 2020 query sets, containing respectively 43 and 54 queries with relevance judgements. In particular, each query in these querysets contains many judgements obtained by pooling various distinct retrieval systems.

**QPP Predictors**: As unsupervised score-based predictors, we apply Max score (MAX) (Roitman et al., 2017a), and NQC (Shtok et al., 2009). As a representative variant of NQC, we choose *RSD*. This bootstrap-based predictor is the most recent NQC variant and was shown to outperform other score-based predictors. Specifically, we use the *RSD(uni)* version which samples documents uniformly. Our choice of this RSD variant is indicated by the fact that the other two variants are based on sampling documents according to other term probability-based predictors; instead a uiform sample is in line with all three retrieval models and especially the single-representation dense retrieval models. For each cutoff, we sample from 0.60 to 0.80 of the initial result list size. We use spatial autocorrelation (AC) (Diaz, 2007), WAND and WD (Arabzadeh et al., 2021a), and the interpolation of WAND and WD with NQC (following the findings of the original paper (Arabzadeh et al., 2021a), which suggest that network metrics further increase the performance of NQC). We also report our embedding variants (AC-embs, WAND-embs, WD-embs, PairRatio, A-PairRatio). For each unsupervised predictor, we tune the hyperparameters of each dataset on the other. Specifically, to tune the cutoff value for the top-$k$ documents all unsupervised predictors including ours, we use a grid of values [5,10,20,50,100,200,500,1000]. For PairRatio and A-PairRatio, we also vary the other upper and lower rank threshold hyperparameters $\tau_i$ and $\tau_j$.

For supervised predictors, we report the bi-encoder and cross-encoder variants of BERT-QPP (Arabzadeh et al., 2021b). To achieve this, we retrained the BERT-QPP cross-encoder and bi-encoder models specifically for each of the dense retrieval models. These supervised predictors exhibit their highest correlations mainly for MRR, which means that they train models that estimate the relevance of the top document of a ranking. In this regard, we check whether an alternative supervised predictor (which we call *top-1(monoT5)*) that uses only the top-retrieved document to a monoT5 model (Nogueira et al., 2020) – i.e. trained for relevance estimation and ranking rather than performance prediction – can perform well in dense retrieval. Note that we deliberately use the term *QPP Predictors* instead of *baselines*, since our purpose is not to demonstrate the superiority of a single predictor, but rather how a *group of predictors* behaves under different contexts and retrieval models.

**Retrieval Systems**: We deploy three retrieval approaches: BM25 sparse retrieval (applying Porter's English stemmer and removing standard stopwords) as implemented by Terrier (Ounis et al., 2006), and two single-representation dense retrieval approaches, namely ANCE (Xiong et al., 2020), and TCT-ColBERT (Lin et al., 2020) with PyTerrier (Macdonald et al., 2021) integrations.[2]

**Measures**: Following the TREC 2019 Deep Learning Track Overview (Craswell et al., 2020), we measure system effectiveness in terms of NDCG@10 and MAP@100. We further add MRR@10, following some recent work (Arabzadeh et al., 2021b; Hashemi et al., 2019). To quantify the accuracy of the QPP techniques, we adopt Kendall's $\tau$ correlation measure, as

[2] https://github.com/terrierteam/pyterrier_dr

Table 3.2: Kendall's $\tau$ correlations of unsupervised and supervised predictors for TREC DL 2019. The highest correlation by an unsupervised predictor in each column is emphasised in bold and (*) indicates significance at $\alpha = 0.05$.

| | BM25 | | | ANCE | | | TCT | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAP@100 | NDCG@10 | MRR@10 | MAP@100 | NDCG@10 | MRR@10 | MAP@100 | NDCG@10 | MRR@10 |
| Effectiveness | 0.232 | 0.479 | 0.639 | 0.332 | 0.643 | 0.806 | 0.402 | 0.720 | 0.898 |
| | Score-based | | | | | | | | |
| Max | 0.171 | 0.157 | 0.087 | 0.428* | 0.316* | 0.241* | 0.297* | 0.250* | 0.015 |
| NQC | 0.322* | 0.281* | 0.075 | **0.499*** | 0.463* | 0.216 | **0.335*** | 0.243* | 0.171 |
| RSD(uni) | 0.328* | 0.288* | 0.077 | 0.495* | **0.467*** | 0.264* | **0.335*** | 0.228* | **0.227** |
| | Sparse Coherence-based | | | | | | | | |
| AC | 0.156 | 0.073 | 0.071 | 0.111 | 0.081 | 0.061 | 0.080 | -0.198 | -0.051 |
| WAND | 0.209* | 0.126 | 0.111 | 0.187 | 0.113 | 0.025 | 0.189 | 0.095 | -0.006 |
| WD | 0.158 | 0.101 | 0.087 | 0.158 | -0.004 | -0.009 | 0.184 | 0.121 | 0.015 |
| WAND(NQC) | 0.258* | 0.148 | 0.124 | 0.178 | 0.113 | 0.025 | 0.189 | 0.095 | -0.01 |
| WD(NQC) | 0.200* | 0.186 | 0.035 | 0.158 | -0.008 | -0.012 | 0.18 | 0.135 | 0.006 |
| | Dense Coherence-based | | | | | | | | |
| WAND-embs | -0.096 | -0.232 | -0.019 | 0.138 | -0.157 | -0.029 | -0.036 | 0.139 | 0.041 |
| WD-embs | 0.224* | -0.170 | 0.014 | 0.089 | -0.219 | -0.241* | -0.147 | -0.033 | 0.045 |
| AC-embs | 0.373* | 0.144 | 0.098 | 0.437* | 0.285* | 0.261* | 0.056 | 0.018 | -0.129 |
| pairRatio(ours) | 0.171 | 0.270* | 0.194 | 0.295* | 0.334* | 0.087 | 0.200 | 0.248* | -0.060 |
| A-pairRatio(ours) | **0.446*** | **0.352*** | 0.142 | 0.382* | 0.403* | 0.216 | 0.280* | **0.259*** | 0.171 |
| | Supervised | | | | | | | | |
| BERT-QPP (bi) | 0.229* | 0.305* | 0.260* | 0.162 | 0.144 | 0.067 | 0.111 | 0.048 | 0.083 |
| BERT-QPP(cross) | 0.264* | 0.254* | 0.174* | 0.198 | 0.117 | 0.038 | 0.211* | 0.088 | 0.041 |
| top-1(mono-T5) | 0.180 | 0.294* | **0.359*** | 0.224* | 0.294* | **0.470*** | 0.058 | 0.038 | 0.086 |

typically reported in QPP literature (Cronen-Townsend et al., 2002; Diaz, 2007; Hauff et al., 2008; Shtok et al., 2009, 2010, 2012; Zamani et al., 2018).[3]

## 3.4 Correlation QPP Results

In this section, we report the QPP results with the well-established evaluation methodology based on correlations with system effectiveness. In this way, we compare the performance of our own proposed predictors with other relevant predictors across a variety of contexts co nsisting of different retrieval models and evaluation metrics. In this regard, Tables 3.2 and 3.3 show the accuracy of all our examined predictors on the TREC DL 2019 and 2020 query sets, respectively. Within each table: groups of columns denote the various retrieval approaches; the uppermost row reports the mean effectiveness of each ranking approach for each evaluation measure; the next group of rows contains the Kendall's $\tau$ correlation of the score-based predictors (using Equation (3.8)), the next one the unsupervised lexical coherence-based predictors (using Equation (3.9)); then we report the results for the embedding-based predictors (using Equation (3.10)); and finally for the supervised predictors (Arabzadeh et al., 2021b) (using Equation (3.11)).

[3] In general, Kendall's $\tau$ gives lower scores than Pearson's and Spearman's correlation, but makes the least assumptions about a linear relationship between variables. Therefore, instead of reporting multiple correlation measures, we prefer to report three evaluation measures.

Table 3.3: Results on TREC DL 2020. Notation as per Table 3.2.

| | BM25 | | | ANCE | | | TCT | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAP@100 | NDCG@10 | MRR@10 | MAP@100 | NDCG@10 | MRR@10 | MAP@100 | NDCG@10 | MRR@10 |
| Effectiveness | 0.275 | 0.493 | 0.614 | 0.363 | 0.607 | 0.803 | 0.454 | 0.686 | 0.831 |
| | Score-based | | | | | | | | |
| Max | 0.215* | 0.214* | 0.184 | 0.213* | 0.285* | 0.337* | 0.342* | 0.243* | 0.062 |
| NQC | 0.526* | 0.438* | 0.281* | **0.443*** | 0.082 | 0.172* | **0.454*** | 0.246* | 0.133 |
| RSD(uni) | 0.568* | 0.431* | 0.288* | 0.403* | 0.275* | 0.155 | 0.335* | 0.341* | 0.208* |
| | Sparse Coherence-based | | | | | | | | |
| AC | -0.199* | 0.017 | -0.097 | -0.115 | -0.022 | -0.014 | 0.018 | -0.118 | 0.030 |
| WAND | 0.189* | -0.031 | -0.026 | 0.130 | 0.009 | -0.065 | 0.208* | 0.220* | 0.023 |
| WD | 0.183* | 0.006 | -0.036 | 0.158 | 0.044 | 0.01 | 0.225* | 0.216* | 0.018 |
| WAND(NQC) | 0.220* | 0.101 | -0.024 | 0.130 | 0.005 | -0.067 | 0.202* | 0.213* | 0.188 |
| WD(NQC) | 0.253* | 0.160 | 0.036 | 0.148 | 0.023 | -0.010 | 0.223* | 0.192* | 0.004 |
| | Dense Coherence-based | | | | | | | | |
| WAND-embs | 0.038 | 0.137 | 0.042 | 0.291* | **0.300*** | 0.077 | -0.05 | 0.107 | -0.066 |
| WD-embs | 0.099 | 0.158 | 0.028 | 0.213* | 0.289* | **0.394*** | 0.127 | 0.127 | -0.161 |
| AC-embs | **0.607*** | **0.443*** | 0.339* | 0.324* | 0.219* | 0.149 | 0.121 | 0.137 | -0.002 |
| pairRatio(ours) | 0.271* | 0.203* | 0.130 | 0.178 | 0.186 | -0.132 | 0.364* | 0.318* | **-0.280*** |
| A-pairRatio(ours) | 0.482* | 0.316* | 0.189 | 0.348* | 0.270* | 0.115 | 0.429* | **0.363*** | -0.244* |
| | Supervised | | | | | | | | |
| BERT-QPP (bi) | 0.322* | 0.315* | 0.351* | 0.274* | 0.047 | 0.058 | 0.353* | 0.195* | 0.083 |
| BERT-QPP(cross) | 0.375* | 0.345* | 0.403* | 0.180 | 0.043 | 0.012 | 0.261* | 0.173 | 0.041 |
| top-1(mono-T5) | 0.371* | 0.400* | **0.534*** | 0.259* | 0.237* | 0.365* | 0.279* | 0.240* | 0.159 |

### 3.4.1 RQ3.1: Score-based vs Coherence-based Predictors

As expected, for BM25, distribution-based score predictors (NQC and RSD(uni) show high accuracy for MAP@100 and NDCG@10, while their accuracy is lower for MRR@10, especially for DL 19. However, unlike older datasets, sparse coherence predictors exhibit very low correlations for TREC DL datasets. As for dense coherence predictors, surprisingly, AC-embs variant is the best performing predictor for AP@100, and for NDCG@10 on 2020. As for our pairRatio variants, they are less effective than other unsupervised predictors, such as NQC and AC-embs (except for MRR@10), as well as supervised predictors on MRR@10.

Next we consider the two dense retrieval settings, i.e. ANCE & TCT-ColBERT. For TCT-ColBERT, we observe that our pairRatio predictors outperform not only supervised predictors, but also NQC (the best performing unsupervised predictor) for NDCG@10 and MRR@10 for both datasets, are only behind RS(uni) for MRR@10 in the DL 2019 dataset, and are competitive for AP@100. Another observation is that A-pairRatio has increased the accuracy compared to pairRatio, particularly for the TCT-ColBERT model, which indicates the need for including document-query relations. In summary, for NDCG@10 and MRR@10, for TREC DL 2020, in all four cases our dense coherence-based predictors (any of them considered) outperform score-based predictors; for TREC DL 2019, in two of the four cases ours are higher, in one case RSD is higher, and in one case they are identical. For ANCE, WAND-embs and WD-embs are better than score-based predictors for NDCG@10 and MRR@10 for the 2020 datatset, while they are only slightly behind them in the 2019 dataset. Overall, for MAP@100, NQC or RSD (uni) consistently outperform coherence-based predictors, while for NDCG@10 and MRR@10, the picture is more unstable; however, in most cases, coherence-based predictors win for dense retrieval. Further, as might be expected, changing the type of representations from sparse to

dense increases the performance of coherence-based predictors across the dense retrieval settings (for ANCE,in 7 out of 9 (QPP, Measure) cases in TREC 2019, and 9 out of 9 for TREC 2020; for TCT-ColBERT, our pairRatio variants are more effective), as the updated representations match those of the retrieval methods. To answer RQ3.1, for dense retrieval, score-based predictors perform well for MAP@100, while coherence-based predictors show increased accuracy for NDCG@10 and MRR@10. For sparse retrieval, dense coherence predictors are in general better than score-based predictors.

### 3.4.2 RQ3.2: Unsupervised vs. Supervised Predictors

Next, we compare the performance of unsupervised with supervised QPP predictors for each retrieval method. For BM25, we are able to reproduce the results of the bi-encoder and cross-encoder variants of BERT-QPP, as reflected by the higher values in MRR and the competitive correlation on the other two metrics. For BM25, we used the authors' checkpoints, while we re-trained the method for ANCE & TCT-ColBERT. However, their values are still lower than NQC, (a simple score-based unsupervised predictor), and RSD(uni) (NDCG@10 on the TREC 2019 queryset), our pairRatio (MRR@10 on the 2019 queryset), AC-embs (AP@100 on 2019, AP@100 on 2020, NDCG@10 on 2020), and top-1 monoT5 (MRR@10 on both datasets). Most importantly, for the two dense retrieval methods, supervised predictors are not as effective as unsupervised predictors, such as Max and NQC. For TCT-ColBERT, supervised predictors are less effective than our pairRatio variants for NDCG@10 and MRR@10, and NQC and RSD(uni) for all metrics. The strongest observed correlations of BERT-QPP variants in dense retrieval are for AP@100. However, they have a cost to deploy (applying a BERT model on the top-ranked result). We argue that this resource would be better spent to re-rank the top results. In addition, the simpler "supervised" variant, *top-1(mono-T5),* which uses the monoT5 score of the top-ranked document is a more accurate predictor than BERT-QPP across all retrieval methods, particularly for MRR@10, which is the metric that BERT-QPP is most competitive. This surprising result shows that BERT-QPP is itself just a relevance estimator for the top-ranked document that has been trained to predict MRR@10; using any effective relevance estimator can do as good a job, if not better. To answer RQ3.2, we find that the existing BERT-QPP supervised predictors are less accurate than unsupervised predictors (existing and ours) for dense retrieval.

### 3.4.3 Conclusions from Correlation Results

In summary, we observe that answering RQ3.2 is more straightforward than answering RQ3.1, mainly due to the fact that both score-based and dense coherence-based predictors outperform supervised predictors, but when they are compared with one another, different results are observed according to the researcher's choices (i.e., metric, model, etc.). Our main diference from previous research (Arabzadeh et al., 2020, 2021a,b; Datta et al., 2022b; Hashemi et al., 2019; Shtok et al., 2009, 2010) is that we explicitly mention that out of a variety of contexts, our pro-

posed predictors are suitable for a subset of those. In particular, we show that in most cases (and excluding interpolations with score-based predictors), our dense coherence-based predictors achieve significant improvements compared to their sparse versions especially for ANCE. For TCT-ColBERT, the best performing dense coherence-based predictors are our own extensions pairRatio and A-pairRatio, which is expected, since the intuitions for them were developed on the basis of TCT-ColBERT embedding visualisations. We also showed that while the main improvements are observed for NDCG@10, we see high correlations also for MRR@10 in many cases especially for TCT-ColBERT. While our predictors seem to optimise the more recall-based metric, we still acknowledge the important contribution of score-based predictors, especially in the cases of predicting rankings with MAP@100 (more recall-based metrics). In the next section, we provide a set of explanations for the reasons why this happens. We claim that further factors play a role and further influence query performance, such as query categorisation, which is modeled together with other information used in QPP.

## 3.5 Modeling Query Differences in QPP

As observed in Section 3.4, the performance of dense coherence-based predictors is particularly accurate in certain dense retrieval settings (for TCT-ColBERT: pairRatio and A-pairRatio, for ANCE: WAND-embs and WD-embs) and shows superior performance for especially NDCG@10. Still, score-based predictors are often better for MAP@100. This difference in QPP correlations among evaluation metrics motivates us to explore whether the relationship between QPPs and retrieval effectiveness is mediated by the type of query. For this purpose, we adopt the query taxonomy originally proposed in Bolotova et al. (2022), where the authors proposed a categorisation of questions and a corresponding classifier to facilitate the transfer of these insights to other datasets. In the original study, some query categories, for instance the ones belonging to Experience, Reason, and Debate, were found difficult to answer than the rest (Bolotova et al., 2022). Then, the authors applied their taxonomy to existing known datasets, some of which are similar to the ones we use for this Chapter. We, therefore, apply their proposed classifier to the two TREC Deep Learning query sets. The results are observed in Table 3.4. We observe a class imbalance very similar to the original study, where they also observed fewer examples of Experience and Reason queries for the TREC and MSMARCO datasets. Instead, for our query sets, as well as the original study, a lot more example of Factoid and Evidence-based queries are included, which is expected due to the nature of the query sets. This result does not prevent us from studying further the effect of query categorisation or *query types*, as we use it. In the rest of the section, we describe how we achieve to quantify the effect of query types in combination with QPP performance of different predictors.

In particular, for this purpose, we apply a distribution-based QPP evaluation approach based on the scaled Absolute Rank Error (sARE) (Faggioli et al., 2021b). Specifically, the sARE value each query is calculated as: $sARE_{q_i} = \frac{|r_i^p - r_i^e|}{|Q|}$ where $r_i^p$ and $r_i^e$ are the ranks assigned to query $i$

by the QPP predictor and the evaluation metric, respectively (one sARE value is obtained per query, instead of a point estimate), while $Q$ is the set of queries. This further allows using sARE in statistical models (Faggioli et al., 2021b, 2023b). Unlike Faggioli et al. (2021b) and Faggioli et al. (2023b) who use ANOVA, we use *Linear Mixed Effects (LME)* models (Curran et al., 1997; Field et al., 2012; Maxwell et al., 2017; Singer and Willett, 2003), which also belong to Generalised Linear Models (GLM) (Madsen and Thyregod, 2010; Nelder and Wedderburn, 1972), but split the total explained variance in *sARE* into 2 levels. At this point, it should be noted that we could have instead opted for repeated measures ANOVA in order to model the repeated measurement of the same query over multiple QPP predictor measurements. However, we opt for Linear Mixed Models, as they make the less assumptions and they are less influenced by the class imbalances (ANOVA is a method more influenced by the number of observations in each group). In addition, as we show in the rest of the section, Linear Mixed models allow us to provide exact proportions of explained variances caused separately by QPPs and query types, which is a useful indicator of the extent of influence of each factor. Next, we describe the 2-Level approach.

Table 3.4: Classification of queries from the two TREC Deep Learning query sets according to the classifier provided by Bolotova et al. (2022). Numbers indicate the amount of queries in each category.

| | dataset | |
|---|---|---|
| Query Type | TRECDL2019 | TRECDL2020 |
| Factoid | 14 | 28 |
| Reason | 1 | 3 |
| Evidence-based | 26 | 20 |
| Instruction | 1 | 2 |
| Experience | 0 | 1 |
| Not-a-question | 1 | 0 |

Specifically, Level 1 specifies the within-query variations (how each query changes or the per query variance over different QPP predictors). Level 2 specifies the between-query differences; it further explains each part of Level 1 by showing, how it changes according to a between-query factor - here we use the type of query or *query type* as proposed in (Bolotova et al., 2022). A 2-Level approach is necessary to model the interplay of QPPs with query types; while sARE can vary between QPP measurements (each query receives a separate sARE value for each QPP predictor), each measurement might fluctuate differently based on what type of query we use (multiple queries in the same type share a similar behaviour regarding query performance), and are, therefore, nested within their group (each query belongs to only one level of query type). Thus, the multilevel approach allows splitting the total variation in sARE into within (due to QPPs - Level 1)- and between-query (due to query types - Level 2) variation. Using separate models for each evaluation measure allows to check which measure is more affected by query types. Note also that a Linear Mixed Models analysis requires us to move from a query-level dataset (as shown in Table 3.6) to a query-qpp measurement dataset (Table 3.7). The new

Table 3.5: Explanation of terms included in the linear mixed effects full model.

| Parameter | Interpretation |
|---|---|
| Fixed effects | |
| $\gamma_{00}$ | average true sARE for the reference QPP predictor for the reference (without the effect of) query type |
| $\gamma_{01}$ | average difference in sARE between different query types for the reference QPP predictor |
| $\gamma_{10}$ | average true rate of change in sARE per unit change in QPP predictor for the reference (without the effect of) query type |
| $\gamma_{11}$ | average difference in sARE between different query types per unit change in QPP predictor |
| Random effects | |
| $\zeta_{0i}$, $\zeta_{1i}$ | allow individual true query trajectories to be scattered around the average query true change trajectory |
| $\varepsilon_{ij}$ | allows individual query data to be scattered around individual query true change trajectory |
| Variance Components | |
| $\sigma^2_{\varepsilon}$ | level 1 (residual) variance, variability around each query's true change trajectory |
| $\sigma^2_0$, $\sigma^1_1$ | level 2 variance in reference predictor and rate of change per predictor measurement, how much between-query variability is left after accounting for query type |
| $\sigma_{01}$ | residual covariance between true sARE for the reference (initial) predictor and rate of change, controlling for query type, across all queries |

dataset format allows us to have multiple measurements by splitting the original QPP columns to separate rows with value and measurement. Next, we describe LMEs in detail.

Table 3.6: Query-level dataset originally used in modeling QPP

| Entry | qid | $sARE_1$ | $sARE_2$ | ... | $sARE_{11}$ | $QPP_1$ | $QPP_2$ | ... | $QPP_{11}$ | QueryType |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.3 | 0.4 | ... | 0.5 | 3.3 | 13.6 | ... | 10.4 | Factoid |
| 2 | 2 | 0.2 | 0.5 | ... | 0.7 | 4.5 | 4.7 | ... | 9.6 | Experience |
| 3 | 3 | 0.4 | 0.2 | ... | 0.6 | 7.9 | 6.5 | ... | 18.5 | Reason |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 97 | 97 | 0.4 | 0.7 | ... | 0.8 | 4.6 | 8.7 | ... | 13.2 | Evidence |

### 3.5.1 Linear Mixed Model Definitions

First, our full model, denoted as $LME_{full}$, is defined as :

**Level 1**

$$sARE_{ij} = \pi_{0i} + \pi_{1i}(QPPPredictor) + \varepsilon_{ij} \tag{3.12}$$

with $\varepsilon_{ij} \sim N(0, \sigma^2_{\varepsilon})$

where $sARE_{ij}$ is the sARE of query $i$ at QPP predictor measurement $j$, $\pi_{0i}$ is the intercept (initial status) of query $i$'s change trajectory (reference QPP predictor, i.e., the first QPP measurement), $\pi_{1i}$ is the slope (rate of change) in sARE (per predictor unit), and $\varepsilon_{ij}$ are the deviations of a query's equation on each measurement. This is also a way for Level 1 to check for statistically significant differences between predictors.

Table 3.7: Query-QPP level dataset using the multilevel approach in modeling QPP as proposed in this Chapter.

| Entry | qid | sARE | QPP value | QPP measurement | QueryType |
|---|---|---|---|---|---|
| 1 | 1 | 0.3 | 3.3 | WAND-embs | Factoid |
| 2 | 1 | 0.4 | 13.6 | WD-embs | Factoid |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 11 | 1 | 0.5 | 10.4 | RSD | Factoid |
| 12 | 2 | 0.2 | 4.5 | WAND-embs | Experience |
| 13 | 2 | 0.5 | 4.7 | WD-embs | Experience |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 22 | 2 | 0.7 | 9.6 | RSD | Experience |
| 23 | 3 | 0.4 | 7.9 | WAND-embs | Reason |
| 24 | 3 | 0.2 | 6.5 | WD-embs | Reason |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 33 | 3 | 0.6 | 18.5 | RSD | Reason |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1057 | 97 | 0.4 | 4.6 | WAND-embs | Evidence |
| 1058 | 97 | 0.7 | 8.7 | WD-embs | Evidence |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1067 | 97 | 0.8 | 13.2 | RSD | Evidence |

**Level 2**

$$\begin{cases} \pi_{0i} = \gamma_{00} + \gamma_{01}(QueryType) + \zeta_{0i} \\ \pi_{1i} = \gamma_{10} + \gamma_{11}(QueryType) + \zeta_{1i} \end{cases} \tag{3.13}$$

$$\text{with } \begin{matrix}\zeta_{0i}\\ \zeta_{1i}\end{matrix} \sim MVN\left[\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}\sigma_0^2 \sigma_{01}\\ \sigma_{01}\sigma_1^1\end{bmatrix}\right]$$

where $\gamma_{00}$ and $\gamma_{10}$ are the average true sARE for the reference query type in the initial status and rate of change, respectively. Similarly, $\gamma_{01}$ and $\gamma_{11}$ show the effect of the between-query factor on sARE, for the initial status and rate of change. For convenience, we use $LME_{full}$ in an equivalent compact form (Levels 1 and 2) as:

$$\begin{aligned} sARE_{ij} = [\gamma_{00} + \gamma_{10}(QPPPredictor_{ij}) + \gamma_{01}(QueryType_i) \\ +\gamma_{11}(QueryType_i)(QPPPredictor_{ij})] \\ +[\zeta_{0i} + \zeta_{1i}(QPPPredictor_{ij}) + \varepsilon_{ij}] \end{aligned} \tag{3.14}$$

Table 3 shows the interpretation of each of the $LME_{full}$ parameters. Next, we introduce two reduced models. We start with $LME_{average}$ that only assumes an average sARE value:

$$sARE_{ij} = \gamma_{00} + \zeta_{0i} + \varepsilon_{ij} \tag{3.15}$$

Table 3.8: LMEs comparison and corresponding variance reduction type. Each row shows the $Pseudo-R^2$ of interest together with its definition.

| Models compared | Quantity | Definition |
|---|---|---|
| $LME_{average}$, $LME_{QPP}$ | $Pseudo-R^2_{\varepsilon}$ | $\frac{\sigma^2_{\varepsilon_{LME_{average}}}-\sigma^2_{\varepsilon_{LME_{QPP}}}}{\sigma^2_{\varepsilon_{LME_{average}}}}$ |
| $LME_{QPP}$, $LME_{Full}$ | $Pseudo-R^2_0$ | $\frac{\sigma^2_{0_{LME_{QPP}}}-\sigma^3_{0_{LME_{full}}}}{\sigma^2_{0_{LME_{QPP}}}}$ |
| $LME_{QPP}$, $LME_{Full}$ | $Pseudo-R^2_1$ | $\frac{\sigma^2_{1_{LME_{QPP}}}-\sigma^3_{1_{LME_{full}}}}{\sigma^2_{1_{LME_{QPP}}}}$ |

Table 3.9: Resulting LME models for each retrieval method and all metrics.

| | |
|---|---|
| | BM25 |
| $sARE_{MAP}$ | $sARE_{ij}=[0.29-0.009(QPPPredictor_{ij})]+[\zeta_{0i}+\zeta_{1i}(QPPPredictor_{ij})+\varepsilon_{ij}]$ |
| $sARE_{NDCG}$ | $sARE_{ij}=0.26+\zeta_{0i}+\varepsilon_{ij}$ |
| $sARE_{MRR}$ | $sARE_{ij}=0.30+\zeta_{0i}+\varepsilon_{ij}$ |
| | ANCE |
| $sARE_{MAP}$ | $sARE_{ij}=[0.28-0.008(QPPPredictor_{ij})+0.25(NotAQ_i)+0.05(NotAQ_i)(QPPPredictor_{ij})]+[\zeta_{0i}+\zeta_{1i}(QPPPredictor_{ij})+\varepsilon_{ij}]$ |
| $sARE_{NDCG}$ | $sARE_{ij}=0.25+\zeta_{0i}+\varepsilon_{ij}$ |
| $sARE_{MRR}$ | $sARE_{ij}=[0.35-0.008(QPPPredictor_{ij})]+[\zeta_{0i}+\zeta_{1i}(QPPPredictor_{ij})+\varepsilon_{ij}]$ |
| | TCT-ColBERT |
| $sARE_{MAP}$ | $sARE_{ij}=[0.32-0.01(QPPPredictor_{ij})+0.05(Experience_i)(QPPPredictor_{ij})]+[\zeta_{0i}+\zeta_{1i}(QPPPredictor_{ij})+\varepsilon_{ij}]$ |
| $sARE_{MAP}$ | $sARE_{ij}=[0.32-0.01(QPPPredictor_{ij})+0.02(Reason_i)(QPPPredictor_{ij})]+[\zeta_{0i}+\zeta_{1i}(QPPPredictor_{ij})+\varepsilon_{ij}]$ |
| $sARE_{NDCG}$ | $sARE_{ij}=[0.32-0.008(QPPPredictor_{ij})]+[\zeta_{0i}+\zeta_{1i}(QPPPredictor_{ij})+\varepsilon_{ij}]$ |
| $sARE_{MRR}$ | $sARE_{ij}=0.32+\zeta_{0i}+\varepsilon_{ij}$ |

Finally, we obtain $LME_{QPP}$ as follows:

$$sARE_{ij}=\gamma_{00}+\gamma_{10}(QPPPredictor_{ij})+\zeta_{0i}+\zeta_{1i}(QPPPredictor_{ij})+\varepsilon_{ij} \tag{3.16}$$

In what follows, we use a model selection strategy, as indicated in Table 4, where each row shows the models being compared, the quantity of interest, and its definition. The difference between $LME_{average}$ and $LME_{QPP}$ is the effect of QPP predictor; $Pseudo-R^2_{\varepsilon}$ tells us how much of the total variability within queries can be attributed to QPPs. Similarly, when comparing $\sigma^2_0$ and $\sigma^2_1$ of $LME_{full}$ with the ones of $LME_{QPP}$, these two models differ in the inclusion of the terms $\gamma_{01}(QueryType)$ and $\gamma_{11}(QueryType)$. $Pseudo-R^2_0$ and $Pseudo-R^2_1$ tell us how much of the total variability between queries in initial status and rate of change, respectively, are due to query type. Starting from $LME_{average}$, we sequentially move to $LME_{QPP}$ and $LME_{full}$, if needed. At each step, we compare between the model that contains the added factor and the one that does not. The decision is made based on the significance of fixed effects and the model Deviance (Maxwell et al., 2017; Singer and Willett, 2003), indicating the goodness-of-fit (the lower, the better). The deviance in this case is: $Deviance=-2LL_{Max}$, where $LL_{Max}$ is the maximised log-likelihood of each model. We implement the proposed LMEs using the `lme4` R package (Bates et al., 2009; Team, 2021), with Full Maximum Likelihood Estimation. We merge the two TREC DL query sets, and each query is assigned to one of the following *query types*: Evidence-based, Factoid, Experience, Instruction, Reason, and Not a Question Bolotova et al. (2022). In our analysis, we only include predictors that are most representative for dense

Table 3.10: Proportion of explained variance per component and included fixed effects in each LME for all three retrieval methods. ✓indicates the presence of a fixed effect in LMEs, while ✗shows the absence of either an important contribution of a factor (top) or a fixed effect (bottom).

| | BM25 | | | ANCE | | | TCT-ColBERT | | |
|---|---|---|---|---|---|---|---|---|---|
| sARE → | MAP | NDCG | MRR | MAP | NDCG | MRR | MAP | NDCG | MRR |
| $Pseudo - R^2_\varepsilon$ | 13.4% | ✗ | ✗ | 7.5% | ✗ | 16.5% | 12.4% | 14.6% | ✗ |
| $Pseudo - R^2_0$ | ✗ | ✗ | ✗ | 17.2% | ✗ | ✗ | 2.2% | 9.9% | ✗ |
| $Pseudo - R^2_1$ | ✗ | ✗ | ✗ | 35.6% | ✗ | ✗ | 22.8% | 8.1% | ✗ |
| $\gamma_{00}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $\gamma_{01}$ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| $\gamma_{10}$ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ |
| $\gamma_{11}$ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |

retrieval: NQC, RSD(uni), Max, dense coherence-based predictors, and supervised predictors. To properly match the type of representations, for BM25, we use the sparse version of coherence predictors, as they were originally used for BM25 rankings[4]. We repeat the procedure for the sARE resulting from each evaluation metric separately and obtain the selected models in each case. We now address the following research questions:

**RQ3.3** Is the accuracy of query performance prediction influenced by query type more for dense retrieval than sparse retrieval?

**RQ3.4** How sensitive are the different evaluation measures to (a) query types and (b) QPPs?

### 3.5.2 RQ3.3 - Importance of Query Type

Table 3.9 provides the resulting LMEs from our model comparison strategy, as outlined in Section 3.5. For the dense retrieval models, Equations with $sARE_{MAP}$ contain a coefficient that indicates sensitivity to a particular type of query, (the first line of ANCE refers to Not-A-Question queries, and the first two lines in TCT-ColBERT refer to Experience and Reason queries). The corresponding BM25 LMEs do not contain a query type coefficient).

Most importantly, in Table 3.10, the top half of shows the proportions of gained explained variance for both levels (with ✗ indicating no significant gains), while the bottom half highlights the included effect terms. The first row shows that variations due to QPPs are similar for the three retrieval methods (similar $Pseudo - R^2_\varepsilon$ values). However, the next two rows have much higher relative gain in explained variance for the two dense models than BM25, especially for $Pseudo - R^2_1$, reaching 35% and 23% for ANCE and TCT-ColBERT, respectively. Indeed, as $Pseudo - R^2_1$ includes query type, this means that a noticeable proportion of the variance is attributed to query type. Therefore, for dense retrieval, some query types are more accurately predicted by certain QPPs, and other query types work better for other QPPs. This indicates that QPP performance cannot be judged in isolation from query taxonomies, which in some cases are

[4] As a sanity check, we also conducted a separate analysis using the dense coherence predictors to obtain sARE from BM25, and the results were similar.

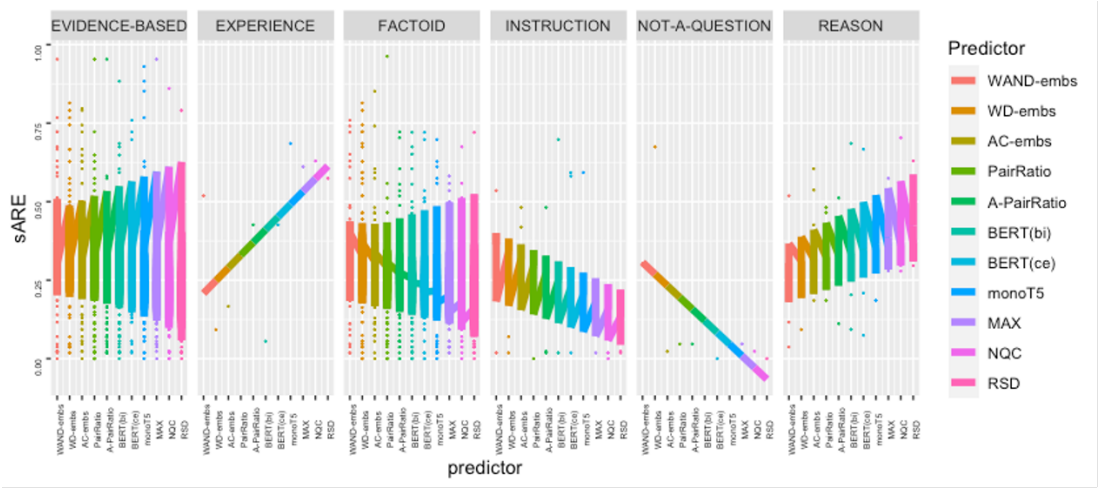
more influential than the predictor itself. To answer RQ3.3, the accuracy of query performance is influenced by query type more for dense retrieval than sparse retrieval.

At this point, it should be noted that the sARE results from MRR@10 are a special case: while it is important to provide the full results to complement the insights from all three evaluation metrics, results from MRR are not extremely suitable to study differences due to query type. In particular, MRR@10 mainly considers the success of the top document being relevant, and therefore, the results are not as sensitive to the effect of different query categories. For example, we note that in Table 3.9, we only observe within-query differences in QPP predictor for MRR. Still, when moving to Table 3.10, we do not see any change in rows 2 and 3 for any of the models. For MRR, we mainly consider the results obtained in Section 3.4, which show similar performance with NDCG@10. Taking that into account, together with the fact that the main differences in Section 3.4 were observed between MAP@100 and NDCG@10, in the following Section, we consider these two metrics and compare how each is sensitive to the different query types.

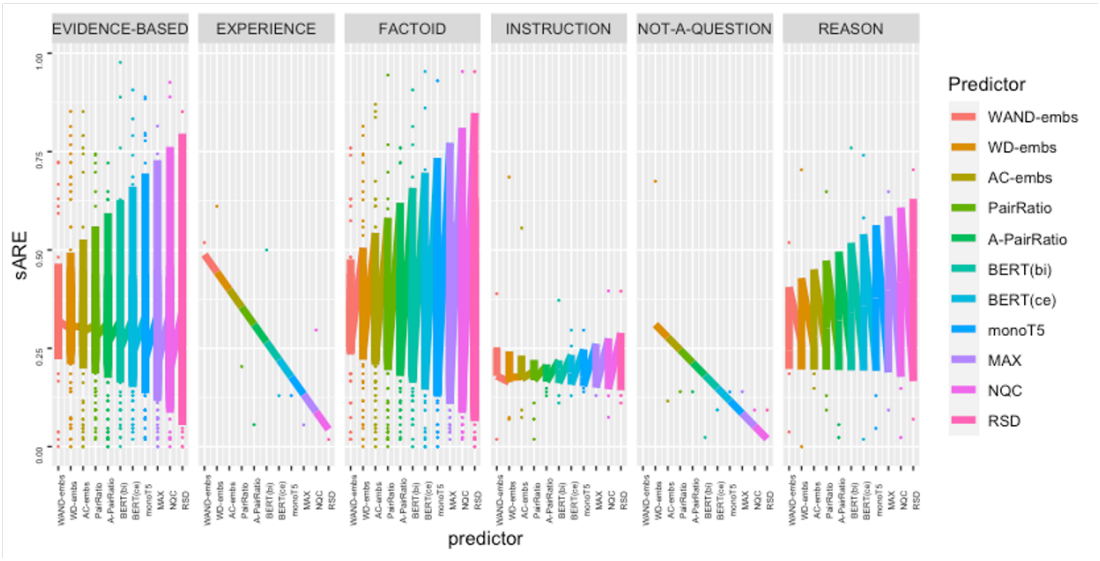
### 3.5.3 RQ3.4 - Sensitivity of Evaluation Measures

Figure 3.3 plots the TCT-ColBERT $LME_{Full}$ of *sARE* prediction for both $sARE_{MAP}$ (a) and $sARE_{NDCG}$ (b). In each plot, the *sARE* (y-axis) values are plotted as a function of QPP predictor (x-axis), with each query type as a separate plot, and colouring indicate different QPP predictors (from left: starting with dense coherence-based predictors, then supervised, and score-based on the right). For $sARE_{MAP}$, the trends for two query types, Experience and Reason, behave differently than the rest; these two types show better performance (lower *sARE*) for coherence-based than score-based predictors, while the opposite holds for Instruction and Not-A-Question queries. As for Evidence-based and Factoid queries, there is higher variance in *sARE* among different queries, but for dense coherence-based predictors, the variance is smaller than score-based predictors, as indicated by the corresponding colours. In general, for $sARE_{MAP}$, performance seems to be affected by the different types of queries, which make QPPs more unstable. Indeed, Experience and Reason were found as *harder* questions in the original categorisation study (Bolotova et al., 2022). This result reflects the selected model for $sARE_{MAP}$, which was $LME_{Full}$ (effect of query type across QPP measurements).

On the other hand, for $sARE_{NDCG}$, QPP performance for different query types seems more uniform. The trend still looks different for Experience and Not-A-Question queries compared to the rest, but those represent only a small portion of the total queries. For the remaining types, the structure is similar, with some variations in strength. Importantly, for Evidence-based, Factoid, Instruction, and Reason queries, there is increasing variance across queries for score-based compared to dense coherence-based predictors. This indicates that our proposed predictors are less sensitive to query type compared to score-based and supervised predictors. Note that while we plot the full model, for $sARE_{NDCG}$, $LME_{QPP}$ was preferred, i.e., only an effect




(a) MAP@100




(b) NDCG@10

Figure 3.3: LME results from the full model for TCT-ColBERT.

of QPP predictor. This is complemented by Table 3.9, where $sARE_{NDCG}$ contain a coefficient for QPPs, but not for query types or their interaction with QPPs.

To summarise, in Section 3.4, we observed that score-based predictors showed improved performance for MAP@100, but our LME analysis showed that this result is susceptible to influential query types. Instead, our dense coherence-based predictors showed higher correlations mainly for NDCG@10, and with the LME analysis (lack of query type terms and $Pseudo - R^2$ terms at Level 2), we showed that this is more stable across different query types. Therefore, our predictors provide promising evidence for generalisability compared to existing predictors. In other words, while both MAP@100 and NDCG@10 are sensitive to QPPs, NDCG@10 is less sensitive to query type variations than MAP@100, thereby answering RQ3.4.

## 3.6 Conclusions

In this Chapter, we relied on the dense embedded representations of single-representation dense retrieval models to provide us with useful information regarding patterns among top-retrieved documents that can be indicative of query performance. In particular, we studied and answered

the first hypothesis of the thesis statement (Section 1.2) that examining the coherence of the top-retrieved items based on their dense embedded textual representations, we can predict the effectiveness of a dense retrieval ranking. This is very important for contemporary CRS systems which consist of image-based result lists and text-based feedback, both of which are represented by dense embedded representations. Therefore, by examining the top-retrieved textual results for the purpose of Query Performance Prediction applied on dense retrieval models, we can easily transfer the insights to conversational recommendation models, taking advantage of the similarity of their dense embeddings. Indeed, throughout the Chapter, we examined the accuracy of QPP upon two single-representation dense retrieval methods (ANCE (Xiong et al., 2020) and TCT-ColBERT (Lin et al., 2020)). In particular, in Section 3.2, we proposed new variants of unsupervised coherence-based predictors and managed to increase their performance for dense retrieval. Specifically, starting from existing sparse coherence-based predictors (Section 3.2.1), we revisited the existing intuitions and also further developed our own intuition to propose a group of dense coherence-based predictors specifically designed for the purpose of optimising single-representation dense retrieval rankings (Section 3.2.2). We achieved this by carefully visualising the dense embeddings of TCT-ColBERT (Figure 2.3) and showing that it is sufficient to consider the ranking method reasoning in order to predict the corresponding ranking. In this way, we showed that changing the representations from TF.IDF to neural embeddings provided by the dense retrieval models together with some further modifications is enough to generalise performance of unsupervised predictors in relation to supervised ones. Indeed, with increasing effectiveness brought by dense retrieval methods, our proposed predictors becomes more competitive, especially for NDCG@10 and MRR@10 (Section 3.4). Also, throughout the Chapter, we highlighted that focusing on a single evaluation measure to optimise a proposed predictor can be problematic and may falsely inform future studies, since MAP@100 and NDCG@10 cannot be used interchangeably.

At the same time, we demonstrated the interplay between the different QPP predictors, evaluation metrics, and the particular types of queries. Indeed, in Section 3.5, we introduced a new perspective to study QPP on the different retrieval models. Specifically, we examined a series of Linear Mixed models, which allow the simultaneous modeling of QPPs and query types (resulting from classifiers of query taxonomies) as two Levels or different sources of variation in query performance. With this methodology, we showed in Section 3.5.2 that dense retireval models are more sensitive to query types and that MAP@100 is more sensitive than NDCG@10 (Section 3.5.3). Importantly, we showed that while score-based predictors still remain very competitive for MAP@100, our examined statistical models indicate that MAP@100 is highly influenced by the type of query. Instead, using NDCG@10, QPP performance is more stable across queries, and since our proposed predictors show higher performance on this metric, this is a promising result for more generalisable performance in dense retrieval. Overall in this chapter, we validated the first claim of the thesis statement: *Initially, we can predict the effectiveness*

*of a ranking of textual items for a textual query, by examining the coherence of the top-retrieved items based on their dense embedded representations.*

While our insights provide a useful starting point, a number of limitations remain. For example, QPP results are limited to a small number of retrieval models and the wide range of QPP predictors have not been applied to recommendation settings, for example in a Conversational Image Recommendation setting, in order to directly test them in the setting of our focus. In addition, predictors have not been examined in a multi-turn setting, taking advantage of the nature of Conversational Image Recommendation ranked lists of items, which correspond to our task of interest. Therefore, in Chapter 4, we test these predictors to the state-of-the-art models of Conversational Image Recommendation, namely GRU (Guo et al., 2018; Wu et al., 2021a) (see Section 2.2.1) and EGE (Wu et al., 2021b) (see Section 2.2.2).

# Chapter 4

# Conversational Performance Prediction (CPP)

As mentioned in Chapter 2, the rising popularity of conversational systems has impacted research in recommender systems in the last few years (Christakopoulou et al., 2016; Jannach et al., 2021; Li et al., 2018; Ren et al., 2022). Especially in the domain of e-commerce, with the increased use of online shopping platforms, Conversational Recomendation Systems (CRSs) assist users in finding items of interest (Sun and Zhang, 2018; Zou and Kanoulas, 2019) in a number of task-oriented goals in the context of a dialogue (Jannach et al., 2021), thus allowing for more complex recommendation settings compared to traditional RS by suggesting items in response to voice or (natural language) chat interactions. In particular, as mentioned in Section 2.1.2, one important aspect of natural language-based CRSs is that they allow a *multi-turn dialogue* with users to assist them with achieving their task-oriented goals (Jannach et al., 2021). Indeed, at each turn, users can provide their natural language feedback in the form of an utterance also known as a *critique* (Tou et al., 1982), which helps the system to improve recommendations (Chen and Pu, 2012). In this way, CRSs allow users to explore the range of available options and elicit their preferences (Cai et al., 2021; Jannach et al., 2021; Jin et al., 2019), also in the cases of limited available options (Bursztyn et al., 2021). Furthermore, in Section 2.2.2, we introduced the research paradigm used throughout our work, which is focused upon conversational fashion image recommendation Berg et al. (2010); Guo et al. (2018); Wu et al. (2021b). A summary of the main setting of conversational image recommendation task is given in Figure 4.1. In short, the user has a target item in mind, and provides textual feedback (critiques) to direct the system towards retrieving images of fashion products that are more similar to to their perceived target item. In this setting, the evaluation of a conversation stops at turn 10 and the system is expected to return the user's target item by the "last turn"; the earlier the turn the item is retrieved, the better the objective performance of a system.

However, in practice, not all conversations lead to a satisfying outcome for the user. This is easily quantified in offline evaluation scenarios such as the scenario presented above, where the

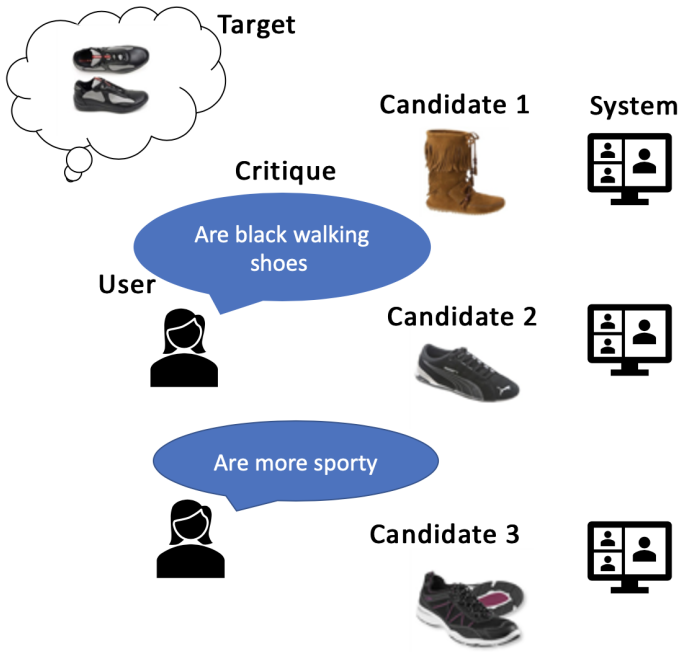



Figure 4.1: Example of Dialog-based recommendation in CRS. Pictures and dialogues from the Shoes dataset (Berg et al., 2010; Guo et al., 2018).

CRS is evaluated across a pre-defined number of turns. For example, Yu et al. (2019) found that, although users had the option to explore a list of options for a number of turns, the system was unable to find a relevant recommendation by turn 7, which might mean that the algorithm was still exploring the space. Also, in Wu et al. (2021b), the target image item was found by the system at rank 1 in only 42% of conversations after (a maximum of) 10 turns. Therefore, exploration might result in an increased number of turns, which on one hand might mean more engaged users (Jin et al., 2019), but at the same time suggests that often the conversations might fail (i.e., target item not found). In this regard, we are interested in identifying indicators that can detect how this happens – for example, a conversation could fail because the system is unable to find the target item or because the target item is not available. With this in mind, we are inspired by the QPP task (introduced in Section 2.3), a prediction task normally applied to ad-hoc retrieval ranking tasks (and models), and aim to apply it on conversational recommendation.

In this regard, in Chapter 3, we examined the coherence of the top-retrieved items (based on their dense embedded representations of text), and found that this helps to predict ranking effectiveness, thus validating the first hypothesis of the thesis statement (Section 1.2). These results provide us with useful insights, primarily regarding the performance of queries in dense retrieval models, and consequently for transferring the QPP problem to other tasks. While, at first glance, predicting query performance in dense retrieval seems independent from predicting systems such as in Figure 4.1, we show that by adapting the various QPP techniques (presented in Section 2.3) according to the special nature of the task, we can obtain predictive information about ranking lists in contemporary Conversational Recommendation Systems (CRS), which address result lists consisting of both text and image-based embedded representations.

Still, predicting performance (success) in Conversational Image Recommendation presents various challenges. These are related to either (a) task-specific settings in CRS systems, such as the dependence of a feedback utterance on the previous turn results and the lack of relevance judgments (which holds both for traditional RSs and CRSs), and (b) the content of the ranked list results, which contain images instead of text. Also, as we described in Section 2.1.2, the task provides a more dynamic process of satisfaction of user needs. In particular, conversation success in a CRS dialogue is determined by how well the system is able to interpret: (i) the feedback utterance provided by the user and (ii) the quality of the recommendations indicated by the results list, both *of the previous turn(s)*. All these parameters make CRS-based predictions more complex than QPP in its retrieval-based setting, and blurs the lines between pre-retrieval (introduced in Section 2.3.1) and post-retrieval (introduced in Section 2.3.2) predictions. Importantly, knowing whether a conversation is likely to be successful allows the CRS to adjust accordingly - for instance, changing its retrieval strategy, or asking a clarifying question.

As a starting point, the coherence-based predictors examined in Chapter 3 relate to our task of interest; still some clarifications need to be made: First, they inform us how the semantic information (contained in embeddings) and the relations either among the retrieved documents or between the query and the documents can be used to predict a document ranking, but the nature of the ranking lists is different. Specifically, as we mentioned in Section 2.2.2, unlike text retrieval, at every turn, a ranked list is composed of image items, of which the user only sees the very top one(s), which prompts the user to provide textual feedback. This multi-modal information (feedback and the top image items) can be used to predict the following turn(s), but it is not yet clear whether this is pre- or post-retrieval information (while queries/feedback are traditionally perceived as prior and ranked lists as posterior information (Shtok et al., 2016)). As a result, we need to check whether such predictors can work with multi-modal information and shorter rank cutoffs. Second, moving away from the text level, we focused on single-representation dense retrieval models, and in this way, we obtained insights that can be transferred to image embeddings. Still, this was examined in a single-turn context, and it is not clear whether the same predictors could work in a complex multi-turn CRS setting. Third, we examined a variety of evaluation metrics and retrieval models of varying effectiveness, and found that QPP is dependent on the ranking model used and other researcher's choices for evaluation. Therefore, we extend those intuitions to examine whether these evaluation settings are suitable to measure conversation success. We believe that evaluation of query performance in a CRS is dependent on the multi-turn and multi-modal nature of the task.

In this regard, the aim of this chapter is to predict the performance of conversations and the effectiveness degree of CRS systems. For this purpose, we will address the second hypothesis posed in the thesis statement (Section 1.2): *Similarly, we can predict the effectiveness of a ranking of items in a Conversational Recommendation Systems (CRS), which are also based on learned embedded representation of images, where user feedback takes the place of a textual*

*query. Indeed, by introducing a framework of Conversational Performance Prediction (CPP), we can predict the degree of success of a conversation by a CRS - such success can be predicted over a short or long time horizon, thereby predicting current user satisfaction or overall satisfaction of a conversation*. This addresses **Limitation 4a)** (*While some early attempts have been made to adjust to a conversational setting, they do not take into account the multi-turn nature of the task*) and **Limitation 4b)** (*While these attempts were made on Conversational Search, no one has addressed QPP in a multi-turn recommendation setting*). Specifically, in what follows, inspired by existing work on Query Performance Prediction (QPP) (e.g., Carmel and Yom-Tov (2010); Cronen-Townsend et al. (2002); Hauff et al. (2008); He and Ounis (2004)), as detailed in Section 2.3, we aim to predict conversational failures by identifying specific indicators that are associated with failure. In particular, we aim to determine the quality of multi-turn critiquing-based CRS recommendation by proposing predictors that consider the multi-turn aspect of conversational recommendation. The proposed predictors address characteristics that mainly refer to the content of the retrieved results lists of image items, but following the feedback received after a system recommendation. In particular, we develop a variety of predictors: score-based, embedding-based, and supervised, and evaluate in a range of time frames to check how predictions can work within a fixed-turn dialogue. In summary, this chapter makes the following contributions:

- Inspired by existing work in QPP, we propose a framework for *Conversational Performance Prediction (CPP)*, which extends the existing work on QPP to a conversational recommendation setting and aims to predict conversation failures by considering the recommendation ranking at different turns of a conversation, either one turn at a time, or by considering multiple consecutive turns. In this regard, we adapt post-retrieval predictors to address the multi-turn nature of the CRS task.

- As a first step, we address characteristics of the retrieved scores of the top-recommended items and can predict poor performance across a shorter or longer number of turns in the conversation. In other words, we adapt QPP evaluation methodologies to a multi-turn conversational setting, which allows to evaluate CPP predictors, which we call short- and long-term *prediction horizons*.

- As a next step, we extend to embedding-based predictors introduced in Chapter 3, still on the unsupervised side, and compare the performance of score-based and embedding-based predictors by using information of a given ranking to predict the ranking of the next turn.

- As a final step, we introduce *Supervised Conversational Performance Prediction (Supervised CPP)*. To achieve this, we move away from employing external ranking models as used in retrieval-based QPP (Arabzadeh et al., 2021b; Datta et al., 2022b; Hashemi et al., 2019; Meng et al., 2023), and focus instead of using the already existing recommendation

models (an insight obtained from Chapter 3) to develop predictors that gradually learn the representations of the retrieved items at various turns. In particular, we use both score-based and embedding-based supervised CPPs, which either use the existing predictors to classify the conversations as successful or not, and move on to propose a supervised predictor that is based on learning a compressed representation of the top items.

- We evaluate our proposed predictors on the Shoes (Berg et al., 2010; Guo et al., 2018) dataset and the Fashion IQ Dresses and Shirts categories (Wu et al., 2020), using a state-of-the-art user simulator (Guo et al., 2018).

Indeed, we examine CPP in a range of evaluation contexts, and the main findings we obtain can be summarised as:

- When examining score-based predictors using the full relative captioning datasets on the various prediction horizons, short-term prediction gives higher correlations than long-term prediction. In particular, long-term prediction is not possible. Also, short-term prediction works better for earlier turns of a conversation.

- When moving to a more traditional QPP-based evaluation setting by considering a sample of 200 target items per dataset and correlating CPP predictors (score-based and embedding-based) with traditional IR ranking metrics, we find that score-based predictors (especially NQC) perform better in general, although in some cases embedding-based predictors show improved performance for later turns.

- In general, CPP correlations are much lower than correlations in a QPP setting, which leads us to develop a new evaluation methodology for predicting dialogue failures.

- When moving to a classification-based CPP evaluation setting, we observe quite high predictive accuracy for multiple predictors, for both single-turn and multi-turn settings. In particular, our new proposed predictor that learns a compressed representation of the retrieved item in previous turn(s) shows improved performance compared to other CPP predictiors across multiple settings.

The rest of the Chapter is structured as follows: Section 4.1 presents the existing research on QPP, including pre- and post-retrieval predictors, as well as their probabilistic interpretation; Section 4.2 outlines our new proposed framework by providing the formal definitions (Section 4.2.1), our Experimental Setup in Section 4.3.1, and the unsupervised results in Sections 4.3.2, 4.3.3, 4.3.4, and 4.3.5. Next, Section 4.4 provides an explanation of our supervised version of CPP framework together with the proposed predictor definitions in Section 4.4.1. In Sections 4.4.3 and 4.4.4, we present the results for our supervised CPP for single-turn and multi-turn predictions, respectively. Finally, we end with some concluding remarks in Section 4.5. Overall, we find some promise in score-based retrieval predictors for correlation-based

CPP, obtaining medium strength correlations with conversation difficulty - for instance, observing a Spearman's $\rho$ of 0.423 on the Shoes dataset, which is comparable to correlations observed for standard QPP predictors on ad-hoc search tasks. Still, our strongest results come when we introduce supervised CPP, which also demonstrates the usefulness of taking advantage of the item embedded representations.

## 4.1 Related Work: QPP Applications and how we move to conversational settings

In order to predict why a conversation with a CRS might fail, we need to identify indicators that show when the user is unable to find the target item during the interaction. In this section, we discuss a number of QPPs that inspire us, first by briefly summarising predictors that are used in ad-hoc retrieval in Section 4.1.1, and then continue with predictors and approaches that have been applied in conversational settings, and specifically in conversational search in Section 4.1.2.

### 4.1.1 Query Performance Prediction in ad-hoc and passage retrieval

As mentioned in Section 2.3, QPP is used to predict the effectiveness of a search results page performed in response to a query in the absence of human relevance judgments (Carmel and Yom-Tov, 2010). *Query performance predictors* are generally grouped into pre-retrieval, and post-retrieval. *Pre-retrieval* predictors are used to estimate the performance of queries before the retrieval stage, and therefore, are independent of the search performed and the ranked list of results (Hauff et al., 2008). Therefore, predictions are based on properties of query-terms or corpus-based statistics (Cronen-Townsend et al., 2002; Hauff et al., 2008; He and Ounis, 2004; Mothe and Tanguy, 2005; Scholer and Garcia, 2009; Zhao et al., 2008). For a more detailed description of pre-retrieval predictors, please refer to Section 2.3.1 of this thesis. On the other hand, *post-retrieval* predictors are applied on the list of the top-ranked retrieved documents, and therefore use the contents of the returned items. In general, as we described in Chapter 3, post-retrieval predictors are grouped into the following categories: (i) Predictors that examine the *focus of the result list* based on the difference in term frequency from the corpus (Cronen-Townsend et al., 2002; Zhou and Croft, 2007), (ii) predictors that measure the scores' distribution of the ranked list (Cummins, 2014; Roitman et al., 2017a,b; Shtok et al., 2009), (iii) coherence-based predictors (Arabzadeh et al., 2021a; Diaz, 2007) that measure the semantic relations among the retrieved documents in an unsupervised way, (iv) relevance feedback-based predictors indicate similarity between the ranking list and pseudo-(in)effective document lists (Shtok et al., 2010; Zhou and Croft, 2007), and (v) supervised predictors that employ external pre-trained language models using multi-vector representations to predict a ranked list; these predictors still use semantic information among queries and documents (Arabzadeh

et al., 2021b; Datta et al., 2022b; Hashemi et al., 2019). For a more detailed description of post-retrieval predictors, please refer to Section 2.3.2 of this thesis.

### 4.1.2 Query Performance Prediction in Conversational Search

While QPP has been widely explored for (single turn) queries in search settings, its study in conversational settings has received much less attention. Some examples are provided by research on conversational search. For example, Sekulić et al. (2022) examine the predicted effectiveness of the top-retrieved documents for deciding to generate clarifying questions, and specifically some extracted features, such as noun phrases or named entities. In this regard, Krikon et al. (2012) proposed a similar approach that employs named entities to determine if a passage contains the answer to the user's question. Indeed, clarifications are useful for both the user and the system (Aliannejadi et al., 2019; Kiesel et al., 2018; Zamani et al., 2020). For this, Aliannejadi et al. (2019) test the type of question the system needs to ask next to understand what the user is looking for by examining the user's previous answers with a post-retrieval QPP predictor (Pérez-Iglesias and Araujo, 2010). Also, Arabzadeh et al. (2022) try to predict whether the system needs to ask a clarifying question to properly understand the user's query by constructing a coherency network and computing the resulting centrality measures.

Another line of research has focused on adapting existing predictors to the conversational search settings. For example, Roitman et al. (2019) examined a constrained retrieval setting, namely the interaction with a conversational assistant, where the assistant needs to decide whether the provided answer could be accepted. The authors built a classifier that determines the answer quality by adapting some existing QPPs to the answer level (using the score of the top item, which is provided as the answer). In addition, Meng et al. (2023) examine some existing predictors (mainly score-based and supervised) in the context of conversational search by showing how these can be evaluated with suitable metrics and evaluation settings. Finally, another evaluation approach for QPP in conversational search comes recently and in parallel to our work. In particular, Faggioli et al. (2023a) proposed a geometric framework for QPP, which separates the evaluation of a single-turn utterance regardless of which conversation it is part of from predicting entire conversations. This framework projects queries and documents at the embedding space and examines geometric relations among them, where the documents form a hyperspace together with the query, and compute how densely the documents distribute around the query (the more dense this hyperspace, the more the semantic correlation of the query with the documents). Moreover, they proposed two embedding-based predictors based on this intuition and found that these outperform other score-based unsupervised predictors when predicting rankings of conversational search models.

Finally, a further attempt has explored the possibility of QPP applications on entirely image-based retrieval systems. For example, Poesina et al. (2023) proposed an approach called *image Query Performance Prediction (iQPP)*, where the query is also an image, and they found that the

adaptation of predictors does not always lead to improved performance. Still, this attempt does not refer to a multi-turn recommendation scenario. In addition, in our task of interest, the user's query is still textual information. Therefore, while some attempts have been made to adapt QPP to more interactive retrieval settings or in image-based queries, QPP for conversational recommendation, and especially conversational image recommendation, has not been addressed. In particular, we are interested in creating a prediction framework for identifying poorly performing or failed conversations in a recommendation setting. We postulate that these predictors can be useful in several use cases, for instance knowing when to ask for clarifications, or when the users target item cannot be found. Towards achieving this goal, we explore score-based and embedding-based predictors on the unsupervised and supervised side, adapting to the multiple turn nature of the task.

In the next section, we define the CPP task, and provide details about how we define our CPP framework.

## 4.2 Conversational Performance Prediction (CPP)

In this section, we outline our prediction framework specifically designed for Conversational Image Recommendation systems. To predict the likelihood of success of a conversation, we need to consider the salient aspect of the conversational setting, such as the users' feedback and the iterative *turn*-based nature of the interaction process. Our proposed framework works primarily as an evaluation methodology that allows predictions in CRSs and at a further point, extends the intuitions of predictors by creating custom predictors for this evaluation methodology. As introduced in Section 2.2.2, at turn $k$, the user provides textual feedback $f_k$ on the current top-ranked candidate item $i_{k,1}$. Based on this feedback, the conversational recommendation system $\mathscr{C}()$ provides a new ranking , i.e.: $\mathscr{C}(i_{k,1}, f_k) \rightarrow S_k$, where $S_k$ is a ranking of $n$ items with corresponding descending retrieval scores $s_1 \dots s_n$, i.e.: $S_k = [\langle i_{k+1,1} s_1 \rangle, \dots \langle i_{k+1,n}, s_n \rangle]$.

Also, in Section 2.2.2, we introduced the relative captioning datasets that are used to train the user simulators in conversational image recommendation, which contain representations of the target image item, the candidate image item, and the critique. In some cases, even at the end of the evaluation turns, the target item may not be returned. Therefore, predicting the likelihood of a user being satisfied with a conversation may improve user experience. We note some key differences of our approach from the approaches in Section 4.1: (i) We consider the ranking quality across both single and multiple turns to predict the user's satisfaction of a conversation. (ii) Our *retrieval units* are images with the corresponding retrieval scores and embedded representations. (iii) Our *query units* are critiques, which are based on the retrieval of the previous turn. Therefore, we argue that there is no clear distinction between pre-retrieval and post-retrieval predictors, since what is considered post-retrieval of one turn could be seen as a pre-retrieval predictor of the following turn.

In the rest of the section, we present our CPP framework in detail. In particular, we first adapt a number of score-based predictors to the nature of our task, as outlined above, and introduce the concept of prediction horizons to differentiate between shorter and longer-term predictions in Section 4.2.1 and in particular in 4.3.2 and 4.3.3. In this way, we focus our attention in the evaluation settings while keeping the predictors at their simplest form. Then, in Sections 4.3.4 and 4.3.5, we continue by comparing existing (adapted) score-based and embedding-based unsupervised predictors using the most effective identified prediction horizon. We find quite diverse results across datasets. All these prepare us for what follows later in Section 4.4, where we introduce the concept of Supervised Conversational Performance Prediction, where we essentially move away from a correlation-based evaluation approach of query performance. In other words, the results from the unsupervised CPP evaluation indicated that the lower correlation values obtained across predictors point towards predicting instead whether a target item is identified by a given turn and a given rank in a dialogue in the form of a classifier that predicts the class of the item. We achieve this by proposing the corresponding supervised predictors that learn the item representations across turns.

### 4.2.1 CPP Framework Definitions

To build our framework, we are inspired by post-retrieval predictors that study the distributions of retrieval scores, the semantic similarity-based predictors, and the use of reference lists, as introduced in Section. In this regard, we define *recommendation success* as the identification of the target image item by the system before a maximum number of turns is reached, which corresponds to a user being satisfied with the conversation. More formally, the CPP task can be described as a function of the form

$$CPP(F,S) \rightarrow \mathbb{R} \tag{4.1}$$

where $F$ is a sequence each containing $f$ feedback critiques over 1 or more turns, and $S$ is a sequence of results (recommendation) lists consisting of retrieval scores or embedded representations contained in retrieved image items, over 1 or more turns. While Equation (4.1) holds for a general description of our framework, it is important to note that this framework can be instantiated for single-turns, or multiple turns. For instance, in a single-turn setting, we can instance CPP task at a given turn $k$, i.e.:

$$CPP_{\text{single}}([f_k],[s_k]). \tag{4.2}$$

In this case, $f_k$ is the equivalent to what we described as pre-retrieval predictors in Section 2.3.1, since it uses the information contained in the feedback, and therefore, retrieval information is ignored. The main difference is that $f_k$ can also be considered as post-retrieval, for example, when using this information to predict the ranking of a later turn within the same conversation, in the sense that this feedback has influenced the follow-up result list. Similarly, $s_k$ is the equivalent to what we described as post-retrieval predictors in Section 2.3.2, including the content

of the top-retrieved items of both score-based and representation-based unsupervised predictors adapted to our task. On the other hand, for two consecutive turns, $k$ and $k+1$, prediction takes the following form:

$$CPP_{\text{consecutive}}([f_k, f_{k+1}], [s_k, s_{k+1}]). \tag{4.3}$$

In this case, we are interested in how the contents of either the feedback utterances or the results lists interact between two consecutive turns. While the information in the feedback utterances contains information about a user's "query", for the purpose of our CPP task, we mainly focus on the contents (image items, scores, and embeddings) of the recommendation list, as these provide richer information across multiple turns. In addition, the feedback from one turn to the next can be very similar, and is therefore, not extremely indicative of the turn's performance. With this in mind, and adapting the notation above to disregard the feedback sequences, we define a number of unsupervised CPPs for single turns by modifying Equation (4.2) to take the form

$$CPP([s_k]). \tag{4.4}$$

Similarly, for consecutive turns, we modify Equation (4.3) into the following:

$$CPP([s_k, s_{k+1})]. \tag{4.5}$$

Our proposed CPP predictors are described in Table 4.1. Here, we include both score-based and embedding-based unsupervised predictors. We use Equation (4.4) to derive the single-turn predictors. For instance, top-1 denotes the maximum score of any retrieved item (and it is the equivalent to the MAX score predictor (Roitman et al., 2017a) used in Chapter 3); when applying these predictors, we also denote the turn $k$ that the predictor is calculated, i.e. top-1@k is the maximum score of any item retrieved in the ranking produced for turn $k$. Mean denotes the average of the scores of the retrieved items. Also, the sd (standard deviation) of top-n items is the equivalent of NQC (Shtok et al., 2009). As for the embedding-based predictors, we use the CPP equivalents of AC-embs used in Chapter 3 adapted from Diaz (2007), the two network metric predictors that we adapted from Arabzadeh et al. (2021a), namely WAND-embs and WD-embs, our proposed dense coherence-based predictors from Chapter 3, namely pairRatio and A-pairRatio, and finally the geometric-based predictor proposed by Faggioli et al. (2023a), namely Reciprocal Volume (RV). Additionally, we use Equation (4.5) to derive the consecutive-turn predictors. For score-based predictors, we use either the difference in maximum score or the overlap of top-ranked items between two consecutive turns. Finally, for embedding-based predictors, we use the cosine similarity between the embedded representations of the retrieved items of two consecutive turns. The derived predictor values described above are then correlated with an objective ranking effectiveness metric, with higher values indicating a stronger association with a recommendation list.

To address the salient aspects upon the nature of the predictors (single-turn and consecutive

turn), we propose the accuracy of the predictors on different prediction *horizons*, i.e., at what point can a prediction be made, and how does it correspond to the effectiveness of the CRS, as measured at a later turn. In particular, we measure *short-term* horizons (i.e., can we predict the effectiveness of the next turn?); and *long-term* horizons (i.e., can we predict the effectiveness of the last turn); as well as measuring the *longevity* of the prediction (i.e., how useful is an early prediction?). Also, for explaining the predicted success of a conversation using the proposed predictors overall, we have the following intuitions that concern successful interactions in the CRS task:

- For a single turn, if the score of the top-ranked item(s) is high, then the system has a clear representation of the user's desired item, and it can find item(s) that closely matches that representation.
- Similarly, if the embedding-based predictor value of the top-ranked items is high, then the system is doing a good job is predicting the user's desired item, and it is more likely to find item(s) that closely matche this representation.
- In a successful conversation, the retrieval scores of the top-ranked item(s) will increase across multiple turns, as the system becomes more confident in its predictions.
- In a successful conversation, the retrieved items become more similar across turns as the system becomes more confident in its predictions and focuses on the correct part of the item catalogue.
- For two consecutive turns, it is only possible to test the short-term predictions between two turns of the same conversation rather than making a long-term prediction.

Overall, from the above different formulations, it is clear that CPP is a distinct task from QPP that can be addressed by different families of predictors. In this chapter, we adapt two categories of unsupervised QPP predictors into the CPP framework. In the remainder of the chapter, we evaluate these predictors on several conversational fashion recommendation datasets.

Table 4.1: Proposed CPP predictors according to number of turns involved.

| Single-turn | |
|---|---|
| Score-based | Embedding-based |
| Top-1 item score (maximum score)<br>Mean score of top-n items<br>Standard deviation (sd) of top-n items | AC-embs<br>Network-embs<br>(A)-pairRatio<br>Geometric |
| Consecutive-turn | |
| Difference in maximum score<br>Overlap of top-ranked items | Embedding cosine similarity |

## 4.3 CPP Experiments (Unsupervised)

In this section, we first detail the Experimental Setup for our unsupervised CPP experiments in Section 4.3.1, and then we continue with the full results corresponding to the RQs defined in it, namely the results for the single-turn score-based predictors in Section 4.3.2, and for the consecutive-turn score-based predictors in Section 4.3.3, thus covering the investigation of prediction horizons. We then move on to describing how a variety of predictors perform in predicting conversational performance by adopting settings of QPP scenarios in Sections 4.3.4 and 4.3.5, where we mainly examine score-based against other representation-based unsupervised predictors. Finally, we conclude with the main insights we obtain throughout the CPP experiments on the unsupervised side in Section 4.3.6, where we highlight the main limitations of studying CPP under the lens of QPP settings and motivate ourselves to create a more customised conversational prediction task.

### 4.3.1 Overview of Experimental Setup

In the first set of experiments, we focus our interest in evaluating CPP across different prediction horizons, as introduced in Section 4.2. For this purpose, we will examine the score-based predictors of Table 4.1 in three different settings. In this regard, for the single-turn predictors, we use three different ground truth settings: the rank of the target item at the end of the conversation (turn 10); the rank of the target item during the conversation, i.e. at a given turn $k$; and the rank of the target item directly after the prediction is made (i.e. $k+1$ for a prediction at turn $k$). Through these different ground truth settings, we can measure CPP accuracy at both short-term and long-term horizons, as well as their longevity. On the other hand, for the consecutive-turn predictors, it is only possible to evaluate with the short-term horizon. Taking all that into account, our first set of research question is:

**RQ4.1** Can we predict conversation performance with predictors based on retrieval scores of a single turn, in terms of (a) long-term and (b) short-term prediction, as well as (c) longevity?

**RQ4.2** Can we predict conversation performance with predictors based on (a) differences in retrieval scores between consecutive turns and (b) overlap in retrieved items of two consecutive turns?

For this purpose, we use the Shoes dataset (Berg et al., 2010; Guo et al., 2018), which contains one relative critique (describing relative differences between recommended and target image pairs) for pairs of shoe images, and the Dresses & Shirts categories of the Fashion IQ dataset (Wu et al., 2020), which contains two relative captions per candidate-target pair. As for the CRS recommendation model, we apply the supervised learning-based version of the GRU sequential recommendation model (Guo et al., 2018; Hidasi et al., 2015) already introduced in Section 2.2.2, which is trained using triplet loss and uses the natural language feedback and the previous recommended images as input, thus maximizing short-term rewards. The model

is configured to retrieve 100 items at each turn. For model training, we use the state-of-the-art user simulator for dialog-based interactive image retrieval based and the relative captioning task (Guo et al., 2018), already described in Section 2.2.2.

In QPP, the accuracy of predictors is evaluated at the query level (a given query is easy or difficult compared to other queries in a set). Specifically, a ranking of queries by the effectiveness of a system, i.e., in terms of Mean Average Precision (ground truth) is correlated with a ranking induced by a predictor. In contrast, we will evaluate CPP predictors at the conversation level (across multiple dialog turns), and therefore, for the ground truth, we evaluate the effectiveness of each *conversation* at identifying the user's target item – more specifically, by considering the rank of the target item at a specific turn of the conversation. Following existing CRS work (Guo et al., 2018; Wu et al., 2020, 2021a,b; Yu et al., 2019), we set the maximum number of turns to be 10. Finally, for quantifying the correlations, we report Spearman's $\rho$. Significance testing is achieved by examining the p-value associated with $\rho$, which indicates the probability of an uncorrelated ranking producing a Spearman correlation as high as that observed.[1]

In the second set of experiments, we instead focus on evaluating the different categories of CPP predictors in terms of their effectiveness and how they perform in relation to the traditional QPP evaluation and corresponding metrics. For this purpose, we will examine both the score-based predictors and embedding-based predictors of Table 4.1 using the prediction horizon that is identified as the most reasonable based on the results of the first set of experiments. In this way, we obtain an equivalent set of results with Chapter 3 and test to what extent the examined predictor families can be generalised to a Conversational Image Recommendation setting. Therefore, we again use the Shoes (Berg et al., 2010; Guo et al., 2018) and Fashion IQ Dresses Wu et al. (2020) datasets and the same user simulator (Guo et al., 2018) as above. However, this time we will also compare between recommendation models; specifically between the GRU (Guo et al., 2018; Hidasi et al., 2015) and the EGE model (Wu et al., 2021b) as discussed in Section 2.2.2. In addition, we will correlate the predictor values using traditional ranking evaluation metrics (described in Section 2.4), similarly to the analyses we conducted for QPP in Chapter 3, in order to compare the performance of the equivalent predictors across tasks. In particular, we use NDCG@10 and MRR@10 of a given turn. Finally, following the common practice in QPP-based evaluation (Arabzadeh et al., 2021b; Datta et al., 2022b; Faggioli et al., 2023a,b; Meng et al., 2023), especially using recent Deep Learning query sets (Craswell et al., 2020, 2021), we will move from evaluating the entire number of image items in the datasets to sampling a subset of 200 images from each with diverse per-query effectiveness (see for example Chapter 3, where we used the TREC DL query sets with a limited number of queries). For this purpose, our second set of research question is as follows:

**RQ4.3** How do score-based CPP predictors compare to embedding-based CPP predictors in

[1] See also `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html`

short-term prediction?

**RQ4.4** How do CPP predictors in general compare across: (a) CRS models (b) evaluation metrics, and (c) datasets?

In what follows, we test our RQs and present the CPP results for the unsupervised evaluation.

Table 4.2: Results of single-turn predictors for short and long-term prediction of rank of target items at various turns. * denotes significant correlations; for Shoes, all correlations are significant, so * is omitted ($p < 0.05$).

| Prediction at turn $k$ with rank@turn10 | | | | Prediction at turn 2 with rank@turn $k$ | | | | Prediction at turn $k$ with rank@turn $k+1$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| k | top-1@k | mean@k | sd@k | rank@k | top-1@k | mean@k | sd@k | k, rank@k | top-1@k | mean@k | sd@k |
| Shoes | | | | | | | | | | | |
| 2 | -0.144 | -0.141 | -0.081 | 2 | **-0.405** | -0.385 | -0.059 | 2,3 | **-0.423** | -0.413 | -0.201 |
| 3 | -0.145 | -0.145 | -0.097 | 3 | **-0.423** | -0.413 | -0.201 | 3,4 | **-0.356** | -0.355 | -0.254 |
| 4 | -0.148 | -0.148 | **-0.105** | 4 | **-0.357** | -0.349 | -0.183 | 4,5 | **-0.318** | -0.317 | -0.211 |
| 5 | -0.155 | -0.153 | -0.089 | 5 | **-0.314** | -0.309 | -0.177 | 5,6 | **-0.293** | -0.292 | -0.180 |
| 6 | -0.165 | -0.165 | -0.093 | 6 | **-0.270** | -0.267 | -0.163 | 6,7 | **-0.254** | -0.254 | -0.135 |
| 7 | -0.173 | -0.173 | -0.100 | 7 | **-0.230** | -0.226 | -0.140 | 7,8 | **-0.235** | -0.234 | -0.126 |
| 8 | -0.178 | -0.177 | -0.073 | 8 | **-0.213** | -0.210 | -0.136 | 8,9 | **-0.208** | -0.207 | -0.067 |
| 9 | **-0.184** | **-0.183** | -0.064 | 9 | **-0.175** | -0.173 | -0.1149 | 9,10 | **-0.183** | -0.183 | -0.064 |
| 10 | -0.183 | -0.181 | -0.026 | 10 | **-0.144** | -0.141 | -0.081 | | | | |
| Dresses | | | | | | | | | | | |
| 2 | 0.012 | 0.003 | -0.036 | 2 | **-0.281*** | -0.279* | -0.161* | 2,3 | -0.248* | **-0.256*** | -0.197* |
| 3 | -0.017 | -0.015 | -0.004 | 3 | -0.248* | **-0.256*** | -0.197* | 3,4 | **-0.262*** | -0.257* | -0.075* |
| 4 | -0.045* | -0.047* | -0.014 | 4 | -0.187* | **-0.198*** | -0.173* | 4,5 | **-0.246*** | -0.239* | -0.038 |
| 5 | -0.055* | -0.051* | -0.007 | 5 | -0.128* | **-0.140*** | -0.137* | 5,6 | **-0.206*** | -0.198* | -0.008 |
| 6 | -0.063* | -0.063* | **-0.041*** | 6 | -0.079* | -0.092* | **-0.102*** | 6,7 | **-0.172*** | -0.168* | -0.034 |
| 7 | -0.069* | -0.072* | -0.033 | 7 | -0.052* | -0.067* | **-0.091*** | 7,8 | -0.139* | **-0.142*** | -0.044* |
| 8 | -0.075* | -0.076* | -0.021 | 8 | -0.039 | -0.051* | **-0.072*** | 8,9 | **-0.103*** | -0.101* | -0.000 |
| 9 | -0.073* | -0.071* | -0.018 | 9 | -0.005 | -0.018 | **-0.053*** | 9,10 | **-0.073*** | -0.071* | -0.018 |
| 10 | **-0.080*** | **-0.078*** | 0.003 | 10 | 0.0127 | 0.003 | **-0.036** | | | | |
| Shirts | | | | | | | | | | | |
| 2 | -0.092* | -0.089* | **-0.074*** | 2 | **-0.305*** | -0.298* | -0.141* | 2,3 | -0.297* | **-0.305*** | -0.201* |
| 3 | -0.124* | -0.119* | -0.033 | 3 | -0.297* | **-0.305*** | -0.201* | 3,4 | **-0.336*** | -0.326* | -0.03* |
| 4 | -0.145* | -0.137* | 0.011 | 4 | -0.264* | **-0.273*** | -0.192* | 4,5 | **-0.323*** | -0.308* | 0.019 |
| 5 | -0.148* | -0.142* | -0.016 | 5 | -0.228* | **-0.231*** | -0.157* | 5,6 | **-0.305*** | -0.293* | 0.018 |
| 6 | -0.139* | -0.134* | -0.003 | 6 | -0.198* | **-0.206*** | -0.155* | 6,7 | **-0.248*** | -0.238* | 0.026 |
| 7 | -0.152* | -0.150* | -0.003 | 7 | -0.166* | **-0.168*** | -0.122* | 7,8 | **-0.203*** | -0.196* | 0.017 |
| 8 | **-0.160*** | **-0.153*** | 0.031 | 8 | -0.1346* | **-0.135*** | -0.096* | 8,9 | **-0.192*** | -0.184* | 0.049* |
| 9 | -0.149* | -0.142* | 0.003 | 9 | **-0.120*** | -0.118* | -0.089* | 9,10 | **-0.149*** | -0.142* | 0.003 |
| 10 | -0.147* | -0.138* | 0.053* | 10 | **-0.092*** | -0.089* | -0.074* | | | | |

### 4.3.2 RQ4.1 - Results of Single-Turn Score-based Predictors

Table 4.2 shows the results for the three single-turn predictors, namely: the score of the top-ranked item at a given turn $k$ (denoted top-1@k); the mean value of all top-ranked items in the recommendation list at a given turn (mean@k); and the standard deviation values of the scores of all top-ranked items (sd@k). In the first group of columns, bold values denote the maximum correlation over all turns for the same predictor and the same ground truth value. For the other two sets of columns, bold values denote the highest performing predictor of the three examined single-turn predictors in the given evaluation setting for each turn – this is because comparison

of correlation values across turns (rows) is not possible, since the ground truth changes for each row. Note that the table is grouped into three sets of columns defining the prediction turn and the ground truth turn or the corresponding prediction horizon accordingly. Specifically, Prediction at turn $k$ with rank@turn10 addresses long-term prediction; the middle group, Prediction at turn 2 with rank@turn $k$, addresses whether prediction at an early turn can help identify success at early or late turns; finally, the third group, Prediction at turn $k$ with rank@turn $k+1$, addresses short-term prediction.

We first examine the first group of columns, which aims to determine the extent that the overall conversation can be successfully predicted (i.e. the ground truth is the rank of the target item at turn 10). Overall, the correlations[2] are weak (-0.184 is the strongest observed for Shoes, and -0.160 for Shirts; Dresses is lower still at -0.080), yet significant ($p < 0.05$). This suggest the difficulty of the long-term prediction task. We do observe that correlations are relatively higher as the prediction turn increases - thus indicating that it is easier to predict performance at turn 10 using evidence of the ranking at turn 10. Finally, among the predictors, the maximum score at each turn, along with the mean score, exhibit higher correlations the standard deviation. To answer RQ4.1(a), we cannot sufficiently predict long-term conversation performance using single-turn score-based predictors.

Turning next to the second group of columns, we observe comparatively stronger correlations. Indeed, the overall higher correlations suggests that predicting at turn 2 gives more accurate predictions, particularly when aiming to predict conversation performance at turn 2 or shortly thereafter. In particular, for the Shoes datasets, medium strength correlations of -0.423 are observed - these are in line with the best accuracy of some QPP predictors for adhoc search tasks (Cronen-Townsend et al., 2002; Shtok et al., 2009, 2010; Zhou and Croft, 2007). Correlations of -0.305 and $-0.281$ are observed for Shirts and Dresses, respectively. Among the predictors, top-1@k is again most successful on Shoes, but on Dresses and Shirts, where correlations are lower, the overall picture is less clear across different prediction horizons (i.e. as the ground truth $k$ is varied). For these datasets, mean is the most accurate for most values of $k \geq 2$. In general, when predicting conversation performance using single-turn retrieval scores, prediction becomes less accurate as the longevity of the prediction increases, thus answering RQ4.1(c).

Finally, the last set of columns of the table shows the correlation of the scores of each turn $k$ (as a predictor) when the effectiveness of the following turn $k+1$ is used as the ground truth (i.e. applying a short-term horizon). The scores of both the top-ranked item and the average score of the top-ranked items at turn $k$ sufficiently predict the rank of turn $k+1$, especially for early turns. This trend weakens as the number of turns increases, but the observed correlations remain quite high for some cases. For example, for Shoes, we start with a correlation of -0.423 (maximum

[2] In our analysis, we ignore the sign of the correlation - indeed, the observed correlations are negative, as our CRS system uses representation *distances* rather than similarities.

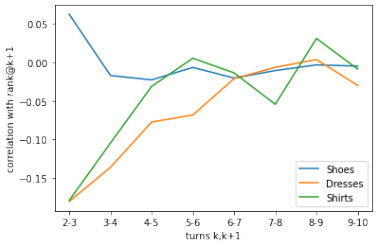
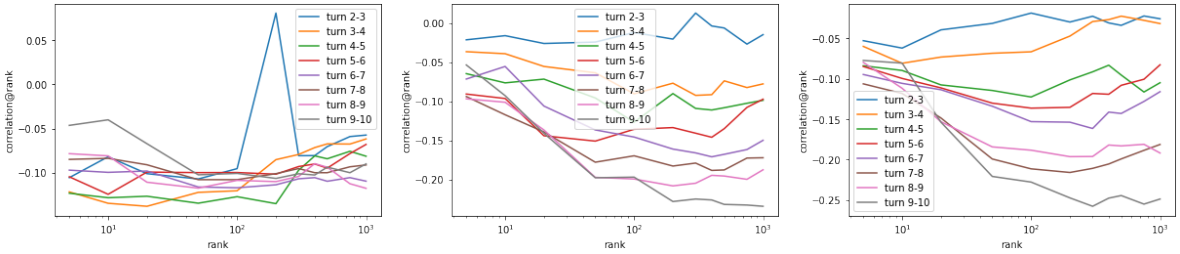



Figure 4.2: Results of the difference in the top-1 ranked item (maximum score) between pairs of consecutive turns as a consecutive turn CPP predictor for each of the datasets.




(a) Shoes (b) Dresses (c) Shirts

Figure 4.3: For each dataset, results for overlap of top-ranked items as a consecutive predictor for all pairs of turns $k, k+1$ for a number of rank cutoff values.

score) and -0.413 (average score) for turns $2,3$ and at turns $8-9$ the correlation is still -0.20. For Shirts, the maximum and average score of top items sufficiently predict the ranking of turn 3 at -0.30 and the score of turn 8 still at -0.20. Finally, although weaker than the other two datasets, the two predictors work reasonably well for Dresses, achieving a maximum value of -0.26 for predicting the rank of turn 3. These values suggest some evidence for short-term prediction when using single-turn score-based predictors, to answer RQ4.1(b).

Overall, we observe that there is some evidence for short (score of one turn predicting the rank of the following turn) and early prediction (a score of initial turn predicting the rank of some turns ahead). The score of the top-ranked item and the mean scores of the recommendation list are shown to be the most promising single-turn score-based predictors. However, contrary to previous QPP research (Shtok et al., 2009), the results for the standard deviation are not as encouraging. The results for long-term prediction are weaker, but still, the score of the initial turn is predictive of later stages. In general, prediction of the system performance (whether it finds the target in the context of a conversation) is possible by using single-turn score-based predictors, particularly for the success of the conversation at early turns and prediction of the next turn. Also, it is obvious that short-term (next turn prediction) is the most promising CPP setting, and this is the one we focus on in the following experiments.

### 4.3.3 RQ4.2 - Results of Consecutive-Turn Score-based Predictors

Figure 4.2 presents the results of our first score-based consecutive-turn predictor, namely the difference in maximum score (top-1 item) for each pair of turns $k, k+1$ when predicting the rank of the target item at turn $k+1$. Within the figure, each dataset is represented as a separate

curve. Considering the different datasets, for Shirts and Dresses, we observe a similar trend across turns, starting from a correlation of -0.18 (the maximum value obtained for this predictor) at turns 2-3, which gradually decreases as the number of turns increases. In contrast, Shoes does not achieve any correlation stronger than -0.016 at turns 3-4. Therefore, we observe only weak correlations for this predictor at short-term prediction, although some correlations are significant. To answer RQ4.2(a), using the scores of two consecutive turns, does not sufficiently predict conversation performance, and is indeed generally less effective than the single-turn predictors examined in RQ4.1.

Next, we test the overlap of top-ranked items (i.e., the size of intersection) between consecutive turns. We considered various rank cutoff values for calculating the overlap, ranging from rank 5 to rank 1000, and all pairs of turns. Figure 4.3 reports the observed correlations (y-axis), where each pair of turns is a curve, and the x-axis is the rank cutoff at which overlap is calculated. Recall that we expect that when the retrieved items are generally similar, this may be indicative that the CRS is reaching a stable conclusion of the likely relative items. If this occurs at a later turn, we may be further confident in the likely positive performance of the system.

On analysing Figure 4.3, we note that Dresses & Shirts (Figure 4.3(b) & (c), respectively) – which are both Fashion IQ datasets – we observe a strengthening trend in the correlations as we increase the rank cutoff value (more items are considered). This happens for all pairs of turns except the initial turn. In addition, the correlations are stronger for later turns than earlier turns, indicating that this predictor is more useful for later turns (as expected). Indeed, improved prediction at later turns is particularly notable, as this contrasts with our results in RQ4.1, where earlier prediction was more accurate.

On the other hand, for the Shoes dataset, the highest correlations are observed for turns 3-4 and 4-5, and for cutoff values at 50 and 100. The correlations for item overlap in Shoes are weaker than the other two datasets, contrasting with the observations in RQ4.1 (where Shoes exhibited higher correlations for the single-turn predictors than Dresses or Shirts). We note that, as a CRS dataset, Shoes is "easier" than Dresses (e.g. the GRU model can attain Mean Reciprocal Rank 0.2 at turn 10 on Shoes, compared to Mean Reciprocal Rank 0.075 at turn 10 on Dresses Wu et al. (2021b)). We postulate that early single-turn prediction works well on Shoes, as more conversations are answered at earlier turns; in contrast, on Dresses, more critiques are required for successful conversations, and the overlap-based evidence later in the conversation is therefore more useful for prediction.

Overall, these results suggest some weak-medium correlations (upto -0.25 $\rho$) on the overlap-based consecutive turn predictor, thereby answering RQ4.2(b).

### 4.3.4 RQ4.3 - Score-based vs Embedding-based CPP Predictors

For studying RQ4.3, we examine all potential unsupervised predictors that can be used in a CPP setting instantiated with Equation (4.4). First, we turn our attention to the GRU supervised

Table 4.3: Short-term horizon CPP results (prediction at turn $k$ with metric (MRR and NDCG) at turn $k+1$) for the Spearman's correlations of all examined unsupervised predictors for the GRU model for both datasets, Shoes and Dresses. * denotes significant correlations at significance level $\alpha = 0.05$. Bold denotes the best performing predictor in each row.

| Shoes | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| MRR | | | | | | | | | |
| k, rank@k | Mean | NQC | Max | AC-embs | WAND-embs | WD-embs | pairRatio | A-pairRatio | RV |
| 2,3 | -0.247* | **0.339*** | -0.235* | -0.023 | -0.010 | 0.011 | 0.006 | 0.095 | 0.211* |
| 3,4 | -0.212* | **0.288*** | -0.197* | -0.021 | 0.054 | 0.090 | 0.077 | 0.098 | 0.207* |
| 4,5 | -0.184* | **0.247*** | -0.176* | -0.072 | -0.040 | -0.091 | 0.113 | 0.093 | 0.199* |
| 5,6 | -0.174* | **0.273*** | -0.168 | -0.059 | 0.028 | -0.071 | 0.092 | 0.080 | 0.193* |
| 6,7 | -0.198* | **0.300*** | -0.195* | 0.057 | -0.143 | -0.087 | 0.122 | 0.134 | 0.226* |
| 7,8 | -0.202* | **0.291*** | -0.193* | -0.041 | -0.083 | -0.034 | 0.068 | 0.125 | 0.226* |
| 8,9 | -0.185* | **0.279*** | -0.180* | -0.041 | -0.042 | -0.011 | 0.274* | 0.234* | 0.221* |
| NDCG | | | | | | | | | |
| k, rank@k | Mean | NQC | Max | AC-embs | WAND-embs | WD-embs | pairRatio | A-pairRatio | RV |
| 2,3 | -0.141 | **0.232*** | -0.126 | -0.070 | -0.032 | -0.028 | 0.111 | -0.212* | 0.127 |
| 3,4 | -0.141 | **0.245*** | -0.122 | 0.069 | -0.007 | 0.018 | 0.124 | 0.059 | 0.159* |
| 4,5 | -0.078 | **0.155*** | -0.069 | 0.12 | -0.125 | -0.081 | 0.052 | 0.130 | 0.102 |
| 5,6 | -0.046 | 0.135 | -0.041 | -0.055 | -0.040 | -0.047 | **-0.167*** | -0.003 | 0.061 |
| 6,7 | -0.107 | **0.177*** | -0.103 | 0.069 | -0.088 | -0.065 | 0.019 | 0.070 | 0.152 |
| 7,8 | -0.071 | 0.154 | -0.06 | **-0.183*** | -0.069 | -0.108 | 0.026 | 0.038 | 0.110 |
| 8,9 | -0.091 | 0.156 | -0.086 | -0.087 | -0.073 | -0.057 | 0.063 | 0.074 | **0.161*** |
| Dresses | | | | | | | | | |
| MRR | | | | | | | | | |
| k, rank@k | Mean | NQC | Max | AC-embs | WAND-embs | WD-embs | pairRatio | A-pairRatio | RV |
| 2,3 | -0.121 | 0.183* | -0.087 | 0.020 | 0.119 | **0.191*** | 0.106 | 0.070 | 0.015 |
| 3,4 | -0.233* | **0.280*** | -0.199 | -0.008 | 0.122 | 0.103 | 0.046 | -0.106 | 0.114 |
| 4,5 | -0.262* | **0.294*** | -0.243* | 0.120 | 0.044 | 0.045 | 0.110 | 0.076 | 0.140 |
| 5,6 | **-0.280*** | 0.270* | -0.269* | 0.105 | -0.023 | -0.026 | 0.079 | 0.085 | 0.178* |
| 6,7 | -0.244* | **0.269*** | -0.234* | 0.015 | 0.021 | 0.006 | 0.044 | 0.173* | 0.120 |
| 7,8 | -0.240* | **0.266*** | -0.233* | 0.019 | 0.010 | 0.005 | -0.007 | -0.044 | 0.146 |
| 8,9 | -0.201* | **0.250*** | -0.176 | 0.019 | -0.014 | -0.022 | 0.222* | 0.221* | 0.141 |
| NDCG | | | | | | | | | |
| k, rank@k | Mean | NQC | Max | AC-embs | WAND-embs | WD-embs | pairRatio | A-pairRatio | RV |
| 2,3 | -0.103 | 0.071 | -0.063 | **0.139** | 0.058 | 0.042 | 0.093 | -0.116 | 0.044 |
| 3,4 | -0.154 | 0.167* | -0.129 | -0.112 | 0.060 | 0.052 | **0.193*** | 0.176* | 0.103 |
| 4,5 | **-0.180*** | 0.135 | -0.160 | -0.122 | 0.068 | 0.068 | 0.087 | 0.057 | 0.098 |
| 5,6 | **-0.182*** | 0.154 | -0.166* | -0.037 | 0.016 | 0.013 | 0.137 | 0.057 | 0.088 |
| 6,7 | **-0.176*** | 0.172* | -0.158* | -0.059 | 0.029 | 0.017 | 0.086 | 0.049 | 0.096 |
| 7,8 | **-0.192*** | 0.179* | -0.175* | -0.086 | -0.006 | -0.007 | 0.017 | 0.008 | 0.137 |
| 8,9 | -0.160 | **0.171*** | -0.152 | 0.000 | 0.028 | 0.019 | 0.170* | 0.170* | 0.107 |

learning recommendation model. The GRU results are presented in Table 4.3, where at the first half, we see the Shoes results (NDCG and MRR, and in the second half, the results for Dresses. Overall, we see that among all predictors, NQC is the best-performing one in most cases. This contradicts the results presented in Section 4.3.2, where the top-1@k (maximum score) and the mean score of the top-recommended items were more effective in predicting (short-term) conversational performance. We believe that this contradiction is related to the change of setting, as now we are closer to a QPP setting with fewer target items. Still, Mean and Max are quite effective for correlations with MRR, sometimes for earlier and sometimes for later turns. This implies that the pattern observed in Section 4.3.2, where single-turn score-based predictors works better for earlier turns is not fully replicated; instead, performance is improved across various turns. Overall, switching from a setting where all target items of a dataset were considered to the more query set-based evaluation of sampled items with varying effectiveness levels improves correlation values for score-based predictors at multiple time points of a conversation, especially for NQC.

Table 4.4: Short-term horizon CPP results (prediction at turn $k$ with metric (MRR and NDCG) at turn $k+1$) for the Spearman's correlations of all examined unsupervised predictors for the EGE model for both datasets, Shoes and Dresses. * denotes significant correlations at significance level $\alpha = 0.05$. Bold denotes the best performing predictor in each row.

| Shoes | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| MRR | | | | | | | | | |
| k, rank@k | Mean | NQC | Max | AC-embs | WAND-embs | WD-embs | pairRatio | A-pairRatio | RV |
| 2,3 | -0.092 | **0.282*** | -0.041 | -0.155 | -0.124 | 0.072 | 0.217* | -0.198* | 0.165 |
| 3,4 | -0.052 | **0.241*** | -0.027 | -0.055 | -0.037 | -0.010 | -0.089 | -0.176* | 0.098 |
| 4,5 | 0.083 | 0.044 | 0.095 | -0.039 | **-0.193*** | -0.180* | -0.100 | -0.018 | 0.136 |
| 5,6 | 0.090 | 0.029 | 0.127 | -0.124 | -0.084 | -0.108 | 0.171 | -0.080 | **0.184*** |
| 6,7 | 0.139 | -0.023 | **0.155** | -0.048 | -0.087 | -0.055 | 0.091 | -0.126 | 0.140 |
| 7,8 | 0.173* | -0.064 | **0.193*** | -0.013 | -0.042 | -0.042 | 0.063 | -0.028 | 0.044 |
| 8,9 | **0.152** | -0.064 | 0.168 | -0.011 | -0.035 | -0.037 | -0.036 | -0.123 | 0.115 |
| NDCG | | | | | | | | | |
| k, rank@k | Mean | NQC | Max | AC-embs | WAND-embs | WD-embs | pairRatio | A-pairRatio | RV |
| 2,3 | -0.100 | 0.180* | -0.035 | -0.118 | **-0.233*** | -0.216* | 0.195 | -0.106 | 0.231* |
| 3,4 | -0.107 | 0.176* | -0.051 | 0.090 | -0.212 | -0.188* | -0.080 | -0.113 | **0.291*** |
| 4,5 | 0.026 | 0.111 | 0.062 | -0.178* | -0.176* | -0.091 | 0.095 | -0.024 | **0.273*** |
| 5,6 | 0.104 | 0.141 | 0.131 | 0.051 | -0.204* | -0.157 | 0.080 | -0.034 | **0.205*** |
| 6,7 | 0.133 | -0.028 | 0.146 | -0.068 | -0.192* | -0.178* | 0.100 | -0.119 | **0.197*** |
| 7,8 | 0.155 | -0.023 | 0.179* | 0.029 | **-0.193*** | -0.177* | 0.110 | -0.121 | 0.138 |
| 8,9 | 0.141 | -0.039 | 0.155 | -0.111 | -0.038 | -0.025 | 0.034 | **-0.171*** | 0.122 |
| Dresses | | | | | | | | | |
| MRR | | | | | | | | | |
| k, rank@k | Mean | NQC | Max | AC-embs | WAND-embs | WD-embs | pairRatio | A-pairRatio | RV |
| 2,3 | 0.048 | 0.078 | 0.080 | 0.000 | 0.072 | **0.092** | 0.025 | -0.065 | -0.030 |
| 3,4 | 0.038 | **0.087** | 0.084 | 0.009 | 0.043 | 0.061 | 0.070 | -0.020 | 0.012 |
| 4,5 | 0.044 | **0.116** | 0.066 | -0.013 | 0.041 | 0.044 | -0.021 | -0.093 | 0.040 |
| 5,6 | -0.018 | 0.094 | -0.001 | **0.110** | -0.036 | -0.026 | 0.095 | 0.035 | 0.098 |
| 6,7 | -0.037 | **0.195*** | -0.025 | -0.002 | -0.041 | -0.037 | 0.095 | -0.046 | 0.098 |
| 7,8 | -0.008 | **0.185*** | 0.009 | -0.049 | -0.007 | -0.003 | 0.042 | -0.025 | 0.061 |
| 8,9 | -0.061 | **0.149** | -0.023 | 0.022 | -0.047 | -0.048 | 0.010 | 0.043 | 0.090 |
| NDCG | | | | | | | | | |
| k, rank@k | Mean | NQC | Max | AC-embs | WAND-embs | WD-embs | pairRatio | A-pairRatio | RV |
| 2,3 | -0.101 | **0.127** | -0.094 | -0.082 | -0.078 | -0.063 | 0.015 | 0.109 | 0.096 |
| 3,4 | -0.077 | 0.111 | -0.054 | **-0.112** | -0.088 | -0.08 | 0.033 | -0.004 | 0.103 |
| 4,5 | -0.060 | 0.095 | -0.044 | **-0.122** | -0.083 | -0.084 | 0.040 | -0.037 | 0.093 |
| 5,6 | -0.124 | **0.185*** | -0.101 | 0.084 | -0.113 | -0.110 | 0.079 | 0.112 | 0.149 |
| 6,7 | -0.119 | 0.139 | -0.103 | 0.003 | -0.130 | -0.133 | 0.053 | **0.150** | 0.137 |
| 7,8 | -0.113 | **0.139** | -0.102 | -0.022 | -0.096 | -0.106 | 0.132 | 0.027 | 0.129 |
| 8,9 | -0.096 | **0.166** | -0.088 | 0.108 | -0.055 | -0.074 | 0.034 | 0.093 | 0.129 |

On the other hand, the picture is less consistent for embedding-based predictors. Specifically, the correlations for AC-embs and the two network-based metrics (WAND-embs and WD-embs) are very low. This indicates that predictors that are based entirely on the semantic relations between retrieved items cannot fully capture the underlying relationships with prediction rankings. As for our own proposed dense coherence-based predictors, namely pairRatio and A-pairRatio, they are mainly useful in predicting later turns (particularly the final turn) when using MRR (and NDCG for Dresses). In other words, when considering the contrast of top/bottom rank semantic information and their interaction with feedback information, we can obtain some indication of conversational performance right before the conversation is coming to an end. This is in line with our intuitions in Section 4.2.1, that as the turns progress, the system learns a better representation of the target item. Finally, we consider the Reciprocal Volume (RV), which was found to outperform other unsupervised predictors by Faggioli et al. (2023a) in a conversational search setting. While the RV values are encouraging for correlating with MRR for the Shoes dataset,

for the rest of the cases are considerably lower and are outperformed by score-based predictors. This indicates the the hyperspace formed by the feedback-retrieved items combination has a different shape in our case, and it might not be as consistent as with TREC query sets.

Shifting our attention to the EGE model results, we note that score-based predictors are not as effective as with GRU. Specifically, Mean and Max values are consistently low, while NQC is only slightly effective for predicting early turns for Shoes and later turns for Dresses. Despite some encouraging results, in general, correlations are lower than their corresponding values in QPP settings. As for the embedding-based predictors, the values are generally low, except for specific cases: (i) WAND-embs for Shoes in a few turns, (ii) A-pairRatio for both datasets when correlating with NDCG for later turns, (iii) RV is the only predictor performing well for shoes, especially when correlating with NDCG early to mid-turns. This is important, as it indicates a promising use of embedding-based predictors for a conversation in CRSs.

Overall, we observe that score-based predictors outperform embedding-based predictors especially for the less effective CRS model (GRU), while embedding-based predictors show some more promising results for the more effective EGE model, thus answering RQ4.3. Still, these results are not consistent. In general, there is no predictor that outperforms all others across all cases.

### 4.3.5 RQ4.4 - Sensitivity of CRS models, Evaluation Metrics and Datasets

For RQ4.4, we now turn our attention to the differences between evaluation settings based on our choices, in particular the CRS models, metrics, and datasets. We first answer RQ4.4 (a) by comparing how the different predictors compare across CRS models. In particular, we observe the same pattern as in Chapter 3, where the increasing effectiveness and complexity of a retrieval model results in reduced correlation values. This implies that the predictors are unable to capture the underlying complexity of the interactive process in a more advanced recommendation model. In addition, embedding-based predictors, and especially RV, become more effective when moving from GRU to EGE, while for GRU, score-based predictors and especially NQC more sufficiently predict conversational performance.

To answer RQ4.4 (b), we compare the performance of the same predictors between the two evaluation metrics, namely NDCG and MRR. In general, for the GRU model, we do not observe major deviations between the two metrics for the same predictors, with the most notable differences being for RV Shoes and Mean Shoes. For the EGE model, we observe a similar pattern; some examples of more marked differences are RV for both Shoes and Dresses and WAND-embs Shoes. Finally, we compare between datasets to answer RQ4.4 (c). In general, correlations are higher for Shoes compared to Dresses. Still, in some cases, correlations for Dresses are higher for some score-based predictors mainly for later turns.

### 4.3.6 Insights from Unsupervised CPP predictors

In this section, we have presented a novel framework for conversational performance prediction (CPP) that aims to detect the factors that indicate effective performance by taking into account the multi-turn aspect of the task of conversational interactive image retrieval. In this regard, we proposed a number of predictors that can be used for both short-term and long-term prediction, and explored the retrieval scores and retrieved items, of both a single turn and consecutive turns. We conducted our analyses on widely-used relative captioning datasets for conversational recommendation systems (CRS) and examined the extent to which our proposed predictors are indicative of the ranking of the users' target items in the recommendation list. In our analyses of single-turn predictors, we found that examining the score of the top-ranked items had a medium correlation with the effectiveness of the conversation, particularly the effectiveness at early turns. Indeed, we observed a Spearman's $\rho$ of 0.423 on the Shoes dataset, which is comparable to correlations observed for standard QPP predictors on adhoc search tasks (Cronen-Townsend et al., 2002; Shtok et al., 2009, 2010; Zhou and Croft, 2007). However, these single-turn predictors became less useful at predicting the success of later turns. On the other hand, among consecutive turn predictors, simply examining the overlap of the retrieved lists had a weak-medium correlation with late turn effectiveness on two out of our three datasets.

Consequently, we examined a wider variety of unsupervised predictors using evaluation setting more similar to traditional QPP. In this case, we observed some deviations from the original CPP results. For example, the correlations for score-based predictor became more consistent, while in some cases, they are still outperformed by embedding-based predictors. While the weak-medium correlations observed for our simple unsupervised predictors of different families provide some promising results, it is still obvious that overall the correlations are significantly lower than QPP settings. This suggests that there is significant scope to extend this work, for instance by introducing supervised predictors. In particular, we assume that correlating a per-query CPP predictor value with the per-query effectiveness metrics is not the optimal way to measure CPP performance. This is due to the fact that a simple correlation might not be able to capture the underlying nature of a successful conversation. For that reason, we aim to extend our analyses to a classification task that aims to predict whether a conversation would fail, as well as testing the efficacy of interventions for failing conversations. In the next section, we present our supervised approach along with a supervised predictor that gradually learns the representations of the top-recommended items of the various turns.

## 4.4 Supervised Conversational Performance Prediction (Supervised CPP)

In the previous section, we introduced our CPP Framework for predicting conversational failures. In particular, we were more interested in *post-retrieval* predictors, which focus on the result list of the top-ranked retrieved documents use their relevance scores (Cronen-Townsend et al., 2002; Diaz, 2007; Shtok et al., 2009, 2010; Webber et al., 2010; Zhou and Croft, 2007) or their semantic relations (Arabzadeh et al., 2021a; Diaz, 2007; Faggioli et al., 2023a). For this purpose, we adapted various QPP predictors to our CPP task, and evaluated them on different prediction horizons, of which short-term was found the most effective. This notion is related to the view of QPP as prediction using reference lists (high association with *pseudo-effective* reference lists and low association with *pseudo-ineffective* lists is a indicative of effectiveness (Shtok et al., 2016)). Therefore, our proposed *Conversational Performance Prediction (CPP)* framework, as introduced in Section 4.2, applies QPP predictors in a conversational multi-turn setting, and is used to predict the rank of a target item at a certain turn. However, our first attempt was an unsupervised approach, evaluated using correlational measures. Our extended experiments with unsupervised predictors revealed that correlations might not be the ideal way to measure a complex relationship of conversational performance and the identification of a target image item at a top rank. Therefore, in this section, we aim to correctly classify conversation failures, through application of CPP in a supervised setting. Specifically, we conduct conversation performance prediction as a *classification task*, with a view to correctly classifying if a given conversation will result in the user's target item being successfully retrieved or not. For this purpose, we examine a variety of CPP predictors in the new evaluation setting: (i) score-based predictors, (ii) coherence or embedding-based predictors as examined in Section 4.2, (iii) our proposed supervised predictor that learns the representations across turns and produces a classification score. Below, we present how we derive these predictors.

### 4.4.1 Supervised CPP Definitions

As mentioned in Section 4.2, in our framework, we are interested in predicting performance in the context of recommendation models at the conversation level. Therefore, we differentiated between predicting at the shorter-term (using predictor turns at turn $k$ and predicted turns at turn $k+1$), and at the longer-term (using predictor turns at each turn $k$ and predicted turns at turn 10). Still, some findings indicated that long-term prediction does not work under this specified evaluation setting, while short-term predictions provide small to medium correlations. For this reason, we consider a further specification for CPP: In particular, we treat CPP as a classification task, where we consider conversation success as the *return of the target item by a given rank* either *at given turn* or *by a given turn*. The success is determined by the resulting label of the target item(s), which can be either found or not found. We now adapt the notations defined in

Section 4.2 to adapt our CPP definitions for the classification setting.

Specifically, for a conversation $C$ consisting of $k$ turns of user feedback utterances or critiques $f_1, \dots f_k$, and ranking of retrieved items $r_1, \dots r_k$, we define a classifier which aims to predict if conversation $C$ will be successful or not as follows:

$$cls(X_{C,k}) \rightarrow \{0,1\} \tag{4.6}$$

where $X_{C,k}$ is the feature representation for a given conversation at a given turn k, which can predict the label of a conversation in a binary classification. Note that for the purpose of this task, we treat CPP as a binary classification problem, since we are interested in whether the target item is returned or not. Therefore, we leave extensions of success definitions which can be operationalised with multi-label classification for future work. The classifier uses the feature representation of the conversation to predict the resulting success of these features for a conversation and therefore, the label of each target. As a feature representation $X_{C,k}$, we apply various predictors following both our unsupervised CPP definitions and the QPP predictor families. Similarly to the unsupervised setting, we focus on adapting the different groups of post-retrieval QPP predictors. Next, we explain how we derive each of them.

At a first stage, the simplest method is to examine the score-based post-retrieval query performance predictors originally used as unsupervised QPP predictors. Specifically, let $S_{c,k}$ be the scores of the retrieved items at a given turn $k$. Then we can calculate a feature representation at a given turn $k$ based on the retrieved scores, as follows:

$$X_{C,k}^{single} = [\max(S_{c,k}), \text{mean}(S_{c,k}), \text{stddev}(S_{c,k})] \tag{4.7}$$

Indeed, $\text{stddev}(S_{c,k})$ can be interpreted as NQC (Shtok et al., 2009), while $\max(S_{c,k})$ is the equivalent to the MAX score (Roitman et al., 2017a). While each one of these score-based values was used as an unsupervised predictor, their combination in order to produce a final score based on various indicators is more relevant to previously-proposed supervised QPP predictors such as the WPM estimator (Roitman et al., 2017a) and Neural-QPP (Zamani et al., 2018). However, when making performance predictions at any turn $k > 1$, we can access both the current retrieved item ranking, as well as the previous *historical* rankings. Therefore, a richer feature representation for turn $k$ uses the predictors calculated at all previous turns $(1 \dots k-1)$, as well as the present turn. Therefore, the above single-turn supervised predictor now becomes:

$$\begin{aligned} X_{C,k}^{multiple} = [ \quad & \max(S_{c,1}), \dots, \max(S_{c,k}), \\ & \text{mean}(S_{c,1}), \dots, \text{mean}(S_{c,k}), \\ & \text{stddev}(S_{c,1}), \dots, \text{stddev}(S_{c,k}) \quad ] \end{aligned} \tag{4.8}$$

Still, as already mentioned in Chapter 3, examining the embedded representations of the re-

trieved documents provides valuable information about the semantic relations either between the documents or between the queries and documents. Therefore, predicting the performance of a conversation at any turn or at the end of a dialogue can still be predicted by capturing these underlying relations. For this reason, we propose a (simple) supervised version of the embedding-based predictors already introduced in the QPP for dense retrieval context (see Chapter 3) and further developed for unsupervised CPP in Section 4.2. Following the intuitions for the feature representations in Equation (4.7), we can obtain a semantic equivalent for single-turn prediction as follows:

$$X_{C,k}^{single} = [CPP_{function}(\Phi_{c,k})] \tag{4.9}$$

where $\Phi_{c,k}$ is the embedded representation of the retrieved items at turn $k$, and $CPP_{function}(\Phi_{c,k})$ corresponds to any embedding-based CPP predictor at that turn. Consequently, for multi-turn predictions, we update Equation (4.9) can be updated as:

$$X_{C,k}^{multiple} = [CPP_{function}(\Phi_{c,1}), \ldots, CPP_{function}(\Phi_{c,k})] \tag{4.10}$$

where we use the embedded representation-based predictor up to turn $k$ to predict turn $k+1$. Note the difference with Equation (4.8), where we used multiple indicators to create a supervised predictor. Instead, in Equation (4.10) and in Equation (4.9), we only use a single CPP value per turn to predict the label of a given conversation. This is because each of the embedding-based predictors is based on the intuition of coherence and essentially captures the same underlying relationship among items. In addition, since the underlying semantic information of the same retrieved item between different turns can be autocorrelated, we introduce a new predictor in the embedding-based family that accounts for this problem. Specifically, we apply a Lasso-based classifier, which uses the L1 norm regularisation and a shrinkage factor $\lambda$, which results in some of the features to be set to zero. In this way, only the important dimensions of the feature representations contribute to the prediction of the label of the conversation. Specifically, we have a new predictor as:

$$X_{C,k}^{multiple} = [Lasso(\Phi_{c,1}), \ldots, Lasso(\Phi_{c,k})] \tag{4.11}$$

where the Lasso loss is given by $Error(Y - \widehat{Y}) + \lambda \sum_1^n |w_i|$; still, instead of the Mean Squared Error, we use this function as a classifier to calculate the resulting predictive accuracy on the test set. Therefore, Equations (4.9), (4.10), and (4.11) can be instantiated by any of the embedding-based predictors (see in Chapter 3 and Section 4.2): WAND-embs and WD-embs, pairRatio and A-pairRatio, and RV (Faggioli et al., 2023a).

Finally, inspired by neural supervised QPP predictors (Arabzadeh et al., 2021b; Datta et al., 2022b; Hashemi et al., 2019; Zamani et al., 2018), we create our own CPP predictor that learns the representations of the retrieved items using the classification labels on the train set and makes predictions on the test set. For this purpose, we use a linear Auto-Encoder (AE) that learns an

overall output function $h_{w,b}(X_{C,k})$ which is the result of the output layer or the reconstructed embedded representation values. In particular, $w$ is the set of input parameters of the retrieved item representations and $b$ is the model bias. AE aims to produce a reconstructed representation of these embedded representations $\hat{X_{C,k}}$ that is as similar as possible to the input representation $X_{C,k}$ based on the activation function applied to its hidden layer. This is done with an Encoder-Decoder structure, which compresses the input representation into a lower dimensional space and then reconstructs it, which produces a Mean Squared Error (MSE) = $(h_{w,b}(X_{C,k}) - X_{C,k})^2$ corresponding to the reconstruction error at the output layer. Our intuitions for this network structure in the conversational setting are the following:

- To predict the top-ranked item of a recommendation model (the item that the user sees), it is sufficient to use the top-ranked item for training.
- To predict an entire ranking of a recommendation model (the item that the user sees), we need to train on the full set of top-ranked items, since their distributions look more similar.
- Over time (turns), to predict the success of a conversation, we can use the retrieved item representations of all turns up to a given turn, and this can increase predictive performance compared to semnatic information from one turn.

Therefore, for a single-turn prediction, we use:

$$X_{C,k}^{single} = [(h_{w,b}(X_{C,k}) - X_{C,k})^2(\Phi_{c,k,1})] \tag{4.12}$$

where we examine the top-ranked item of the recommendation list. In turn, for the multi-turn prediction, we use a richer representation as follows:

$$X_{C,k}^{multiple} = [(h_{w,b}(X_{C,k}) - X_{C,k})^2(\Phi_{c,1}), \ldots, (h_{w,b}(X_{C,k}) - X_{C,k})^2(\Phi_{c,k})] \tag{4.13}$$

where we examine the top-k representation of all rankings up to turn $k-1$ to predict turn $k$. In the following, we experiment to determine the utility of our supervised CPP framework.

### 4.4.2 Overview of Experimental Setup

First, we examine the single-turn prediction using Equations (4.7), (4.9), and (4.12), and consequently, the multi-turn predictors using Equations (4.8), (4.11), and (4.13). For the score-based predictors, as already specified in Equations (4.7) and (4.8), we use the mean, maximum score, and the standard deviation of the retrieved item scores and use the scikit-learn implementation of the Random Forest classifier[3]. For the embedding-based multi-turn predictor as specified in Equation (4.11), we select among the coherence-based predictors the one that showed the most

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

promising performance in Section 4.2, namely Reciprocal Volume (RV) (Faggioli et al., 2023a), and we apply the L1-based classifier with a regularisation factor $\lambda$ equal to 0.1 by adapting the scikit-learn implementation of the Lasso Regression[4]. For the corresponding single-turn embedding-based predictor based on Equation (4.9), we use a Random Forest-based classifier for RV, since we only use a single predictor value to predict the ranking, and therefore, applying regularisation does not add anything. For both Random Forest and L1-based classifiers, we split the available conversations 70% for training, 30% for testing.

Finally, for our new AE-based CPP predictor, we use an Auto-Encoder with classification capabilities by adding two losses: (i) An MSE reconstruction loss, which forces the network to output a representation as similar as possible the input representation by producing its compressed version, (ii) A classification loss; here, we use the cross-entropy (CE) loss, which takes the compressed representation and target labels and calculates the negative log-likelihood loss (this leads the Encoder to output a compressed representation that aligns with the target class). We also add a loss multiplier to control the contribution from each loss adding equal weight to reconstruction and classification accuracy. We use a linear activation on the first hidden layer and a ReLu on the second hidden layer, and we train the model with an Adam optimizer with a learning rate of 0.01 for a total of 100 epochs. For single-turn prediction, we predict the performance of a conversation by simply examining the top-ranked item of turn $k$ to predict a ranking (or the top-ranked item) of turn $k+1$, and therefore, we use the contents of one turn for training and the rest for testing. This contradicts the traditional concept of allocating more data for training than testing, but in this case, it is according to our CPP short-term evaluation scenario. In contrast, for multi-turn prediction, we inevitably use more data for training, as we consider the top-100 returned item representations of all turns up to $k$ to predict turn $k+1$. We report Accuracy as measure of classification performance. More specifically, in this section, we answer the following research questions:

**RQ4.5** How do the different types of CPP predictors (score-based, embedding-based, representation learning-based) perform in single-turn prediction?

**RQ4.6** How do the different types of CPP predictors (score-based, embedding-based, representation learning-based) perform in multi-turn prediction?

### 4.4.3 RQ4.5: Supervised Single-turn CPP Prediction

First, we examine our proposed supervised CPP predictors in the single-turn setting. For this, we look at Tables 4.5 and 4.6. First, we examine the results of the GRU model in (Table 4.5; this table contains the predictive accuracy results on the test set of each classifier as described in Section 4.4.2, where each group of columns indicates a different prediction rank cutoff (ranks 1, 20, and 100). For single-turn prediction, we use the retrieved items of turn $k$ (which is denoted as *train* turn in the table) and we use the ranking (recommendation list) of turn $k+1$ to make

[4] `https://scikit-learn.org/stable/modules/linear_model.html`

Table 4.5: Single-turn CPP Supervised Predictor Accuracy results for the GRU model. Results for the Shoes dataset are shown in the top part of the table, while the bottom part shows the results for Dresses. Each group of columns indicates a different prediction rank cutoff (ranks 1, 20, and 100). The first two columns indicate the turn used to produce the predictor (denoted train) and the single turn whose ranking is used for prediction (denoted test). In each group of columns, bold denotes the best performing predictor for that specific rank cutoff. In case all predictors obtain identical values in a certain cutoff, none of them is denoted with bold.

| | | found at rank 1 | | | found at rank 20 | | | found at rank 100 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Shoes | | | | | | | | | | |
| train | test | score | RV | AE | score | RV | AE | score | RV | AE |
| 2 | 3 | **0.90** | 0.88 | 0.88 | **0.68** | 0.60 | 0.56 | **0.77** | 0.63 | **0.77** |
| 3 | 4 | **0.88** | 0.77 | 0.84 | **0.70** | 0.63 | 0.48 | 0.67 | 0.65 | **0.78** |
| 4 | 5 | 0.82 | **0.83** | **0.83** | **0.65** | 0.53 | 0.52 | 0.70 | 0.60 | **0.81** |
| 5 | 6 | 0.80 | 0.67 | **0.83** | **0.63** | 0.48 | 0.57 | 0.77 | 0.70 | **0.81** |
| 6 | 7 | 0.78 | 0.77 | **0.84** | **0.63** | 0.55 | 0.59 | 0.77 | 0.77 | **0.80** |
| 7 | 8 | 0.80 | 0.70 | **0.82** | **0.68** | 0.63 | 0.58 | 0.75 | 0.75 | **0.79** |
| 8 | 9 | 0.77 | 0.73 | **0.81** | 0.52 | 0.50 | **0.59** | **0.70** | 0.68 | 0.22 |
| 9 | 10 | **0.78** | 0.70 | **0.78** | **0.60** | 0.55 | 0.57 | 0.73 | 0.67 | **0.77** |
| Dresses | | | | | | | | | | |
| train | test | score | RV | AE | score | RV | AE | score | RV | AE |
| 2 | 3 | **0.98** | **0.98** | 0.96 | 0.82 | 0.82 | **0.97** | 0.57 | 0.58 | **0.97** |
| 3 | 4 | **0.98** | 0.97 | 0.97 | 0.87 | 0.77 | **0.98** | 0.63 | 0.58 | **1.00** |
| 4 | 5 | **0.98** | 0.97 | 0.97 | 0.82 | 0.78 | **0.99** | 0.57 | 0.57 | **1.00** |
| 5 | 6 | **0.97** | 0.92 | **0.97** | 0.82 | 0.73 | **0.98** | 0.53 | 0.72 | **1.00** |
| 6 | 7 | **0.97** | **0.97** | 0.94 | 0.75 | 0.72 | **0.98** | 0.57 | 0.60 | **0.98** |
| 7 | 8 | **0.97** | 0.93 | 0.95 | 0.72 | 0.68 | **0.99** | 0.72 | 0.60 | **0.99** |
| 8 | 9 | **0.97** | 0.92 | 0.95 | 0.77 | 0.67 | **0.98** | 0.60 | 0.63 | **0.98** |
| 9 | 10 | **0.95** | 0.90 | 0.94 | 0.75 | 0.70 | **0.99** | 0.55 | 0.62 | **0.98** |

predictions (denoted as *test* turn). In particular, when predicting the top-ranked item (the cutoff which we denote as *found at rank 1*), we observe that the results are mixed across both datasets. For Shoes, all three types of predictors perform better at different turns, with AE scoring higher most of the times and especially as the turns progress. For Dresses, score-based predictors seem to perform highest, but with only marginal differences, since all predictors seem to perform equally across turns. For predicting a successful conversation by considering returned target items by rank 20 (*found at rank 20*), the results again diverge between datasets; for Shoes, there is a clear trend for a better performance of score-based predictors, while for Dresses, AE performs best across turns. Finally, when examining a ranking of the top-100 items (*found at rank 100*), our AE-based classifier performs best for both datasets, and especially for Dresses, the difference is notable with other predictors.

Next, we turn to the EGE results in Table 4.6. Again, when predicting successful conversations at the top-ranked item in the first group of columns, the results are mixed, with Shoes indicating a split between all predictors at various turns, while Dresses indicate that AE is per-

Table 4.6: Single-turn CPP Supervised Predictor Accuracy results for the EGE model. Notation as per Table 4.5.

| | | found at rank 1 | | | found at rank 20 | | | found at rank 100 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Shoes | | | | | | | | | | |
| train | test | score | RV | AE | score | RV | AE | score | RV | AE |
| 2 | 3 | **0.97** | 0.87 | 0.90 | **0.57** | 0.50 | 0.56 | 0.78 | 0.50 | **0.82** |
| 3 | 4 | **0.90** | 0.80 | 0.88 | **0.65** | 0.53 | 0.50 | 0.77 | 0.53 | **0.83** |
| 4 | 5 | 0.88 | **0.90** | 0.84 | 0.52 | **0.53** | 0.46 | **0.85** | 0.53 | 0.15 |
| 5 | 6 | 0.77 | 0.73 | **0.80** | 0.50 | **0.62** | 0.53 | 0.82 | 0.62 | **0.86** |
| 6 | 7 | 0.77 | 0.72 | **0.79** | **0.73** | 0.52 | 0.54 | 0.77 | 0.52 | **0.83** |
| 7 | 8 | **0.80** | 0.63 | 0.79 | **0.62** | 0.57 | 0.55 | 0.82 | 0.57 | **0.86** |
| 8 | 9 | **0.87** | 0.70 | 0.77 | **0.73** | 0.52 | 0.46 | 0.80 | 0.52 | **0.87** |
| 9 | 10 | 0.73 | 0.58 | **0.76** | 0.52 | 0.43 | **0.57** | 0.85 | 0.43 | **0.87** |
| Dresses | | | | | | | | | | |
| train | test | score | RV | AE | score | RV | AE | score | RV | AE |
| 2 | 3 | 0.97 | 0.97 | **0.98** | 0.82 | 0.77 | **1.00** | 0.60 | 0.48 | **0.99** |
| 3 | 4 | 0.95 | 0.92 | **0.97** | 0.77 | 0.62 | **1.00** | 0.58 | 0.60 | **1.00** |
| 4 | 5 | **0.95** | **0.95** | 0.94 | 0.77 | 0.73 | **0.99** | 0.55 | 0.55 | **1.00** |
| 5 | 6 | 0.92 | 0.90 | **0.95** | 0.77 | 0.68 | **0.99** | 0.57 | 0.52 | **0.99** |
| 6 | 7 | 0.90 | 0.90 | **0.95** | 0.77 | 0.65 | **0.99** | 0.42 | 0.43 | **0.99** |
| 7 | 8 | 0.87 | 0.85 | **0.93** | 0.77 | 0.72 | **0.98** | 0.53 | 0.43 | **0.99** |
| 8 | 9 | 0.87 | 0.85 | **0.94** | 0.73 | 0.67 | **0.99** | 0.48 | 0.50 | **0.99** |
| 9 | 10 | 0.87 | 0.87 | **0.95** | 0.72 | 0.75 | **0.95** | 0.53 | 0.53 | **0.95** |

forming best. As for the items found by rank 20, we observe a similar pattern with the GRU model: in the Shoes dataset, score-based predictors perform higher (with RV being the highest in the middle turns), while for Dresses AE shows the highest performance. Finally, looking at a ranking of the top-100 items, and similarly to Table 4.5, for both datasets, AE shows higher performance than both score-based and RV.

Overall, we do not observe large deviations between the two recommendation models (except for some turns in "found at rank 1"), unlike the results in Section 4.2, where we saw that the CPP correlations were significantly lower when moving from GRU (a less effective CRS model) to EGE (a more effective CRS model). Also, there is no large deviation between datasets, although in general, the accuracy is higher for Dresses than for Shoes. This contradicts the results in Section 4.2, where Shoes exhibited larger correlations. Indeed, while Dresses was found as a more difficult dataset in previous studies (Wu et al., 2021a,b), the supervised single-turn CPP results indicate that our proposed predictors can effectively predict a successful conversation by using its success label at various ranks. Finally, we note that our AE-based predictor performs best in various setting, and is consistently optimal for predicting a top-100 ranking. This is quite surprising: While learning a compressed representation of the top-ranked item is expected to perform well to predict the top-1 ranking, we see that it also shows promising results for a full ranking. This is encouraging as a less expensive supervised predictor for single-turn prediction

Table 4.7: Multi-turn CPP Supervised Predictor Accuracy results for the GRU model. Results for the Shoes dataset are shown in the top part of the table, while the bottom part shows the results for Dresses. Each group of columns indicates a different prediction rank cutoff (ranks 1, 20, and 100). The first two columns indicate the final turn up to which we use the contents to produce the predictor (denoted train, which means CPP values up to turn $k$) and the single turn whose ranking is used for prediction (denoted test, which means the turn that comes after the multi-turn predictor). In each group of columns, bold denotes the best performing predictor for that specific rank cutoff. In case all predictors obtain identical values in a certain cutoff, none of them is denoted with bold.

| | | found at rank 1 | | | found at rank 20 | | | found at rank 100 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Shoes | | | | | | | | | | |
| train | test | score | RV | AE | score | RV | AE | score | RV | AE |
| 2 | 3 | 0.90 | **0.97** | 0.88 | 0.68 | 0.57 | **0.98** | 0.77 | 0.77 | **1.00** |
| 3 | 4 | 0.90 | **0.92** | 0.84 | 0.72 | 0.55 | **1.00** | 0.78 | 0.78 | **1.00** |
| 4 | 5 | 0.85 | **0.87** | 0.83 | 0.75 | 0.53 | **0.98** | 0.63 | 0.77 | **1.00** |
| 5 | 6 | 0.82 | **0.85** | 0.82 | 0.63 | 0.48 | **0.99** | 0.75 | 0.77 | **1.00** |
| 6 | 7 | 0.82 | **0.83** | 0.82 | 0.70 | 0.52 | **0.99** | 0.80 | 0.80 | **1.00** |
| 7 | 8 | 0.78 | **0.83** | 0.81 | 0.72 | 0.63 | **1.00** | 0.82 | 0.80 | **1.00** |
| 8 | 9 | 0.80 | **0.82** | 0.80 | 0.68 | 0.60 | **0.99** | 0.70 | 0.75 | **0.99** |
| 9 | 10 | **0.82** | **0.82** | 0.77 | 0.72 | 0.63 | **1.00** | 0.73 | 0.78 | **1.00** |
| Dresses | | | | | | | | | | |
| train | test | score | RV | AE | score | RV | AE | score | RV | AE |
| 2 | 3 | 0.98 | 0.98 | 0.98 | 0.87 | 0.87 | **0.99** | 0.57 | 0.62 | **0.97** |
| 3 | 4 | 0.98 | 0.98 | 0.98 | 0.85 | 0.83 | **0.99** | 0.62 | 0.57 | **1.00** |
| 4 | 5 | **0.98** | **0.98** | 0.97 | 0.83 | 0.85 | **1.00** | 0.60 | 0.62 | **1.00** |
| 5 | 6 | 0.97 | 0.97 | 0.97 | 0.82 | 0.82 | **1.00** | 0.70 | 0.55 | **1.00** |
| 6 | 7 | 0.97 | 0.97 | 0.97 | 0.78 | 0.80 | **1.00** | 0.55 | 0.53 | **0.98** |
| 7 | 8 | 0.97 | 0.97 | 0.97 | 0.78 | 0.78 | **1.00** | 0.60 | 0.50 | **0.99** |
| 8 | 9 | **0.97** | **0.97** | 0.96 | 0.83 | 0.83 | **0.99** | 0.68 | 0.52 | **0.98** |
| 9 | 10 | **0.97** | **0.97** | 0.96 | 0.75 | 0.78 | **0.99** | 0.63 | 0.47 | **0.98** |

that only needs the top item of an embedded representation already contained in the CRS models. To answer RQ4.5, score-based predictors perform quite well on predicting the top-item and items found at rank 20 only for Shoes, while the new AE-based predictor shows promising performance across more settings, and RV (which performed quite well in Section 4.2) performance is encouraging only in a few cases for single-turn prediction.

### 4.4.4 RQ4.6: Supervised Multi-turn CPP Prediction

We now examine the supervised CPP performance for multi-turn prediction. For this, we look at Tables 4.7 and 4.8 for GRU and EGE, respectively, where we use the contents of all turns up to turn $k$ to produce the predictor (denoted train, which means CPP values up to turn $k$) and the single turn whose ranking is used for prediction (denoted test, which means the turn that comes after the multi-turn predictor. Similarly to single-turn prediction, we use three different rank

Table 4.8: Multi-turn CPP Supervised Predictor Accuracy results for the EGE model. Notation as per Table 4.7.

| | | found at rank 1 | | | found at rank 20 | | | found at rank 100 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Shoes | | | | | | | | | | |
| train | test | score | RV | AE | score | RV | AE | score | RV | AE |
| 2 | 3 | **0.97** | **0.97** | 0.90 | 0.57 | 0.53 | **0.99** | 0.80 | 0.78 | **1.00** |
| 3 | 4 | 0.88 | **0.90** | 0.88 | 0.55 | 0.48 | **1.00** | 0.80 | 0.82 | **1.00** |
| 4 | 5 | **0.92** | 0.88 | 0.84 | 0.67 | 0.52 | **0.99** | 0.83 | 0.87 | **1.00** |
| 5 | 6 | 0.82 | **0.85** | 0.80 | 0.57 | 0.52 | **1.00** | 0.82 | 0.82 | **1.00** |
| 6 | 7 | **0.87** | 0.83 | 0.79 | 0.65 | 0.53 | **1.00** | 0.82 | 0.82 | **1.00** |
| 7 | 8 | **0.83** | 0.82 | 0.79 | 0.57 | 0.48 | **1.00** | 0.85 | 0.85 | **1.00** |
| 8 | 9 | 0.80 | **0.82** | 0.77 | 0.57 | 0.48 | **0.99** | 0.83 | 0.83 | **1.00** |
| 9 | 10 | **0.82** | 0.78 | 0.76 | 0.57 | 0.42 | **1.00** | 0.88 | 0.88 | **1.00** |
| Dresses | | | | | | | | | | |
| train | test | score | RV | AE | score | RV | AE | score | RV | AE |
| 2 | 3 | 0.97 | 0.97 | **0.98** | 0.82 | 0.83 | **0.99** | 0.60 | 0.68 | **1.00** |
| 3 | 4 | 0.95 | 0.95 | **0.97** | 0.78 | 0.78 | **1.00** | 0.52 | 0.55 | **1.00** |
| 4 | 5 | 0.95 | 0.95 | **0.96** | 0.77 | 0.78 | **1.00** | 0.52 | 0.55 | **1.00** |
| 5 | 6 | 0.92 | 0.92 | **0.96** | 0.75 | 0.75 | **1.00** | 0.55 | 0.53 | **1.00** |
| 6 | 7 | 0.90 | 0.90 | **0.94** | 0.77 | 0.78 | **1.00** | 0.53 | 0.55 | **1.00** |
| 7 | 8 | 0.87 | 0.87 | **0.95** | 0.77 | 0.78 | **1.00** | 0.52 | 0.57 | **1.00** |
| 8 | 9 | 0.87 | 0.87 | **0.94** | 0.80 | 0.78 | **1.00** | 0.50 | 0.55 | **1.00** |
| 9 | 10 | 0.87 | 0.87 | **0.95** | 0.77 | 0.77 | **1.00** | 0.58 | 0.53 | **1.00** |

cutoffs to make predictions. First, we observe that for the deeper rank cutoffs ("found at rank 20" and "found at rank 100"), our AE-based predictor outperforms the other two types across both CRS models and datasets. This is the case where the predictor learns the entire top-100 ranking representations of all turns up to turn $k$ used to predict the ranking of turn $k+1$. Therefore, we expected the predictor to show higher performance for these deeper rankings, and especially the predictions of top-100 ranking result in a train-test pattern correspondence. On the other hand, for predicting successful conversations on the top item ("found at rank 1"), we observe mixed results. For Dresses, our AE-based predictor still shows optimal performance for EGE, while for GRU it performs equally well with the other predictors (all of them show high performance). For Shoes, we observe that either score-based predictors or RV perform best for EGE, while for GRU, our L1-based version of RV is the best-performing predictor. Therefore, we see that using regularisation on the multi-turn representations of this embedding-based predictor is particularly useful in these cases and using the important features helps in the top-item prediction. On the other hand, score-based predictors do not offer that much added value in supervised CPP multi-turn prediction. In general, learning a compressed representation of the top-ranked items from previous turns (AE) shows improved performance for CPP multi-turn prediction, the L1-based RV shows improved performance in some cases for prediction the top item, and score-based predictors can still be used as a strong baseline, thus answering RQ4.6.

## 4.5 CPP Conclusions

In this Chapter, we have presented our novel Conversational Performance Prediction (CPP) (Section 4.2) framework, which proposes indicators of effective performance of a conversation in Conversational Image Recommendation dialogues. In our framework, our main focus is to take into account the multi-turn aspect of the task of conversational interactive image retrieval task. For this purpose, we proposed a number of CPP predictors that cover various evaluation settings. In particular, we started with simple score-based predictors and used them for both short-term and long-term prediction, both for single-turn and consecutive-turn prediction (Section 4.2.1). We performed our experiments using the widely-used relative captioning datasets for conversational recommendation systems (CRS) and examined the extent to which our proposed predictors are indicative of the ranking of the users' target items in the recommendation list (Sections 4.3.2, 4.3.3). In short, we found that for single-turn prediction, score-based predictors are quite effective at early turns, but for consecutive-turn prediction, examining the overlap of the retrieved lists had only a weak-medium correlation with late turn effectiveness on two out of our three datasets. Furthermore, we extended our evaluation to a more similar setting to traditional QPP (Sections 4.3.4 and 4.3.5) by using a subset of target items and also including embedding-based predictors that were already examined in Chapter 3. We found that in general, score-based predictors outperform embedding-based predictors in most cases, but embedding-based predictors show improved performance for earlier turns in short-term settings (RQs 4.3 and 4.4). Overall, the observed correlation values for CPP were much lower than the corresponding values observed in traditional QPP settings (Arabzadeh et al., 2021a,b; Cronen-Townsend et al., 2002; Datta et al., 2022b; Faggioli et al., 2021b, 2023a; Hashemi et al., 2019; Meng et al., 2023; Roitman et al., 2017b; Shtok et al., 2009, 2010; Zamani et al., 2018; Zhou and Croft, 2007). These results indicated that CPP evaluation could be extended to a more appropriate setting that better captures the nature of the task. In particular, in Section 4.4, we introduced a supervised version of our CPP framework, where we based our analyses on a classification task that aims to predict whether a conversation would fail, as well as testing the efficacy of interventions for failing conversations. We further proposed a new embedding-based supervised predictor (inspired by supervised QPP predictors) that learns a compressed representation of the retrieved item(s) of previous turn(s). In our experiments (Sections 4.4.3 and 4.4.4), we found that using classifier-based evaluation and the predictive accuracy of a predictor on the test set more effectively captures the underlying nature of a multi-turn conversation and shows high accuracy across both single-turn and multi-turn predictions (RQs 4.5 and 4.6). In addition, we found improved performance across multiple rank cutoffs and predictors. Overall, in this chapter, we have validated the second claim of the thesis statement that *we can predict the effectiveness of a ranking of items in a Conversational Recommendation Systems (CRS), which are also based on learned embedded representation of images, where user feedback takes the place of a textual query. Indeed, by introducing a framework of Conversational Performance Prediction (CPP),*

*we can predict the degree of success of a conversation by a CRS - such success can be predicted over a short or long time horizon, thereby predicting current user satisfaction or overall satisfaction of a conversation..*

While our CPP framework shows promising results for predicting the success of a conversation in our task of interest, there are a number of limitations that have not been addressed in this Chapter. In particular, we addressed different levels of dialogue success based either on a correlation or labels of success with a classifier. In these cases, dialogue failure is considered to be a failure of the system to deliver the target user's item by a certain turn, and this is what we tried to predict. Still, system failure is not the only reason why a target item is not returned. It might be the case that the target item does not exist in the catalogue. Therefore, when making predictions about conversation success, we need to differentiate between system failure and unavailability. Furthermore, while the user can have a more strict information need, which is restricted in a single target item, sometimes a user can have a more flexible need, which can be equally satisfied with multiple alternative items. Indeed, QPP evaluation is more reliable when there are more than a single relevant document (Datta et al., 2022b; Faggioli et al., 2023b). In Chapter 5, we collect real user opinions about the relevance of alternative items to given target items for the Shoes and Dresses datasets. This dataset has better knowledge of alternative relevant items, and we use that to reevaluate CPP in Chapter 6.

# Chapter 5

# Evaluating User Simulators with Alternatives

In Chapter 4, we introduced our framework of Conversational Performance Prediction (CPP), which extends the traditional Query Performance Prediction in search tasks to a Conversational Image Recommendation setting. In this way, we predicted the rankings resulting from multimodal CRS, and more generally, we showed how we can predict the degree of success of a conversation with a CRS, namely the second proposition of the thesis statement (Section 1.2). In this regard, we confirmed our hypothesis that by considering the multi-turn aspect of our task of interest, we can predict conversation success over multiple horizons - in a shorter or longer timeframe, and therefore, differentiate between current user satisfaction and overall user satisfaction in a dialogue context. Still, in order to be able to adequately predict conversational performance, we need to take into account the context in which it takes place. This means that we need to make predictions in a realistic setting that sufficiently mimics a real life user shopping scenario. In this chapter, we account for this requirement by collecting real user opinions about alternative fashion items with an aim to evaluate CPP under this new alternatives-based evaluation setting. In particular, we experimentally test our third hypothesis of the thesis statement, which revolves around the following: *Furthermore, by obtaining user opinions about the relevance of items, we improve the completeness of the evaluation mechanism by identifying alternatives recommendations for existing target items, which could be used to both inform the user simulator and therefore improve the overall evaluation of CRS systems.* This addresses **Limitation 1a)** (*A system trying to find a single item that is already known by the user contradicts the recommendation intuition*) and **Limitation 1b)** (*Focusing on a single target item without having any more options to choose from highly restricts system performance. In this way, there is a chance that the system keeps repeating the same recommendations, thus influencing the distribution of items being recommended.*).

Indeed, as we mentioned in Section 2.2.2, for the purpose of training and evaluation, many CRSs use a reinforcement learning approach (Guo et al., 2018), which ideally requires access

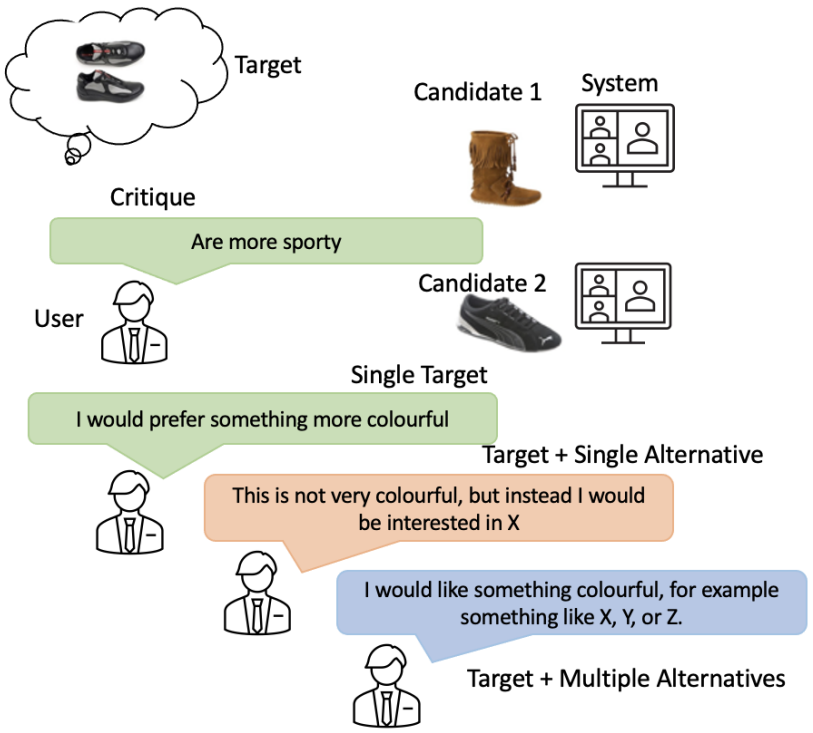



Figure 5.1: Example of a fashion Conversational Image Recommendation scenario. At each turn, the user provides natural language feedback on a candidate item. In existing systems, users are assumed to have a specific target in mind (green). Instead, the presence of a single alternative (orange) or multiple alternative (blue) items can guide the system to find a target of a certain type.

to a large amount of data from interactions with the environment (Shi et al., 2019). To compensate for the lack of human data at large scale, a common solution is to rely on interactions with a *simulated* user to train the system, and similarly evaluation is done in an offline setting. In Section 2.2.2, we also described our setting of interest, namely Conversational Image Recommendation. In this context, a user has a desired or *target* item "in mind" and provides relative feedback on the current or *candidate* item at each turn. An example of such a context, where a user interacts with a CRS, is shown in Figure 5.1. Here, the system starts with a random suggestion at the first turn, and the user provides textual feedback on the recommended item, aiming to guide the system to their assumed target item. The process and the different variants of the task were described in detail in Sections 2.2.2 and 2.2.2. In all cases, the overall setting is suggestive of a known-item type of task (Broder, 2002), where the target item is assumed to be defined and exist in the item catalogue.

The next turn in Figure 5.1 displays the options that a hypothetical (simulated) user could behave in existing CRSs. Currently, a simulator supports the first case (shown in green), where the same pattern follows the previous turn after Candidate 1. Instead, a user could react in a more flexible way; they could ask the system for another item that is not their initial target, but is quite similar to it, and provide one ("instead I would be interested in X") or more ("something like X, Y, or Z") additional preferences. In other words, in existing user simulators, there is no option

for a (simulated) user to request for one (orange) or more (blue) alternatives to their target item, which is something that happens in a real life shopping scenario. This problem connects back to the *recommendation scenarios* presented already in Section 2.2.3 (Limitations 1a and 1b). In this chapter, we provide a solution to the scenarios or strategic changes of users' preferences, which could have an impact on the evaluation of system performance (through changes in generated feedback).

Indeed, we argue that the current fashion recommendation evaluation setting presents some limitations: First, the *realisticity* of an interaction does not necessarily aid user experience. Specifically, the user is assumed to be infinitely patient, and willing to interact with the CRS for a large number of turns until the target item is found. This setting is not representative of a real user experience, where a user might become frustrated. To complement this, a simulated user is assumed to be single-minded, meaning that it is not flexible enough to change its strategy or initial plan. On the contrary, recommender systems are typically used to aid exploratory user behaviour (Broder, 2002; O'Brien, 2006), and therefore, by persisting on a single desired item, users are not exploring the product space. Moreover, unlike information retrieval systems, which are evaluated using test collections that aim to provide a reasonably *complete* coverage of relevant documents (Craswell et al., 2020; Dalton et al., 2020a), recommender systems suffer from a lack of completeness (Chaney et al., 2018,?; Jadidinejad et al., 2020). In particular, as mentioned in Section 2.1, search engines use *pooling* of documents retrieved from a various systems and a per query *relevance judging* of pooled documents to obtain more complete assessments (Craswell et al., 2020; Dalton et al., 2020a). For example, the MSMARCO test collection has thousands of queries contains shallow judgements, while the TREC Deep Learning track provided $\sim$100 queries with deeper judgements (Craswell et al., 2020). In this regard, it was found that the sparse MSMARCO assessments are not a suitable replacement for more complete assessments (MacAvaney and Soldaini, 2023). Similarly, the presence of more reliable relevance judgments for CRS would benefit the reliability of their evaluation. In this chapter, we show that more relevance judgments for CRS items can be obtained by directly asking users about their alternative preferences, thus allowing the user to update their preferences during the dialogue. A summary of the limitations of existing user simulators (or relative captioners, as described in Section 2.2.2) can be seen in Table 5.1, along with the reasons why we decide to address them. Overall, our aim is to provide a more realistic setting from the user side (in terms of how a user expresses their preferences), which in turn, leads to a more exploratory nature of a CRS, and contributes to a more reliable and generalisable conclusions.
In short, this chapter makes the following contributions:

- The first extended dataset for fashion recommendation that contains labels about the presence of sufficient alternatives for a number of known target items by real users on different fashion item categories. The fashion categories (shoes and dresses) are derived from two popular fashion-related CRS datasets, namely Shoes (Berg et al., 2010; Guo et al., 2018)

Table 5.1: Summary of the limitations in existing user simulators in a Conversational Image Recommendation setting.

| Limitation | CRS function | Reason for addressing |
|---|---|---|
| infinitely patient user | allows evaluation for multiple turns | realisticity of a CRS |
| single-minded user | single know target image item | to aid user's exploratory behaviour |
| lack of item completeness | not established way of assessing relevance | to aid reliability of evaluation |

and FashionIQ (Wu et al., 2021a). In this way, we contribute to evaluation completeness for relevance and create a parallel with information retrieval

- Consequently, the first user simulator that uses relevance judgments about alternatives or as we term, a *meta-user simulator*, which wraps an existing user simulator to provide feedback for possible alternatives items. Our proposed improved user simulator allows simulated users not only to express their preferences about alternative items to their original target, but also to change their mind and level of patience.

- A study of existing CRS models evaluated with and without alternatives. In this regard, whether a user changes their mind with respect to their target item in the previous turn might be influential for system performance. In other words, opting for an alternative item would produce feedback with slightly altered semantic information, thus leading the system to produce a considerably improved ranking of items.

Importantly, the main findings of this chapter are: (i) CRS performance improvements are observed across all three CRS systems and evaluation metrics. In particular, using the knowledge of alternatives by the simulator can have a considerable impact on the evaluation of existing CRS models. In other words, the existing single-target evaluation of CRS underestimates their effectiveness, and when simulated users are allowed to instead consider alternatives, the system can rapidly respond to more quickly satisfy the user. (ii) The exact level of patience of a user before switching to an alternative does not impact performance to a great extent. (iii) Users tend to select an alternative, and the earlier they do this, the more they get earlier increase in satisfaction. (iv) Introducing alternatives results in a slight reordering of the different CRS models. The rest of the chapter is organised as follows: We present some related work on Conversational Recommendation, user simulators, and data pooling in Section 5.1, and introduce our alternative-based user simulator in Section 5.2. Furthermore, we introduce the way we collected our alternatives extended datasets in Section 5.3. We continue with our experimental setting, evaluation measures, and results in Section 5.4, and finish with some conclusions in Section 5.5.

## 5.1 Related Work

The main differences of Conversation Image Recommendation from text-based CRS are that: (a) recommended items are displayed as images, and more specifically, a user only sees the top-

ranked image item at each turn, and (b) the setting does not differentiate between user feedback and providing the user need, since the user feedback is provided in natural language form and describes specific attributes of the desired item. Examples of such systems were provided in Section 2.2.2. The common assumption in these approaches is that the dialog with the user proceeds with a narrowly-defined target item. However, by being single-minded and not allowing for any other option, the evaluation setting is not realistic, while the user cannot adequately explore the product space. Moreover, while displaying the top-ranked images resembles the context of online shopping, a more natural conversation usually involves a user that changes their mind, and does not wish to interact for an infinite number of turns. In contrast, in our work, we allow the user to reconsider after a certain threshold and provide alternative options. In the rest of this section, we present some related work to our proposed approach for collecting alternative opinions for our meta-simulator. For this purpose, we start with some information on user simulation in CRSs (Section 5.1.1) and continue with existing approaches in information retrieval that use pooling to create realistic and representative relevance judgments (Section 5.1.2).

### 5.1.1 User Simulation for Evaluating CRS

In this section, we present a more detailed view of user simulators in conversational systems. Training CRS systems in a multi-turn setting requires a large amount of data (Li et al., 2016; Shi et al., 2019). To compensate for the increased need for real users, user simulators are used as a surrogate of human behaviour (Li et al., 2016; Shi et al., 2019). Indeed, several approaches have been proposed that employ user simulators in interactive systems (Chung, 2004; Griol et al., 2013; Owoicho et al., 2023; Sun et al., 2023; Verberne et al., 2015; Zhang and Balog, 2020; Zhang et al., 2022). For example, Owoicho et al. (2023) observed improved performance of mixed-initiative conversational search systems with multiple rounds of simulated user feedback, while Sun et al. (2021) simulated user satisfaction with training data from annotators who judged the level of satisfaction of each turn from the dialogue context. As for CRSs, recent work on user simulators builds on an agenda-based framework that uses push and pull operations to update the user needs per turn (Balog, 2021; Schatzmann et al., 2007; Vakulenko et al., 2019; Zhang and Balog, 2020). For Conversational Image Recommendation, the state-of-the-art simulator framework is explained in Section 2.2.2.

For evaluating CRS systems, some approaches compare the resulting dialogue with human dialogues using different performance metrics (Sun et al., 2021; Wu et al., 2021a; Zhang and Balog, 2020; Zhang et al., 2022). In addition, some simulation approaches collect annotated datasets for training CRSs. However, they are usually limited to rating the level of dialogue success or user satisfaction (Sun et al., 2021). On the other hand, our work is focused on extending the completeness of the ground truth by introducing more options to the target space. In other words, we aim to enrich our simulated users with a target *group* instead of single target items in a relative captioning setting. In that sense, our approach is similar to Sun et al. (2023), where

they assume an analogical thinking of users, i.e., users comparing new items with prior knowledge. Still, they do not necessarily provide other preference options. Instead, the basis of our work is to inform the user simulator with alternatives in order to provide more helpful feedback and simulate a realistic scenario.

### 5.1.2 Data Pooling and Evaluation Completeness

In general, the evaluation of recommender systems is plagued by a lack of completeness, as typically past interactions are "replayed" and the prediction ability of the recommender system to predict the hidden "future" interaction(s) is measured by classical evaluation measures such as MRR and NDCG. This tends to favour systems that behave similarly to the system originally deployed when the user interactions were collected (Chaney et al., 2018,?; Jadidinejad et al., 2020). In contrast, search engine evaluation uses test collections (Sanderson et al., 2010), which combine two techniques for obtaining a more *complete* coverage of relevant documents: the *pooling* of documents retrieved by a number of diverse effective systems; and the explicit judging of the relevance of all pooled documents to a user's query. Incomplete test collections are well known to result in unreliable evaluation (Buckley and Voorhees, 2004; Buckley et al., 2007). Recently, Craswell et al. (2020) found a good correlation between evaluation using thousands of single known relevant queries versus using deeply judged TREC queries, however, pseudo-relevance feedback techniques have been shown to work on the latter but not the former (Wang et al., 2023).

Pooling and assessing is typically not used for recommendation, as the user's exact information needs are not clear. However, for fashion-based CRS, where the user has a target item in mind, we argue that it is possible to ask a 3rd party assessor to reasonably interpret their need and consider what other items they may have considered as relevant alternatives. In this way, we develop more complete test collections for fashion-based CRS (using alternative target items), and a more realistic user simulator that can make use of these alternatives during evaluation.

## 5.2 Proposed Approach: Simulated Users with Alternatives

In this section, we outline our proposed approach for an alternative-based user simulator that expresses user needs in an adaptive way. In particular, we build on the state-of-the-art user simulators used in Conversational Image Recommendation, as introduced in Section 2.2.2, and we extend the user's target space from a single target to a target group, based on real user relevance judgments. More specifically, as mentioned in Section 2.1.2, conversational recommendation with image items in the fashion domain is another example setting of a dialog-based ranking task, where, at each interaction turn, the user provides a critique of the current recommendation, aimed at directing the recommender system towards their desired target item. This assumes a list of ranked image items at each turn, where the ranking of a given turn is updated based on

the user feedback received in the previous turn. At the same time, to stay in line with the *reinforcement learning (RL)* approach adopted by the task, which allows optimizing the decision process based on the long-term rewards (Shi et al., 2019), the system needs to be interacting with the environment, and obtaining many samples is hard by relying on real users (Li et al., 2016; Shi et al., 2019). This challenge is dealt with with human surrogates or the state-of-the-art user simulator, which is used for training and evaluating such systems. In our task of interest, a user simulator is termed as the *relative captioner* (Section 2.2.2), whose generated feedback was found to correlate with human satisfaction (Guo et al., 2018). Therefore, in the remainder of the section, we first describe the user simulator in the existing conversational image recommendation settings, namely *relative captioning*. Then, we continue by introducing our meta-user simulator, which takes into account relevance judgments obtained from real users regarding alternatives to a given target image: We describe its conception and functionality, and specifically how it uses the knowledge of alternatives to inform the CRS about user preferences. In addition, we provide some explanations about the actions of the simulator and the system.

### 5.2.1 User-simulator based evaluation in CRS

We firstly start by providing some general notation providing the general principles of a user simulator. More formally, at a given interaction turn $k$, the user provides textual feedback $f_k$ on the current top-ranked candidate item $i_{k,1}$. Based on this feedback, the conversational recommendation system $\mathscr{C}()$ provides a new ranking of items in the next turn, i.e.: $\mathscr{C}(i_{k,1}, f_k) \rightarrow \{i_{k+1,1}, ...i_{k+1,n}\}$. In this regard, as briefly introduced in Section 2.4.2, a relative captioning dataset contains tuples of the following form: $\langle rep_{target}, rep_{cand}, tq_{cand,target} \rangle$ where $rep_{target}$ is a representation of the target item (for instance an image), $rep_{cand}$ is the current candidate item being presented to the user and $tq_{cand,target}$ is the critique by the user on the candidate, which directs the system more towards the target. Therefore, datasets of this type are used as training data for the simulator or relative captioner. In other words, a *User Simulator* using relative captioning is an object with a single function defined as

$$Usersim.critique(user, top\_ranked, target) \rightarrow \mathscr{T} \tag{5.1}$$

which takes as input the user's id *user*, the current top-ranked image, $top\_ranked$, and the user's actual target image $target$. It then calls a learned relative captioning model, which has been trained given $target$ to critique $top\_ranked$, and returns as output a text string describing the visual differences between $top\_ranked$ and $target$.

### 5.2.2 A Meta User Simulator for Evaluation with Relevant Alternatives

After describing the user simulators currently used in conversational image recommendation studies, our aim is to extend them in order to be able to take alternative relevant items as input. More specifically, we present a set of intuitions for a user that considers alternative items as follows:

> **Intuition 1 (I1)**: A user's patience when critiquing a single candidate item may run out after number of turns (the user gets frustrated when a certain amount of time is exceeded and their target item is not returned by the system).
>
> **Intuition 2 (I2)**: When a user selects an alternative item as a new target, they are influenced by the current item they see. In other words, the choice (or not) of an alternative item is influenced by the top-ranked candidate item at the given turn.
>
> **Intuition 3 (I3)**: Once an alternative item is chosen by the user, the existing relative captioner-based user simulator can be called with the alternative as a new target.

These intuitions are operationalised with a new updated user simulator, as shown in Figure 5.2, and will be detailed below.

The pseudocode for our meta-user simulator procedure is provided in Algorithm 1. More specifically, our *MetaUserSim* firstly requires knowledge of all possible alternative items for target items. This is akin to the "qrels" in test-collection based evaluation. Then, when the meta-user simulator is called, if the turn number exceeds the patience $tolerance$ parameter, then alternatives are considered (line 2, addressing in Intuition I1); Among all of the alternatives for a given target, we select the alternative that is closest in image similarity to the current top ranked image as the target (lines 5 & 6, addressing I2). The existing relative-captioning based user simulator is then asked to critique the current top-retrieved item, this time with respect to the newly selected target (line 8, I3); Note that we choose to consider the target as part of the alternatives, so that the ranker can make a choice between the nearest item at a later stage. Finally, we instrument Algorithm 1 (our proposed meta-simulator algorithm) to provide data about how often alternatives are chosen.

Using the new meta-simulator algorithm, the updated learned relative captioning model, can be written as:

$$Metausersim.critique(user, top\_ranked, all\_targets) \rightarrow \mathscr{T} \tag{5.2}$$

which has been trained given each $target_i$ to critique $top\_ranked$, and this time returns a text string describing the visual differences between $top\_ranked$ and either of the $target_i$. In this way, Metausersim is still a critiquing method using a user simulator, with the difference being that the assumed simulated user is more flexible, changes their mind, loses their patience their

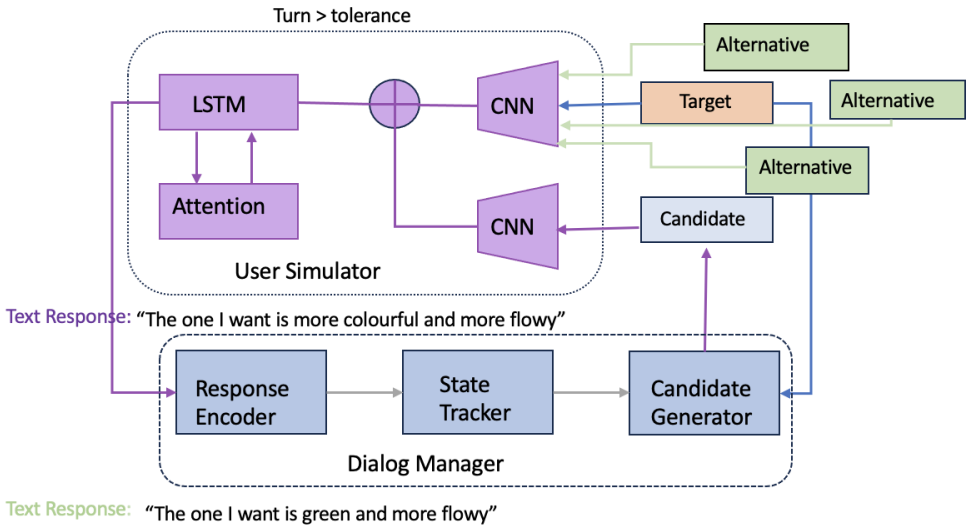



Figure 5.2: Schematic representation of our meta-simulator that uses alternatives to produce feedback.

**Algorithm 1** Meta-User Simulator

**Require:** $Usersim$: base user simulator for Conversational Image Recommendations
**Require:** $sims$: function to find the similarity of a set of image from a given query image
**Require:** $tolerance$: patience parameter referring to the turn a user starts to ask for an alternative
**Require:** $all\_alternatives$: A set of known alternative relevant images for all targets

**procedure** METAUSERSIM.CRITIQUE($turn, top\_ranked, target$)
  **if** $turn > tolerance$ **then**
    $alternatives = all\_alternatives[target]$
    $alternatives$.target($target$)
    $all\_dists = sims(alternatives, top\_ranked)$
    $target = alternatives[\arg\max(all\_dists)]$
  **end if**
  **return** Usersim.critique($turn, top\_ranked, target$)
**end procedure**

behaviour resembles more a real-user shopping scenario. Table 5.2 provides a summary of the intuitions of our met-simulator in comparison with the corresponding functionality of the base (non-alternative) simulator. Importantly, Figure 5.2 shows a schematic representation of our new proposed meta-simulator with alternatives. In particular, we consider the case when the user's patience is exceeded (turn > tolerance). In this case, each of the identified alternatives is also encoded together with the candidate and target images. Consequently, when a given turn exceeds the tolerance level and given a set of known alternatives to a target image, the simulator is asked to critique the candidate with respect to one of the alternatives, provided that one of them is close enough to the candidate (otherwise the simulated user will not opt for an alternative). Here we see that the feedback is modified (in green). This happens as the instruction from the user simulator is given to the Dialog Manager, which leads to the modification in the generated textual response or feedback. In the next section, we discuss how our dataset with alternatives

is created in order to be used as input data for the new simulator.

Table 5.2: Summary of differences of our meta-simulator from the base simulator (relative captioner) according to our proposed intuitions.

| Intuition | base simulator | alternative-based simulator |
|---|---|---|
| I1 | infinitely patient user (tolerance exceeds the final turn) | tolerance parameter (patience ran out after a given threshold) |
| I2 | user remains with the option of the initial target | user selects the most similar alternative to the given candidate |
| I3 | user critiques the top-ranked candidate item | user has the option to critique the alternative instead |

## 5.3 Enriching of CRS Datasets with Alternatives

For the purpose of training our meta-simulator, we assume that a dataset's representativeness is a crucial factor in training a user simulator, since it can have an impact on guiding the system, and in turn, its performance. Therefore, we need to ensure that we obtain a reliable dataset that follows some general principles in information retrieval. As mentioned in Section 5.1.2, information retrieval evaluation is done with TREC test collections (Craswell et al., 2020, 2021; Dalton et al., 2020a,b), where pooling is applied. In our case, we attempt to create a parallel evaluation setting for CRS systems, by taking into account the following intuition, according to which a smaller number of representative deeper relevance judgments is not interchangeable with thousands of shallower examples and recent techniques were found to work better for them (MacAvaney and Soldaini, 2023; Wang et al., 2023). Therefore, we believe that a careful selection of target items guarantees a representative sample. We now describe how we enriched two fashion CRS datasets with alternative judgements to create our dataset of alternative judgments consisting of different fashion categories (shoes and dresses).

### 5.3.1 Original Datasets

To build our dataset, we use the two popular datasets in the conversational fashion recommendation domain, namely the Shoes (Berg et al., 2010; Guo et al., 2018) dataset, and the FashionIQ Dresses dataset (Wu et al., 2021a), as mentioned in Section 2.4.2. In particular, Shoes contains 4658 test target images, while Dresses contains 2454 test images. Since these datasets were originally collected to train and evaluate CRS in the relative captioning setting (Guo et al., 2018; Wu et al., 2020), each target image is accompanied by a corresponding paired candidate image, as well as a relative critique or caption per candidate-target pair, which describes the relative visual differences between the candidate and target image pairs and are used as training data for the user simulator. For our task, we focus on the target images contained in the original datasets. We obtain *labels of relevance* (whether a set of candidate images are a sufficient alternative to a given target image) for a portion of the target images in the original datasets, which we treat as

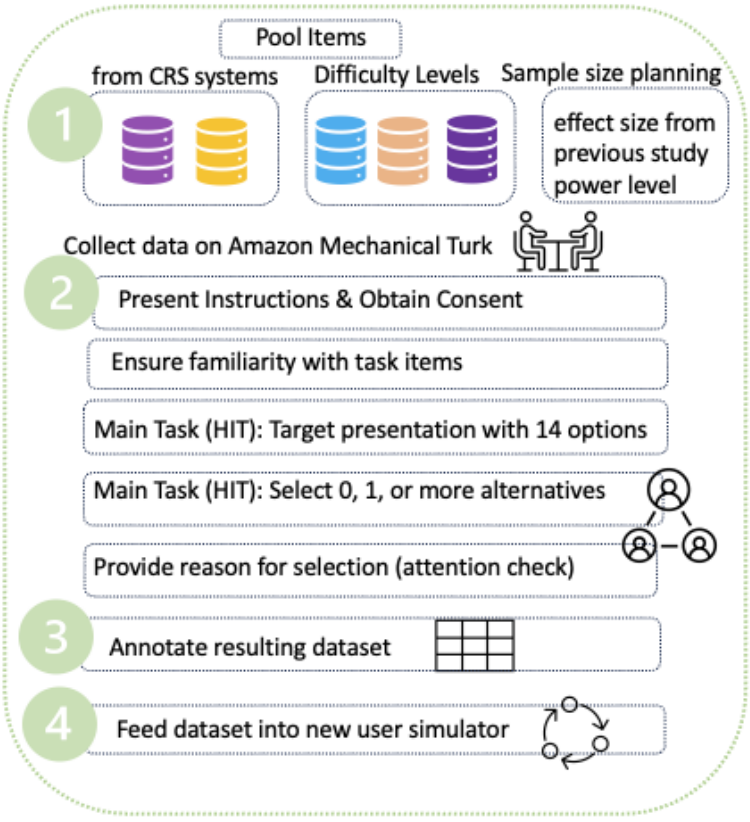



Figure 5.3: Schematic representation of the user study with both data pooling and data collection steps.

different *fashion categories*[1].

### 5.3.2 User Study Details

In this section, we describe the details of our user study, which can be further divided into two main stages: target pooling (described in section 5.3.2) and data collection (described in section 5.3.2). In particular, we use Amazon Mechanical Turk[2] to obtain assessments on alternative items. Target pooling was completed in August 2023, while data collection extended from 18 August 2023 until 4 October 2023. It involved several smaller test batches for each fashion category before the final deployment of the image set. A schematic representation of our user study in steps is presented in figure 5.3. Next, we describe further how each of the two sub-tasks was conducted.

#### Target Pooling

The purpose of the study was to collect relevance judgments for CRS systems in order to introduce a parallel to the test collection paradigm in information retrieval test collections. Therefore,

[1] Our new collected dataset with alternative labels of relevance for Shoes and FashionIQ dresses (shoes and dresses) together with the source code of our meta-simulator can be found at [2] https://www.mturk.com

we need to ensure that we select a number of representative target image items (as the queries in test collections) for which users will provide their opinions about relevant alternatives. This brings two requirements: (i) Select a sufficient amount of items to achieve a level of generalisability. (ii) Select items with prior knowledge derived from different systems - this ensures a variety in the sample of items that are more representative of the underlying population of items. To account for the first requirement, we estimate the required number of sampled target images with a power analysis (using the `pwr`R package (Champely et al., 2018)) using the reported correlation values of our Conversational Performance Prediction analysis (as described in Chapter 4) as effect sizes (by converting the correlation values to Cohen's *d* with the 'effectsize' R package (Ben-Shachar et al., 2022)), a significance level (alpha value) of $\alpha = 0.05$, to achieve a power of 90%. The power analysis estimation provided a number lower than 200 targets for each fashion category (shoes, dresses), but we opted for 200 from each to obtain a sufficient amount of targets. Table 5.3 shows the exact required sample size. We see that our sample of 200 targets from each dataset already exceeds the required sample.

To account for the second requirement, we select the target images by sampling 200 target items from each fashion category with varying levels of difficulty (we checked this by conducting a preliminary QPP analysis of the sampled items using score-based predictors (as introduced in Section 2.3.2). In addition, for assessment, we derive a pool of candidate images for each target by using existing state-of-the-art CRS models for conversational image recommendation, specifically GRU (Guo et al., 2018; Hidasi et al., 2015) and EGE (Wu et al., 2021b) (detailed further in Section 5.4.1 below). In particular, we select both the nearest neighbours (in their corresponding image embedding spaces) to the target (60%) and their top-retrieved images of the final evaluation turn (40%). We place more importance on the nearest neighbours, because we are more interested in the similarity of the images rather than how each CRS model ranks them. Specifically, we use the top-4 ranked nearest neighbours of each target item from each CRS model, and the top-3 top-ranked results from each CRS model at turn 10. This results in a total of 14 candidate images per target item. To ensure no duplicates, we checked how many items overlap between the two CRS models (both for nearest neighbours and retrieved results), and in case of common entries, we replaced them with additional items from lower ranks. In summary, we followed a detailed strategy for data pooling, which resulted in an amount of precisely estimated and representative target items from each fashion category, which already exceed both the more recent TREC Deep Learning (Craswell et al., 2020) test collection query sets by roughly four times and the TREC CAsT for conversational tasks (Dalton et al., 2020a,b).

### Data Collection

We conduct our study on Amazon Mechanical Turk, which has been used as a platform for data collection in several online studies for various conversational systems (Jurcıcek et al., 2011; Owoicho et al., 2023; Sun et al., 2021; Sun and Zhang, 2018) and CRS systems (Liu et al.,

Table 5.3: Summary of the required sample size of target image items from each original dataset resulting from the power analysis.

| Dataset | Spearman's $\rho$ | Cohen's $d$ | Required Sample |
|---|---|---|---|
| Shoes | -0.423 | -0.933 | 54 |
| Dresses | -0.281 | -0.585 | 128 |

2020; Zhang and Balog, 2020; Zhou et al., 2020), and also in our setting of interest, namely relative captioning, where the original datasets were obtained with crowd-sourcing (Guo et al., 2018; Wu et al., 2020). We take some additional steps to ensure the representativeness and the knowledge level of our sample. Specifically, participants are selected based on their location (US, to ensure an adequate level of English) and to the extent they could identify with a person who wears dresses or women's shoes (familiarity with and knowledge of the target items), and are paid based on the rules of Mechanical Turk. We obtained institutional ethical approval for the study, and we paid participants $0.63 for each MTurk task (or *HIT*) for a total duration of 3 minutes (this is above the living wage in our country), making a total cost of the study was $305 (we rejected only 3 HITs for spammy behaviour). In our study, we simulate a real user (online) shopping scenario. Participants are instructed that each presented target image is an item they want to buy. Simultaneously, participants see a set of candidate image items that could be a sufficient alternative to the target which, as instructed, is not available in the catalogue. The task is to select, out of the displayed set of candidates, the ones (if any) that best satisfy the user need as an alternative to the target item. Finally, participants are asked to indicate the reason for their selection (this also works as an attention check, as we ask them to respond with a full sentence and set a minimum required length). Each participant was allowed to complete one or more HITS. For each of the resulting alternative labelled fashion categories of our dataset, there were on average 3.5 identified relevant alternatives per target image. We performed a second round of assessment on 40 target items (10% from each dataset), and measured assessor agreement. We observed a Cohen's $\kappa$ between the two sets of judgements of 0.87, demonstrating a high level of agreement. In the following, we now analyse three representative fashion CRS systems using the alternatives-based user simulator from Section 5.2, and using the alternatives dataset for 200 target items. An example HIT for the Dresses dataset of our user study can be seen in Figre 5.4.

The collected data are used as input to the user simulator with alternative, which is a similar critiquing methodology. Our collected dataset also contains relevance assessment annotation as follows: (i) A csv file, where each row denotes the target image and the columns represent a relevance for each presented alternative option from 1 to 14, (ii) A csv file, with each row containing the image names (docnos) of all identified alternatives of a given target image included in the sample. Below we present our experimental setup.

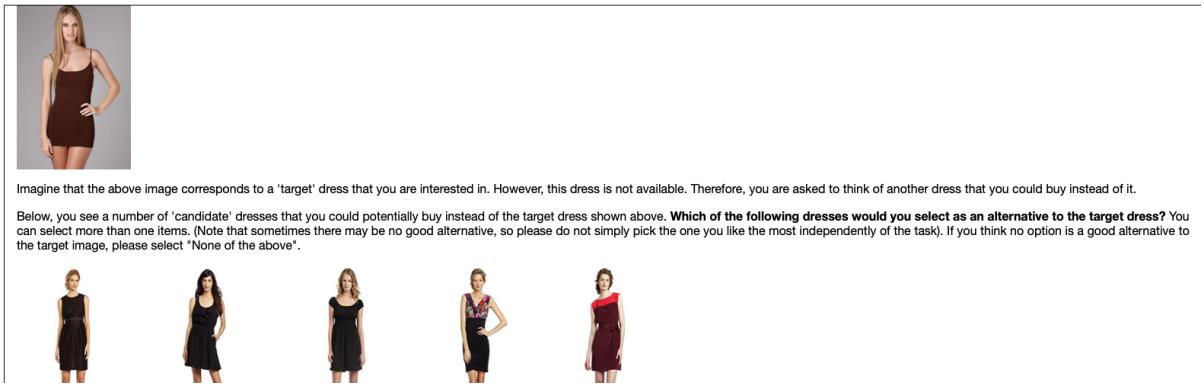



Figure 5.4: Example HIT (Amazon Mechanical Turn task) from or user study for the Dresses dataset. The target item appears at the top, while the worker is intructed to select one or more alternatives from the items appearing below as candidates.

## 5.4 Experiments

In this chapter, we investigate the impact of our alternative-based simulator on CRS performance. For this purpose, we test the performance of a number of CRSs with both simulators, as desribed in Section 5.2. We are also interested in how our introduced patience parameter further influences this performance. In addition, we check the resulting changes after applying the new simulator, such as the resulting ranking of CRSs and how frequently a user switches to an alternative in relation to how the system behaved previously. In this section, we experiment to address the following research questions:

**RQ5.1** What is the impact of using an alternative-based user simulator on the evaluation of existing CRS models?

This requires us to check the objective performance of the different CRSs with both simulators (using Equations (5.1) and (5.2)). This allows us to measure the relative improvement (if any) brought to the systems due to the new simulator.

**RQ5.2** (a) What is the impact of patience of an alternative-based user simulator and (b) how does it behave for each CRS model?

With RQ5.2 (a), we test the impact of the introduced tolerance parameter and check for any sensitivity that might be crucial for system performance. RQ5.2 (b) tests our hypothesis about the representativeness of the collected dataset and checks for the reliability of the evaluation setting, in the sense that a performance that is not extremely variable across systems indicates that our dataset is diverse. We check this in relation to other predictors, such as tolerance.

**RQ5.3** Does introducing patience change conclusions about what are the most effective models?

This RQ tests the relative ordering of the different systems and how this changes when alternatives are introduced.

**RQ5.4** How often do users prefer an alternative over their initial target item?

WIth this, we measure the amount of times a simulated user opts for an alternative instead of

the original target. To provide a more precise view, we compare this with how the system was doing before and after the user changed their strategy.

In the rest of this section, we provide an overview of the deployed CRS models (Section 5.4.1) and evaluation measures (Section 5.4.2) used in our experiments, and answer our RQs in Sections 5.4.3 to 5.4.6.

### 5.4.1 Setup: Conversational Recommendation Systems (CRS)

We deploy three existing CRS models using both the original relative captioning single-target evaluation setting and our own meta-simulator with alternatives. Each system retrieves 100 top items per turn and the conversation stops at turn 10 (the default final evaluation turn in the task (Guo et al., 2018). The CRSs are the following:

- A GRU model (Guo et al., 2018; Hidasi et al., 2015) with reinforcement learning (GRU-RL), which combines input representation with the historical information representation from the previous turn to produce an updated aggregated representation vector and is, therefore, optimised for maximising short-term rewards. For a detailed description of the model, please refer to Section 2.2.2.

- A GRU variant trained with supervised learning (GRU-SL), i.e. lacking short-term rewards during training. Due to this limitation, this GRU variant was found to be less effective than GRU-RL (Guo et al., 2018).

- The Estimator - Generator - Evaluator (EGE) model (Wu et al., 2021b), which learns a policy that depends on observations but also on action histories (historical feedback and recommendations), and conditions its actions on the entire history. Therefore, it maximises longer-term rewards. For a more detailed description of the model, please refer to Section 2.2.2.

Note that we retain the original training for these models in the cases where we use a user simulator that considers a single target item (base simulator). For the cases where we use our proposed meta-simulator, we modify the evaluation setting, as described next.

### 5.4.2 Setup: Evaluation Measures

In Section 2.4.1, we gave an overview of commonly used evaluation metrics in IR and recommender systems. In this line, and following existing work in CRS (Guo et al., 2018; Liu et al., 2020; Wu et al., 2020, 2021b, 2023; Zhang and Balog, 2020; Zhou et al., 2020), we use classical IR evaluation measures to evaluate the ability of each CRS system to retrieve relevant items. In particular, we measure the ability of the CRS to show the user's desired target at rank 1 (Success Rate @ 1) at each turn of the conversation; Moreover, as there is a ranking of images

Table 5.4: Performance Results of the three CRS models of the Shoes dataset at various turns after applying our meta-simulator. (w/o) Indicates before and (w/) after introducing alternatives. The numbers in brackets indicate the percentage of improvement compared to traditional non-alternative user simulators.

| CRS Model | NDCG@10 | | | MRR@10 | | | SR@1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | turn | | | | | | | | |
| | 3 | 5 | 10 | 3 | 5 | 10 | 3 | 5 | 10 |
| GRU-SL (w/o) | 0.178 | 0.201 | 0.209 | 0.161 | 0.181 | 0.196 | 0.100 | 0.110 | 0.150 |
| GRU-SL (w/) | 0.205 | 0.252 | 0.237 | 0.346 | 0.437 | 0.495 | 0.257 | 0.352 | 0.436 |
| % Improv. | (14.11) | (22.82) | (12.88) | (72.76) | (82.69) | (86.47) | (87.95) | (104.76) | (97.61) |
| GRU-RL(w/o) | 0.234 | 0.275 | 0.303 | 0.218 | 0.255 | 0.291 | 0.150 | 0.180 | 0.240 |
| GRU-RL (w/) | 0.227 | 0.248 | 0.230 | 0.356 | 0.459 | 0.543 | 0.257 | 0.368 | 0.489 |
| % Improv. | (-3.03) | (-10.32) | (-29.31) | (48.08) | (57.14) | (60.43) | (52.58) | (68.61) | (68.31) |
| EGE (w/o) | 0.197 | 0.242 | 0.277 | 0.171 | 0.216 | 0.263 | 0.080 | 0.140 | 0.210 |
| EGE (w/) | 0.240 | 0.277 | 0.286 | 0.350 | 0.474 | 0.611 | 0.236 | 0.384 | 0.552 |
| % Improv. | (19.68) | (13.48) | (3.19) | (68.71) | (74.78) | (79.63) | (98.73) | (93.13) | (89.76) |

Table 5.5: Performance Results of the three CRS models of the Dresses dataset at various turns after applying our meta-simulator. Notation as per Table 5.5.

| CRS Model | NDCG@10 | | | MRR@10 | | | SR@1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | turn | | | | | | | | |
| | 3 | 5 | 10 | 3 | 5 | 10 | 3 | 5 | 10 |
| GRU-SL (w/o) | 0.071 | 0.078 | 0.072 | 0.058 | 0.069 | 0.068 | 0.071 | 0.078 | 0.072 |
| GRU-SL (w/) | 0.131 | 0.139 | 0.125 | 0.235 | 0.306 | 0.353 | 0.127 | 0.238 | 0.316 |
| % Improv. | (59.64) | (55.44) | (53.87) | (120.24) | (126.33) | (135.12) | (56.81) | (100.68) | (125.57) |
| GRU-RL (w/o) | 0.073 | 0.088 | 0.075 | 0.066 | 0.080 | 0.074 | 0.035 | 0.045 | 0.045 |
| GRU-RL (w/) | 0.110 | 0.121 | 0.099 | 0.209 | 0.257 | 0.269 | 0.127 | 0.177 | 0.216 |
| % Improv. | (40.84) | (31.76) | (27.19) | (103.53) | (105.11) | (113.12) | (113.99) | (119.20) | (131.21) |
| EGE (w/o) | 0.060 | 0.074 | 0.085 | 0.055 | 0.072 | 0.084 | 0.060 | 0.074 | 0.085 |
| EGE (w/) | 0.157 | 0.200 | 0.225 | 0.317 | 0.419 | 0.541 | 0.233 | 0.327 | 0.472 |
| % Improv. | (88.48) | (91.79) | (90.22) | (140.24) | (140.85) | (146.22) | (117.35) | (126.18) | (138.70) |

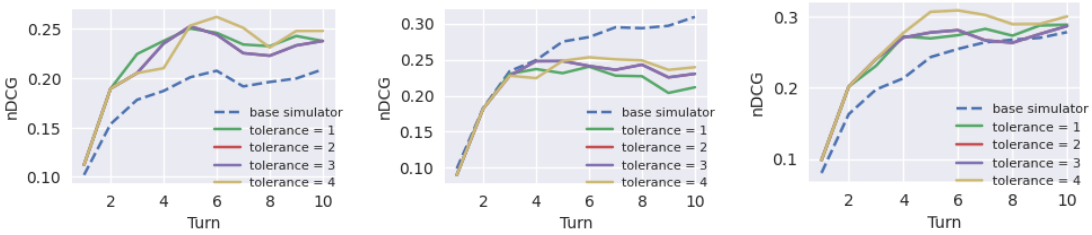
created at each turn, we use nDCG@10 and MRR@10 to evaluate the presence of target items in the ranking. Examining a variety of metrics provides a stronger indication of a generalisable system performance. Following Guo et al. (2018) and Wu et al. (2021b), we terminate conversations at turn 10 (the last evaluation turn); if a target item has been found before turn $k$, then all evaluation measures after turn $k$ are set to 1. Finally, and differing from previous work, we consider the alternatives as relevant items for the purposes of evaluation - in this case, a conversation is successful if any alternative (or the original target) is retrieved (this holds for evaluating the meta-simulator; we still use the classical evaluation setup for the base simulator). In what follows, we answer the above RQs.

### 5.4.3 RQ5.1 - Impact of alternative-based user simulator on the evaluation of existing CRS models

Tables 5.5 and 5.6 show the performance of the three CRS models at turns 3 (early turn), 5 (middle turn) and 10 (end of evaluation turn) of a conversation for the three evaluation metrics after applying our proposed alternative-based simulator on the different fashion categories (coming from the original Shoes and FashionIQ Dresses datasets), respectively. To indicate the difference in performance caused by the updated user simulator and the larger number of relevant items, we include the percentage of improvement compared to the traditional setting with the non-alternative user simulator. For this purpose, we fix the tolerance (user patience) parameter at turn 2, which resembles a real shopping scenario, in the sense that a real user would provide feedback for a couple of turns, and if the assistant's suggestion was not close to their desired item, they would start changing their strategy. Note that we examine the performance at turns 3, 5 and 10 for the following reasons: (a) Turn 3 comes after the tolerance threshold, which means we observe what happens immediately after the user stops being patient and decides to consider other target items. Also, turn 3 is an early turn, and we are interested in how early the new simulator can impact the evaluation of current systems; (b) Turn 5 is a medium-term turn, which means we allow some time for the user to adapt their search behaviour, and also to observe any impact on system performance. (c) Turn 10 in relative captioning settings is the final evaluation turn, i.e., the end of a dialogue, and therefore, we would like to see what the impact of our new simulator is at that stage (bu the end of a conversation).

Overall, we observe improved performance on all three evaluation metrics and both Shoes and Dresses. More specifically, as shown in Table 5.5, for the Shoes dataset, there are considerable improvements on MRR@10 and Success Rate across all three CRS models. For instance, the highest improvement in both MRR@10 and Success Rate is observed for the GRU-SL model, followed by EGE with only small differences. Surprisingly, for NDCG@10, we observe negative values for GRU-RL, which means that CRS performance does not improve or even drops for this model when introducing alternatives. Note that typically comparing performance over different "qrels" is not common practice, since, due to the increased number of relevant items, we should expect higher performances on datasets with larger numbers of relevant items. Still, we opt for this type of analysis, as it shows us how changing the user simulator can increase performance on the same models. Also, in this case, the CRS was performing relatively well for GRU-RL, and therefore, introducing alternatives would not lead to the same improvement compared to a worse performing CRS model. If we compare these numbers with the first row of Table 3, the improvements are proportionally related to the initial ranking of systems in the traditional relative captioning setting; the system that is initially performing worst (GRU-SL) is most increased, and the opposite holds for GRU-RL, which improves the least. Performance improvements are in general greater for MRR@10 and Success Rate than for NDCG@10.

As for the Dresses dataset, we observe improvements across all three evaluation metrics,




(a) GRU-SL (b) GRU-RL (c) EGE

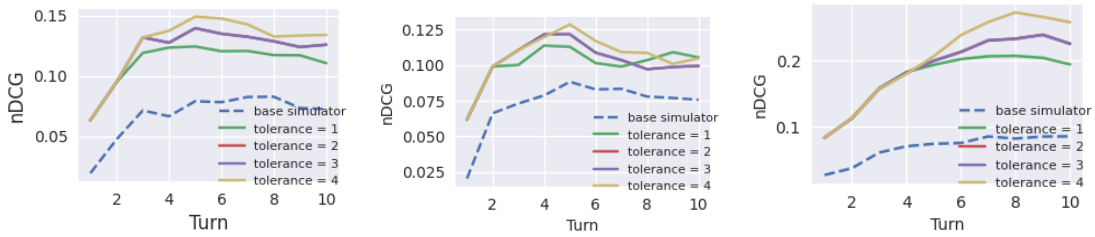
Figure 5.5: nDCG@10 for the various tolerance levels before selecting an alternative for the Shoes dataset.

especially for the initially worst-ranked system (GRU-SL, see also Table 3). Unlike Shoes, for Dresses we observe a positive difference in all cases, and specifically for MRR@10 and Success Rate, effectiveness is doubled when adding the alternative options. Finally, performance is already improved at turns 3 and 5, which means that when a user switches their behaviour at turn 2, they don't have to wait very long to see an alternative product. To answer RQ1, the impact of using an alternative-based simulator is marked positive when evaluating existing CRS models. This suggests that the previous single-target based user simulators were under estimating the effectiveness of the CRS for real users.

### 5.4.4 RQ5.2 - Impact of patience on alternative-based simulator

We approach this RQ with two separate analysis methods, which highlight both the average and the per query performance of each CRS model. More specifically, to answer RQ5.2 (a), we consider the average performance of each CRS model for each tolerance level and also comparing with the base simulator (without alternatives). For this, we turn to the graphical results in Figures 5.5 and 5.6, which show the NDCG@10 average (for all target images) system performance at various tolerance levels for Shoes and Dresses, respectively. The solid lines correspond to the different tolerance levels, while the dashed line denotes the baseline evaluation setting without alternatives for each system. Our observations can be summarised as follows: (i) In general, the earlier a simulator "loses" its patience, the earlier the turn there is a boost in performance. However, when tolerance increases (patience lost at later turns), there is a higher performance improvement (compared to the non-alternative simulator) in the long-term. (ii) Strangely, for Shoes, we observe a decrease in performance for all tolerance levels for GRU-RL (the initially best performing system), but this differences is more prominent in later turns; for earlier turns performance is quite similar with the base simulator. (iii) In general, values between tolerance levels do not differ significantly. To sum up, the impact of patience is more direct in the turns that follow the tolerance turn, but it is not necessarily different across different levels, thus answering RQ5.2 (a).

To answer RQ5.2 (b), we turn our attention on the per target (using the individual target




(a) GRU-SL (b) GRU-RL (c) EGE

Figure 5.6: nDCG@10 for the various tolerance levels before selecting an alternative for the Dresses dataset.

image results) NDCG@10 results for each system, and study the influence of a number of factors on CRS performance using statistical modeling. We treat the per target NDCG@10 at each turn as a repeated measurement dependent variable, and we examine the effect of CRS model and tolerance as independent factor variables. In particular, we test this relationship with a two-way repeated measures ANOVA with CRS model and level of tolerance as within-target image factors and NDCG@10 at each turn as the repeated measures dependent variable. We examine the data for each fashion dataset (Shoes and Dresses) in separate models. For this purpose, we compare between a number of models as follows:

$$
\begin{aligned}
Y_{ijk} &= \mu + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \varepsilon_{ij} (FM1) \\
Y_{ijk} &= \mu + \beta_k + (\alpha\beta)_{jk} + \varepsilon_{ij} (RM1a) \\
Y_{ijk} &= \mu + \alpha_j + (\alpha\beta)_{jk} + \varepsilon_{ij} (RM1b) \\
Y_{ijk} &= \mu + \alpha_j + \beta_k + \varepsilon_{ij} (RM1c)
\end{aligned} \tag{5.3}
$$

Equation (5.3) presents a series of ANOVA models that test for the presence of 3 effects, 2 main effects (CRS model and tolerance) and their interaction on NDCG@10 at ech turn. We start with the full model (FM1), where each $\alpha_j$ denotes the fixed effect of CRS model, the $\beta_k$s denote the fixed effect of tolerance, and $(\alpha\beta)_{jk}$ is the interaction effect of CRS model*tolerance, while $\varepsilon_{ij}$ is the residual variance. $Y_{ijk}$ is the NDCG@10 of target image $i$ of CRS model $j$ at the $k-th$ level of tolerance. To test for each main effect of CRS model and tolerance, we compare with a reduced model where $\alpha_j$ (RM1a) and $\beta_k$ (RM1b) are set to zero, while RM1c checks for the interaction effect where $(\alpha\beta)_{jk}$ is set to zero. For each comparison, the RM corresponds to an $H_{null}$, while the FM to the $H_{alternative}$. We hypothesise that CRS performance does not vary systematically between CRS models and tolerance levels.

Table 5.6 shows the results of the repeated measures ANOVA models for both shoes and dresses. While both CRS model and tolerance level are significant, they account for a very small percentage of the explained variance in system performance, as indicated by the small effect sizes $\eta_p^2$. Indeed, this is expected, since our goal for designing our dataset collection was to have a dataset that when used, would show similar performance across different CRS models, and would not

Table 5.6: Results of Two-way Repeated Measures ANOVA for each fashion category. P-values and effect sizes are shown for each specified model.

| | Shoes | | Dresses | |
|---|---|---|---|---|
| | p-value | $\eta_p^2$ | p-value | $\eta_p^2$ |
| CRS model | $< 10^{-4}$ | 1% | $< 10^{-4}$ | 5.30% |
| tolerance | $< 10^{-4}$ | 0.11% | $< 10^{-4}$ | 0.20% |
| CRS model*tolerance | 0.186 | - | 0.097 | - |

be extremely influenced by the specific time of strategic changes.

We conduct a further analysis to check for the influence of interaction turn and fashion category on system performance. For this, we use a mixed ANOVA model on the combined dataset (merging shoes and dresses). Here, we treat turn and fashion category as fixed factors, and treat the specific target images as a random factor, since they would not be the same if the experiment was repeated with another sample. To test for the main effect of turn and fashion category, we compare the following models:

$$\begin{aligned} Y_{ij} &= \mu + \alpha_j + \pi_i + \varepsilon_{ij} (FM2) \\ Y_{ij} &= \mu + \pi_i + \varepsilon_{ij} (RM2) \end{aligned} \tag{5.4}$$

Equation (5.4) outlines the models for the mixed effects ANOVA, where $\alpha_i$ is the fixed effect of turn (or fashion category), and $\pi_i$ is the random effect of target image. The reduced model (RM2) includes no effect of turn. Note that we assume that $\sigma^2_{\alpha\pi} = 0$ parameter is not included in the model because it is impossible for these data to distinguish the interaction from the error. This is because there is only one measurement per combination of target image and the fixed factor condition. We fit separate models for turn and fashion category. The RM corresponds to an $H_{null}$ where $\alpha_j = 0$, while the FM to the $H_{alternative}$ where at least one $\alpha_j \neq 0$.We hypothesise that CRS performance does not differ to a great extent between fashion category, while we assume that there is a slight effect of turn. The results are shown in Table 5.7. Indeed, turn and fashion

Table 5.7: Results of two-way mixed-model ANOVA for the target images of both fashion categories. (*) indicates that for both examined models, a significant effect of the random factor target image was found.

| | p-value | $\hat{\omega}^2$ |
|---|---|---|
| turn (*) | $< 10^{-4}$ | 0.80% |
| fashion category (*) | $< 10^{-4}$ | 0.64% |

category account for a small percentage of variation, which implies that a noticeable amount of variance is due to the random variation of the random factor target image. This indicates that our collected dataset is indeed diverse and representative of a population of images. To answer the second part of RQ2, the system performance change due to the new simulator differs only slightly for different CRS models. These results are in line with previous results on query performance (Faggioli et al., 2023b), where the topic in a TREC collection was found to explain

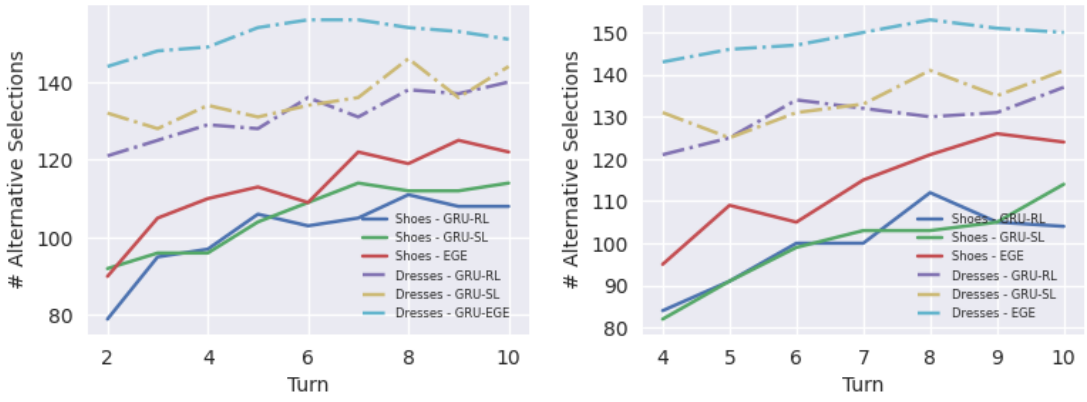
a much higher proportion of the total variance compared to other experimental factors.

Table 5.8: Resulting ranking (based on NDCG@10, as shown in the numbers within brackets) of the 3 CRS models at turn 10 (end of dialogue evaluation setting) using the non-alternative simulator and the various tolerance levels of the alternative-based simulator.

| Simulator type | Shoes | Dresses |
|---|---|---|
| no alternatives | **GRU-RL**(0.309) > **EGE** (0.277) > **GRU-SL** (0.209) | **EGE** (0.085) > **GRU-RL** (0.075) > **GRU-SL** (0.072) |
| tolerance 1 | **EGE** (0.288) > **GRU-SL** (0.237) > **GRU-RL** (0.211) | **EGE** (0.194) > **GRU-SL** (0.110) > **GRU-RL** (0.105) |
| tolerance 2 | **EGE** (0.286) > **GRU-SL** (0.237) > **GRU-RL** (0.230) | **EGE** (0.225) > **GRU-SL** (0.125) > **GRU-RL** (0.099) |
| tolerance 3 | **EGE** (0.286) > **GRU-SL** (0.237) > **GRU-RL** (0.230) | **EGE** (0.225) > **GRU-SL** (0.125) > **GRU-RL** (0.099) |
| tolerance 4 | **EGE** (0.300) > **GRU-SL** (0.487) > **GRU-RL** (0.239) | **EGE** (0.258) > **GRU-SL** (0.133) > **GRU-RL** (0.104) |

### 5.4.5 RQ5.3 - Role of patience in the effectiveness of CRS models

We test this RQ by examining how the CRS models perform in comparison to one another before and after introducing our alternative-based meta-simulator. Table 5.8 shows the NDCG@10 ordering of the three CRS models before (first row) and after (remaining rows, each at another tolerance level) introducing the meta-simulator. For both datasets, the relative ordering changes at tolerance 1 compared with the non-alternative setting, but then the ordering remains stable with the varying tolerance levels, with one main difference; for Shoes, there is a swap between the first and the second systems (EGE is ranked first when alternatives are introduced, and GRU-RL moves to the second place), while for Dresses, patience reorders the second and third systems (GRU-SL is improved compared to GRU-RL). To answer RQ5.3, introducing patience partially changes conclusions about the effectiveness of models, but this change is not further influenced by the increasing level of patience and is not replicated across fashion categories.

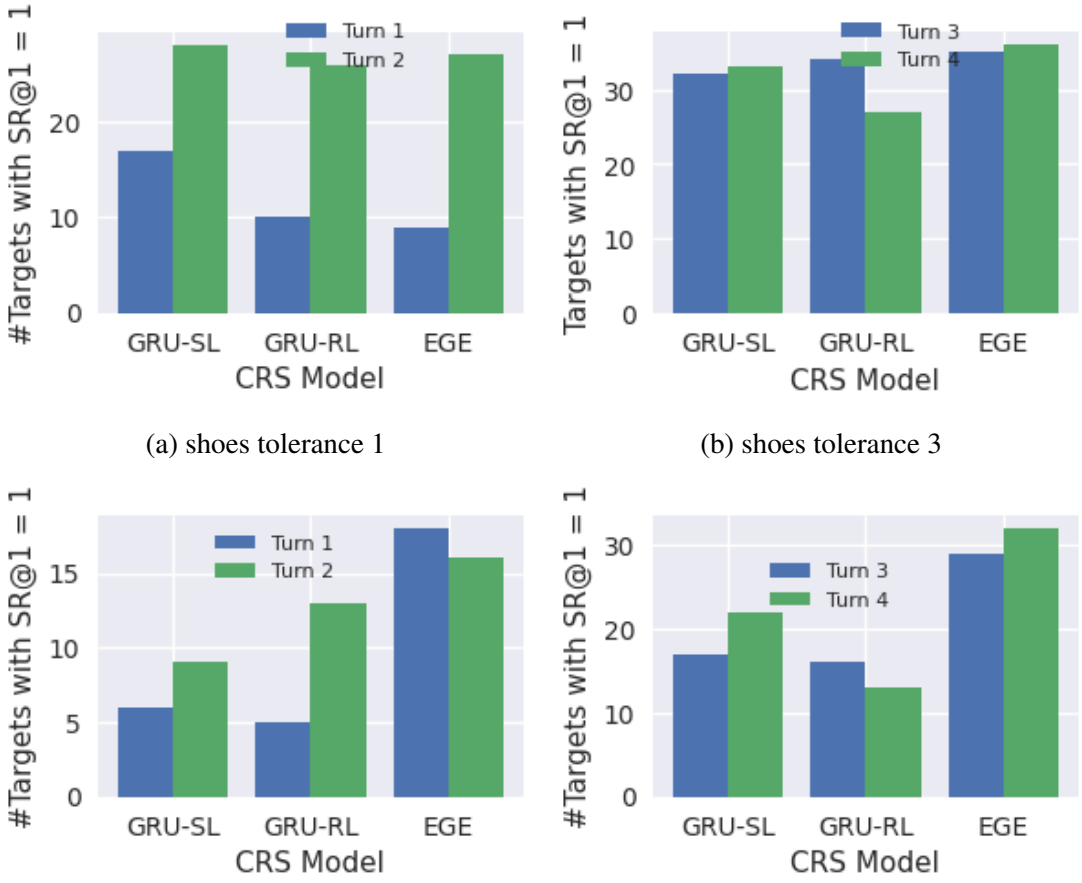



(a) tolerance = 1 (b) tolerance = 3

Figure 5.7: Number of target images for which the simulator selects an alternative over the target for the three CRS models for tolerance 1 and 3.

### 5.4.6 RQ5.4 - Frequency of selecting an alternative

In Section 5.2, we mentioned that the we instrument our proposed meta-simulator algorithm to provide data about how often alternatives are chosen. Therefore, to address this RQ, we use this information. Specifically, Figure 5.7 shows the number of times, out of the 200 sampled target items, the simulated user opts for an alternative over the initial target, for an earlier (turn 1) and a later (turn 3) tolerance level. The solid lines represent the selection of Shoes CRS models, while the dashed lines denote the performance of Dresses. While the pattern is similar between the two fashion categories, simulators trained on Dresses tend to select on average more alternative items than simulated users trained on Shoes. This might be explained by the fact that in the initial stage before introducing alternatives, all three systems performed worse when trained on Dresses. Indeed, they might benefit more when there is an option to make an alternative selection. In addition, there is an increased tendency to select alternative items as turns increase and the user patience decreases, which seems reasonable. For this reason, a further check is necessary to check the immediate change in system performance with respect to the tolerance level.




(a) shoes tolerance 1

(b) shoes tolerance 3

(c) dresses tolerance 1

(d) dresses tolerance 3

Figure 5.8: Number of target image items that achieve an SR@1=1 for for an early tolerance level (patience = 1) and later tolerance level for each of the alternative fashion categories.

To compare the system performance before and after the alternative selection more precisely, we examine system performance in terms of how many times a system has identified the target item. Specifically, Figure 5.8 shows for tolerance levels 1 and 3, how many target items were found (had SR = 1) at the turn before and the turn after the strategy change (user switching to an alternative target). In general, we see that the improvement is more rapid when the user patience is lost at an earlier stage (tolerance = 1), for example changes between turns 1 and 2 compared to changes between turns 3 and 4 (tolerance = 3). Losing patience after turn 3 still leads to small improvements in some cases, but overall this is when system behaviour is becoming more stable.

Returning to Figure 5.7, we see that the selection of alternatives is more frequent for the EGE model for both fashion categories compared to the two GRU-based models. This result is a bit surprising, but can be linked to the following: System performance over different relevance assessments is usually not compared, as the increased number of relevant items is expected to lead to higher performances on datasets with larger numbers of relevant items. Still, our aim to some extent is to study the simulator behaviour under different contexts. In this regard, the different simulator settings are expected to lead the meta user simulator to change feedback when it changes to an alternative, and this may confuse a system that might be working quite well previously - that might explain this behaviour. In general, observing the high frequency of alternatives indicates that the lower performance of CRS models cannot be solely attributed to their retrieval ability, but also on the lack of sufficient target items. In short, users often tend to pick an alternative when they have the option to do so, thus answering RQ5.4.

## 5.5 Concluding Remarks

In this chapter, we have addressed the issue of obtaining relevance judgments in Conversational Recommendation Systems to achieve a more realistic recommendation setting and more accurately predicting user preferences, thus addressing the third hypothesis of our thesis statement as outlined in Section 1.2. In particular, we have introduced a new relevance annotation approach which is based on directly asking real users about the relevance of items with respect to their similarity with a given target item. For this purpose, we have conducted a user study that used crowd-sourcing to expand the existing well-used Shoes and FashionIQ Dresses datasets into a unified dataset with alternatives. In this way, we managed to extend the target space of a simulated user in the Conversational Image Recommendation setting by including the identified alternatives into the input datasets used to train the user simulator. In this regard, we have shown how a sufficient amount of target items can be identified based on precise estimations that include pooling from diverse systems and various levels of difficulty, moving away from a perspective that selects a dataset size with a fixed require items. As a result, we ended up with an equivalent to TREC collections in information retrieval. Consequently, we created a more realistic novel dialog-based recommendation scenario, where a user is assumed to have a more widely-defined information need, is flexible to adjust their strategy during a conversation according to what they see and have the opportunity to change their mind. This was done by

introducing a meta-user simulator that uses the alternative relevant items for training and evaluation of CRSs. Our simulator informs the existing base (non-alternative) user simulator with knowledge of the alternative options to given target items, and therefore, allows the (simulated) users to change their mind during the CRS interaction. Therefore, in addition to proposing a new recommendation scenario and user perspective, we proposed an evaluation methodology that adapts the user simulator based on the newly collected annotation data.

Overall, we found that a system's performance is increased when changing the way a user simulator requests for a given item. For the same CRS models, using these extended datasets and the corresponding meta-simulator for evaluation, we showed that previous (single-target) evaluations may underestimate the effectiveness of CRS systems on these datasets. Indeed, if they accept other alternative items, and are willing to switch strategy, then the system may satisfy them sooner. In particular, we obtained improved performance on the same CRS models up to 140% compared to the previous setup (Tables 5.4 and 5.5). At the same time, our experiments showed that the patience of a simulated user (indicated by the turn at which they choose to change their feedback) has only a small impact (Table 5.6), but is noticeably different from the base simulator (Figures 5.5 and 5.6), while a similar performance is observed for across CRS models. In contrast, some degree of variation in performance is due to the random variation in target images (Table 5.7), proving that our collected dataset is indeed diverse. To summarise, users indeed tend to prefer alternatives when they have the option (Figure 5.7, and the earlier they do this the more immediate the increase in performance at the next tun (Figure 5.8). Overall, in this chapter, we have validated the third claim of the thesis statement, according to which *by obtaining user opinions about the relevance of items, we improve the completeness of the evaluation mechanism by identifying alternatives recommendations for existing target items, which could be used to both inform the user simulator and therefore improve the overall evaluation of CRS systems.*

As for our collected dataset with alternatives, its use is not restricted to a multi-turn recommendation setting. For example, it could also be used for single-turn image retrieval, which is a concept more similar to traditional TREC collections. Additionally, it could be used for different recommendation settings by modifying our meta-simulator accordingly. Still, one limitation of our dataset is that it does not provide annotations at scale, while it could also incorporate more fashion categories. This is something that we plan to do in the near future. As a further limitation, we note that while our meta-simulator supports opting for alternatives, and therefore a wider information need, it does not support cases where a user's target is not contained in the available database. Therefore, in the following chapter, we extend our set of recommendation scenarios with a third scenario that includes predictions for cases with missing items together with predicting recommendation success with alternatives. Finally, in the following chapter (Chapter 6), we use our collected alternative datasets to evaluate our CPP framework under a new scenario that includes alternatives in addition to the original target.

# Chapter 6

# Predicting Conversation Performance across Recommendations Scenarios

The results of Chapter 4 indicated that predicting the success of a conversation within the context of interacting with a conversational agent is possible, since it can predict both the degree of success of a conversation and the stage when the item is more likely to be returned. In particular, we introduced a novel framework of Conversational Performance Prediction (CPP), which transforms the task of Query Performance Prediction (QPP) (Carmel and Yom-Tov, 2010; Cronen-Townsend et al., 2002) from the query level (search task) to the conversation level (conversational recommendation task). In this regard, we considered the multi-turn aspect of the task and showed how we can differentiate between predicting current user satisfaction or overall satisfaction of a conversation. Specifically, we predicted the CRS rankings consisting of image items by using traditional QPP evaluation measures, and followed by proposing a classification-based evaluation approach, where we predicted the success label of a conversation on the test set. Indeed, while correlation-based evaluation does not properly capture the underlying relationship between per-query predictor and metric values (therefore providing lower correlations than what those observed in QPP) (Section 4.2), treating CPP as a binary success prediction task, and specifically learning the embedded representations of the image items contained in the train set, provided a promising solution for detecting conversational failures and to what extent using multi-turn features adds value to a single-turn predictor (Section 4.4). In this way, we confirmed the second proposition of the thesis statement (Section 1.2) Still, although we used a variety of predictors and evaluation settings, our CPP task (as defined in Chapter 4) only involves the base settings of the Conversational Image Recommendation task, as these were defined in Section 2.2.2. For example, it assumes a clearly defined target item which is always available in the item database. However, in practice, the situation is more complicated, and the definition of recommendation success can vary accordingly.

More relevant to this, in Chapter 5, we highlighted the importance of making predictions in a realistic setting that sufficiently mimics a real life user shopping scenario. Indeed, within

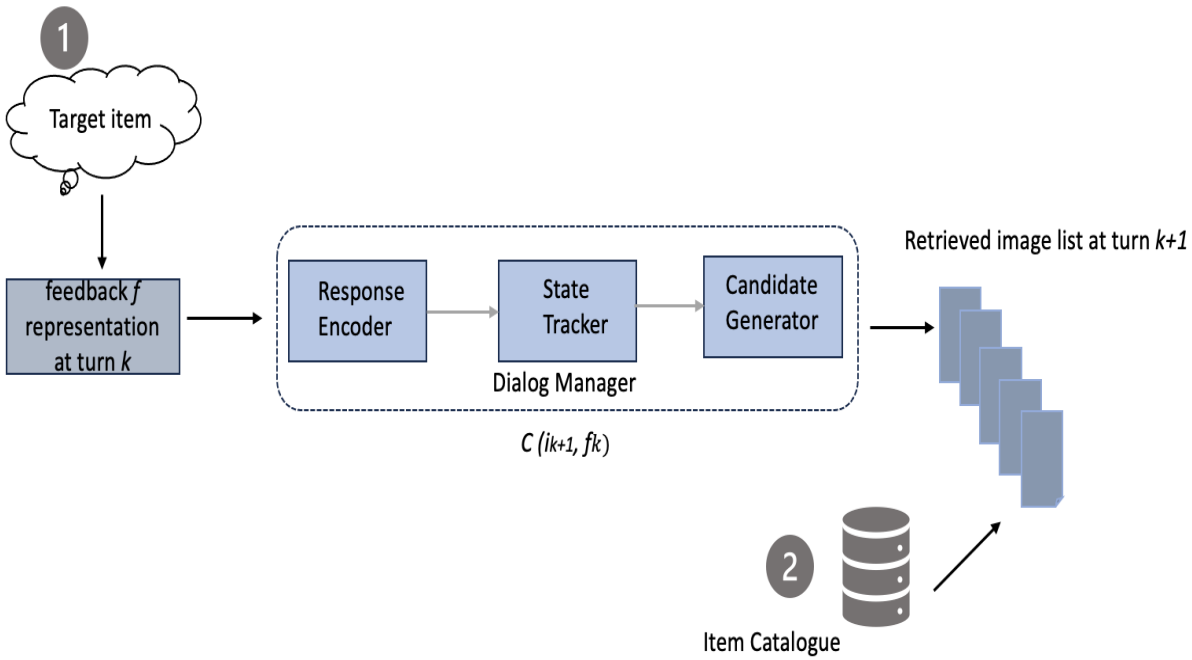



Figure 6.1: Description of the Conversational Image Recommendation steps, expressed in terms of a ranking task as introduced in Section 2.1.2. The different parts of the Dialog Manager receive user feedback $f$ at turn $k$, which is influenced by the user's target item in order to produce a recommendation list of items at turn $k+1$. Two issues arise in this process not currently taken into account by CRSs: (i) The target item might not be available in the Item Catalogue (Step 2), which we call Scenario 2, and (ii) The target item is not always clearly defined (Step 1), which we call Scenario 3.

the context of reinforcement learning-based simulated conversations (Guo et al., 2018) (as mentioned in Section 2.2.2), a user provides a relative feedback utterance or critique that describes the visual differences between the presented item and their desired item. In Chapter 5, we redefined the meaning of this *target* item that the user has "in mind", not at each turn but overall in a conversation. More specifically, we detected a limitation in all the different variants of the task (Guo et al., 2018; Wu et al., 2020, 2021b; Yu and Grauman, 2017), namely that the target item suggests a known-item type of task (Broder, 2002), where the target item is assumed to be defined and exist in the item catalogue of the system. Still, in a real shopping scenario, a user might have multiple equally desired options that can be considered as alternative target items. For this purpose, we collected real user opinions for the relevance of a number of alternative image items for different fashion categories and retrained three CRS models. In this way, we showed that by extending the target space and making user needs more flexible and realistic, we can increase CRS performance, which was previously considered ineffective, thereby confirming the third hypothesis of the thesis statement (Section 1.2). At the same time, while we managed to increase the performance of three CRS models when introducing our concept of alternatives, predictions about the success of a conversation are not yet explored using the CPP task.

In other words, in order to predict conversational performance for a facet of user needs,

we need to consider the particular context and variations of the Conversational Image Recommendation task, and in particular the settings of the different parts of the task (as defined in Section 2.2.2). Examples of these variations in the task setting can be seen in Figure 6.1. This image describes two variations factors that can lead to modifications in the CRS functionality; Step 1 denotes whether the user's target item is clearly defined, while Step 2 refers to whether the same target item is available in the corresponding *item catalogue*. The traditional task setting (Section 2.2.2) corresponds to a clearly defined target that is available in the catalogue. We call this case *Base Scenario* or *Scenario 1*. Sometimes, a user can have a unique target item, but this item might be unavailable. In this case, the item will not be returned, simply because it does not exist in the item catalogue; we call this *Missing Target Scenario* or *Scenario 2*. Finally, while the catalogue might be rich and multiple items are available, a user might have only a vague need or a more flexible definition of a target. In this case, multiple items can satisfy their need, which could be seen as alternatives. In this case, we refer to this as the *Alternative Scenario* or *Scenario 3*. Scenario 3 results from what we introduced as alternative options in Chapter 5.

Therefore, in this chapter, we account for these variations by examining different recommendation scenarios as part of the CPP task. In particular, we experimentally test our fourth hypothesis of the thesis statement (Section 1.2), which states the following: *Finally, using these alternatives datasets, and by predicting conversational performance under different Recommendation Scenarios, we detect different types of conversational failure, such as when a user cannot find an item, versus when the system's catalogue does not contain the relevant item.* In this way, we create different variations of the CPP task by adapting it to more realistic real user settings. Overall, we address **Limitation 1a)** (*A system trying to find a single item that is already known by the user contradicts the recommendation intuition*), **Limitation 2a)** (*In some cases, an item requested by the user is not contained in the item database or catalogue. If a system does not account for this, system failure will be assigned as a reason to a problem that is best described as* catalogue failure) and **Limitation 2b)** (*The given context of Conversational Image Recommendation does not allow more generalised user satisfaction where the user has a vague information need that could be satisfied when the system returns the user's target item or another item similar to the original target based on a certain criterion.*), since we propose two novel recommendation scenarios and account for various types of recommendation failure. In particular, the contributions of this chapter can be described as follows:

- We introduce the concept of *Recommendation Scenarios* in the task of Conversational Image Recommendation by considering two factors of variation (the definition and the availability of the target item). Therefore, we define the *Missing Target* and the *Alternative* scenarios.

- We introduce the different types of recommendation failures resulting from each scenario, and consequently, we differentiate between the system not being able to retrieve the correct item and the item not being available.

- We experimentally test and extend our CPP framework on the two new recommendation scenarios under different CRS models and datasets and using a variety of CPP predictors. We compare the results of each new scenario with our base CPP evaluation setting (Scenario 1).

- Essentially, this Chapter links the Alternatives options (obtained in Chapter 5) to CPP (proposed in Chapter 4). In this way, on top of proposing two novel CPP recommendation scenarios, we investigate whether the conclusions about CPP change when using the alternative-based evaluation.

Consequently, the main findings of this chapter can be summarised as:

- For both single-turn and multi-turn prediction, moving to Scenario 2 (Removed Target) increases CPP performance when predicting the top item, while it decreases its performance when predicting deeper rank cutoffs (items found by rank 100). This implies that after removing targets, predicting the effectiveness by using items found at rank 1 is an easier task than the same prediction in Scenario 1 (target exists), while predicting the effectiveness by using items found by rank 100 is more difficult than the same prediction in Scenario 1.

- For both single-turn and multi-turn prediction, moving to Scenario 3 (Alternatives) decreases CPP performance when predicting the top item, while it increases its performance when predicting deeper rank cutoffs (items found by rank 100). This implies that after introducing alternatives, predicting the effectiveness by using items found at rank 1 is more difficult task than the same prediction in Scenario 1 (single target), while predicting the effectiveness by using items found by rank 100 is easier than the same prediction in Scenario 1.

- In some cases, our proposed embedding-based predictor, which is based on selecting the important embedding features of a multi-turn conversation, namely the L1-based variant of RV (introduced in Section 4.4.1, and in other cases, our proposed AE-based predictor (introduced in Section 4.4.1) lead to change of the ordering of CPP predictor performance from Scenario 1 (target exists) to Scenario 2 (missing target).

- Our proposed AE-based predictor helps in generalising CPP behaviour across Scenarios 1 (single target exists) and 3 (alternatives equally relevant to the target), which shows that it is a promising predictor of conversational performance.

- The differences in CPP results indicate the difficulty of predicting conversational performance under different scenarios, and the pattern is different according to the depth of the predicted ranking.

The rest of the chapter is organised as follows: We present our new recommendation Scenarios and corresponding CPP evaluation for each of them in Section 6.1, we continue by outlining our Experimental Setup in Section 6.2, and present the results for Scenario 2 in Section 6.3 and for Scenario 3 in Section 6.4. Finally, we end with concluding remarks in Section 6.5.

## 6.1 CPP Scenarios

In this Section, we introduce and detail the two novel recommendation scenarios (Scenario 2: Missing Target and Scenario 3: Alternatives) for our CPP Framework and explain how they differ from the base scenario (Scenario 1: Target Exists). First, we describe the settings of each recommendation scenario (Section 6.1.1) and later in Section 6.1.2, we explain how we transform the proposed CPP predictors of Chapter 4 to match the requirements of each scenario.

### 6.1.1 Recommendation Scenarios Definition

As we introduced in Figure 6.1, the traditional setting of the Conversational Image Recommendation task (Section 2.2.2) can vary. This variation can be induced by two different variation factors; whether the target is clearly defined and whether it is available. First, we examine the case of a clearly defined user target image item. In this case, the evaluation settings are similar to the traditional scenario (where a single target is defined by the user and is assumed to exist in the catalogue, we call this Scenario 1), in the sense that only one item is considered as relevant. Still, if the target item is not returned, there are two likely scenarios: First, the system is unable to return the target because the CRS model is ineffective, and second, the target simply does not exist. In current CRS models, only the first scenario is considered, and therefore, any failure to return the target to the user is considered as a model defect; we call this *system failure*. While this is true in some cases, a lot of times in a real user shopping scenario, the target item is not available. If the user is not aware of it, they will keep providing feedback without success. This type of failure is different from system failure, as it results from the product unavailability rather than an inability of the system to retrieve the target item; we call this *catalogue failure* and the corresponding recommendation setting as the *Missing Target Scenario* or *Scenario 2*. Consequently, to compare CPP performance between Scenarios 1 and 2, we need to differentiate between two types of failures: system failure and catalogue failure. More formally, the CPP task under Scenario 2 can be described as a function of the form

$$CPP(F^{rem}, S^{rem}) \rightarrow \mathbb{R} \tag{6.1}$$

where $F^{rem}$ is a sequence each containing $f$ feedback critiques over 1 or more turns when the target item is missing, and $S^{rem}$ is a sequence of results (recommendation) lists returned when the target item is not available, over 1 or more turns.

Next, we consider the case of a user target image item that is not clearly defined, for example when the user has a vague user need that could be satisfied by more than one item. In this case, the evaluation settings are not in line with the traditional scenario (Scenario 1), since more than one items are considered as relevant. In Chapter 5, we accounted for this limitation in current CRSs by creating a new CRS evaluation setting, where multiple identified alternatives are considered equally relevant to a given target item and are included in the new resulting target space. In this case, if the target item is not returned, it means that the system was not able to return either the original target or any of the alternatives. In current CRS evaluation, any failure to return any item of a target space to the user was not previously considered, and this was part of our motivation for Chapter 5. We call this recommendation scenario *Alternatives Scenario* or *Scenario 3*. Therefore, to compare CPP performance between Scenarios 1 and 3, we need to differentiate between system failure and *alternative failure*. More formally, the CPP task under Scenario 3 can be described as a function of the form

$$CPP(F^{alt}, S^{alt}) \rightarrow \mathbb{R} \tag{6.2}$$

where $F^{alt}$ is a sequence each containing $f$ feedback critiques over 1 or more turns when there are multiple alternative target items, and $S^{alt}$ is a sequence of results (recommendation) lists returned when the user is looking for an item with a more flexible user need, over 1 or more turns.

### 6.1.2 CPP Predictors Definitions Per Scenario

We now explain how we adapt the CPP definitions from Chapter 4 to each of the recommendation scenarios. Starting with Scenario 2 and following the notation introduced in Equation (6.1): Similarly to Equation (4.1) in Chapter 4, Equation (6.1) can be instantiated for single-turns, or multiple turns. For example, for the single-turn setting, the CPP task at a given turn $k$, i.e.:

$$CPP_{\text{single}}([f_k^{rem}], [s_k^{rem}]). \tag{6.3}$$

where $[f_k^{rem}]$ the information contained in the feedback at turn $k$ that describes an item that does not exist, and $[s_k^{rem}]$ includes both score-based and representation-based features contained in the retrieved list of items of turn $k$ resulting from feedback describing an item that does not exist. Similarly, for the consecutive turn setting, we define the task as:

$$CPP_{\text{consecutive}}([f_k^{rem}, f_{k+1}^{rem}], [s_k^{rem}, s_{k+1}^{rem}]). \tag{6.4}$$

where the notation for feedback and retrieved contents is similar to Equation (6.3), but for two consecutive turns. Finally, and more relevant to our setting, we define the CPP - Scenario 2 under the classification setting. Specifically, following Equation (4.6), we define a classifier

which aims to predict if conversation $C^{rem}$ will be successful or not as follows:

$$cls(X_{C^{rem},k}) \rightarrow \{0,1\} \tag{6.5}$$

where $X_{C^{rem},k}$ is the feature representation for a given conversation at a given turn k in a missing target setting. Consequently, each of the supervised CPP predictors defined in Chapter 4 can be used in the same way as defined in Equations (4.7) to (4.13).

Next, we define the task for Scenario 3 following the notation introduced in Equation (6.2). For the single-turn setting, the CPP task at a given turn $k$, i.e.:

$$CPP_{\text{single}}([f_k^{alt}],[s_k^{alt}]). \tag{6.6}$$

where $[f_k^{alt}]$ the information contained in the feedback at turn $k$ that describes an item that is either the original target or the alternative item that is most similar to the current candidate item, and $[s_k^{alt}]$ includes both score-based and representation-based features contained in the retrieved list of items of turn $k$ resulting from feedback describing any of the original target or an alternative. Similarly, for the consecutive turn setting, we define the task as:

$$CPP_{\text{consecutive}}([f_k^{alt},f_{k+1}^{alt}],[s_k^{alt},s_{k+1}^{alt}]). \tag{6.7}$$

where the notation for feedback and retrieved contents is similar to Equation (6.6), but for two consecutive turns. Finally, the supervised CPP - Scenario 3 task for a conversation $C^{alt}$ is defined as:

$$cls(X_{C^{alt},k}) \rightarrow \{0,1\} \tag{6.8}$$

where $X_{C^{alt},k}$ is the feature representation for a given conversation at a given turn k in an alternative-based setting. Again, each supervised CPP predictor defined in Chapter 4 is the same way as defined in Equations (4.7) to (4.13). Specifically, these predictors consider the features of a retrieved recommendation list at a given turn $k$ to predict turn $k+1$ (single-turn predictors), or they considered the contents of lists up to turn $k$ to predict turn $k+1$ (multi-turn predictors). As explained in Section 4.4.1, these predictors examine score-based features (combining the mean, maximum score, and standard deviation of a results list), or the embedding based features that adapted the Reciprocal Volume (RV) initially proposed for Conversational Search (Faggioli et al., 2023a), and finally our novel AE-based predictor that gradually learns a compressed version of the embedded representations. After testing their CPP performance in the base scenario, in this chapter, we extend the CPP predictions to our two novel recommendation scenarios by comparing CPP perofrmance with Scenario 1 on these predictors. Next, we describe how we conduct our experiments.

## 6.2 Experimental Setup

For our experiments, we first compare CPP performance of Scenario 2 with Scenario 1 (base scenario), and therefore, test the case of catalogue failure against system failure (which one is easier to detect with CPP). For this purpose, we address the following research questions:

**RQ6.1** How do single-turn CPP predictors compare between Scenarios 1 and 2 (after removing a portion of the target items) for (a) predicting the top-ranked item and (b) predicting a full ranking of items?

**RQ6.2** How do multi-turn CPP predictors compare between Scenarios 1 and 2 (after removing a portion of the target items) for (a) predicting the top-ranked item and (b) predicting a full ranking of items?

More specifically, we define a successful conversation as one where the target item is retrieved by a given rank (1 or 100) at a given turn (these are easy items), and a system failure otherwise (difficult items). To induce Scenario 2, we select 30% of easy items and prevent their retrieval to emulate catalogue failures. The reasoning behind our choices is the following: For *Conversation Failure Ground Truths*, we consider three cases for any conversation: (i) the conversation is successful, as the target item is retrieved; (ii) the conversation fails, because the system is unable to retrieve the target item based on the user's feedback before a fixed number of turns expires (i.e., a system failure); and (iii) the conversation fails because the system's does not contain the target item (i.e., a catalog failure). In practical terms, for difficult items, which the system struggles to retrieve, there is no difference between system and catalog failures. Therefore, to emulate catalog failures, we sample easy items (which the system can normally retrieve successfully), and prevent them from being retrieved, to emulate catalog failures. When doing so, we recalculate the CPP features.

Next, we will compare CPP performance of Scenario 3 with Scenario 1 (base scenario), and therefore, test the case of alternative failure against system failure.. For this purpose, we address the following research questions:

**RQ6.3** How do single-turn CPP predictors compare between Scenarios 1 and 3 (after introducing alternatives) for (a) predicting the top-ranked item and (b) predicting a full ranking of items?

**RQ6.4** How do multi-turn CPP predictors compare between Scenarios 1 and 3 (after introducing alternatives) for (a) predicting the top-ranked item and (b) predicting a full ranking of items?

To induce Scenario 3, we use the evaluation setting of Chapter 5 that we termed as "after alternatives" or "(w/)". For each CRS model, we use the evaluation setting and train the models as defined in Section 5.4. For all three scenarios, we study single-turn predictors with Equations (4.7), (4.9), and (4.12), and the multi-turn predictors using Equations (4.8), (4.11), and (4.13). For Scenario 3, we use the tolerance level after turn 2, similarly to Section 5.4. We examine the CPP predictors by using two rank cutoffs in the ground truth turn: rank = 1 and

rank = 100. Similarly to Chapter 4, we compare CPP results for the GRU (Guo et al., 2018; Wu et al., 2021a) and EGE (Wu et al., 2021b) CRS models, and use the Shoes (Berg et al., 2010; Guo et al., 2018) and FashionIQ Dresses (Wu et al., 2021a) datasets. This time, we use two baseline classifiers: First, we deploy one that always predicts the majority class in the training data (denoted *Most Frequent*), as well as a random classifier that predicts classes based on their training likelihood (denoted *Stratified*). For all three scenarios, we use the 200 sampled target items from each dataset used in Chapters 4 and 5. We report Accuracy as measure of classification performance, thus inducing a setting with a smaller number of per query-based results, similarly to the traditional QPP evaluation setting. We compare the CPP results of each of our novel recommendation scenarios with Scenario 1 in the following Sections.

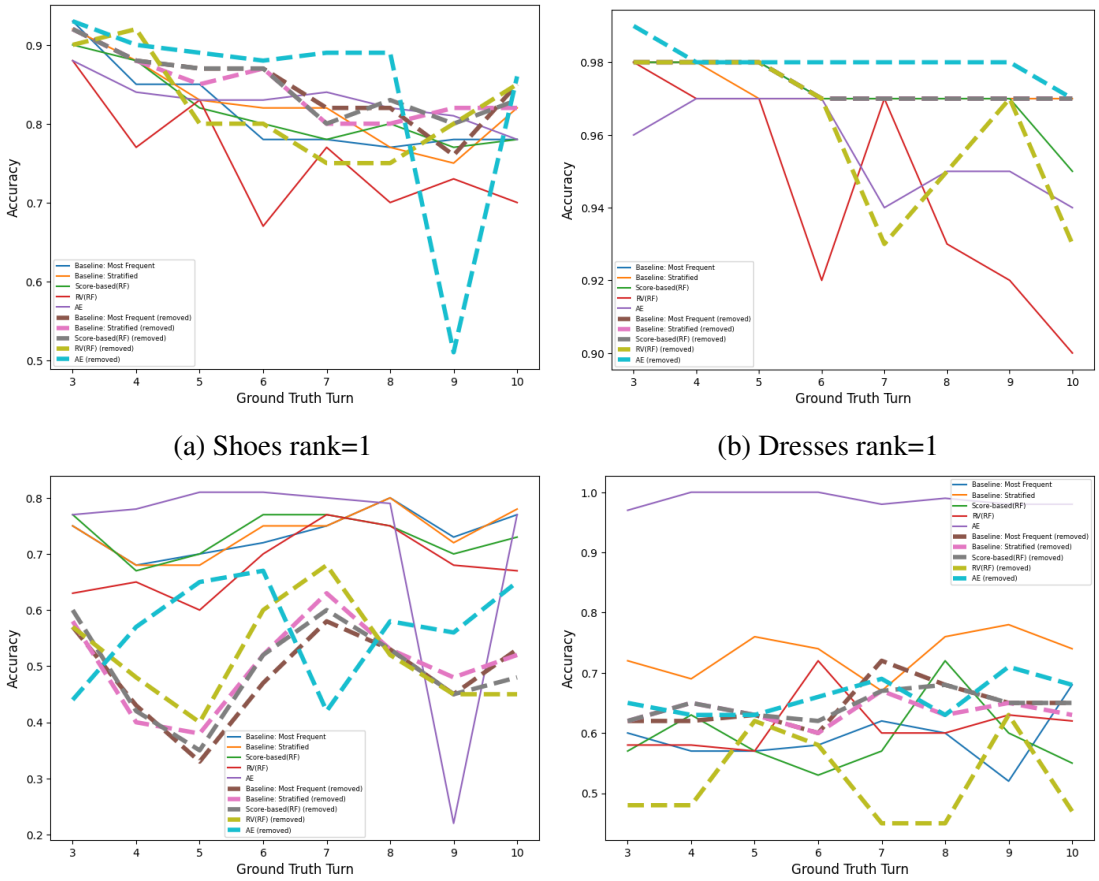
## 6.3 Results Missing Target (Scenario 2) vs Existing Target (Scenario 1)

In this Section, we present the results for the Missing Target Scenario and compare them with the results of the base scenario or Scenario 1 with a target that exists. First, we test the single-turn results in Section 6.3.1, and then we continue with the multi-turn results in Section 6.3.2.

### 6.3.1 Single-turn CPP Results (Missing Target vs Base Scenario)

We examine the single-turn CPP results in Figures 6.2 and 6.3, which display the CPP classification accuracy for the GRU and the EGE model, respectively. In each plot, the x axis shows the ground truth turn of a conversation used for predictions, and the y axis is the predictive accuracy on the test set; each curve corresponds to a separate CPP predictor. The solid lines represent CPP predictors in Scenario 1 (base scenario), while the dashed lined correspond to the same predictors after removing selected target items (Scenario 2) (see Section 6.2 for how we selected targets to remove). The Shoes results are shown on the left and the Dresses results on the right. In each figure, the top two plots show the results for predicting the success of a conversation at rank = 1, and the two bottom plots show predictions at rank = 100.

We start by describing the results for predicting the top-ranked item, for both the GRU (plots 6.2 (a) and (b)) and the EGE model (Figures 6.3 (a) and (b)). In all cases, we observe the following: (i) For Scenario 1 (Existing Target), there is a downward trend for all predictors as turns increase, (ii) The results of Scenario 2 (Missing Target) cannot easily be distinguished from the ones of Scenario 1 (as we see the dashed lines intersecting with the solid lines), but this trend is more prominent for Shoes; for Dresses, we see some relative improvement in accuracy in the removed target Scenario for EGE, (iii) In general, the ordering of predictors in Scenario 1 in each individual plot follows the ordering of Scenario 1. In other words, the predictors follow a similar pattern across scenarios. In addition, for both CRS models, we note that the




(a) Shoes rank=1

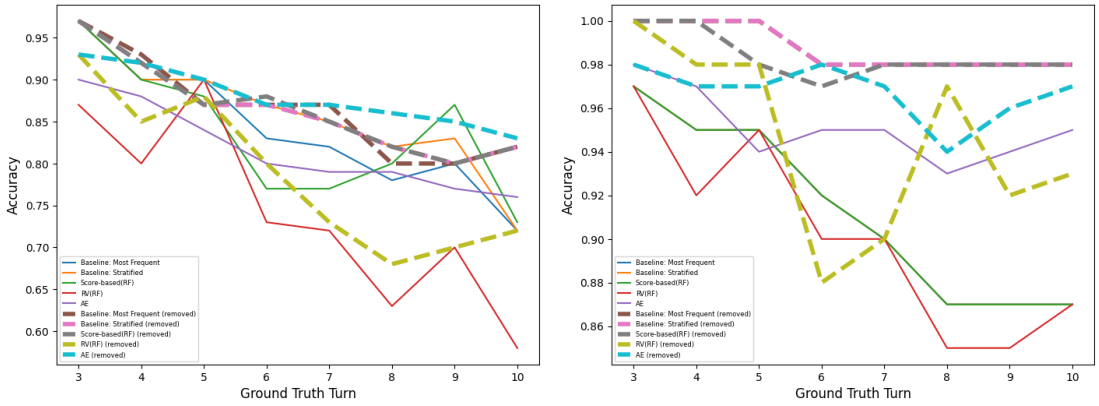
(b) Dresses rank=1

(c) Shoes rank=100

(d) Dresses rank=100

Figure 6.2: CPP Single-turn Results for Scenario 2 for the GRU model for a target item found at rank 1 (top figures) and rank 100 (bottom figures) for each dataset.

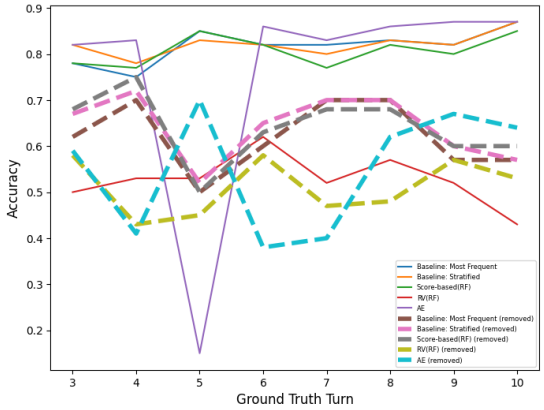
Reciprocal Volume or RV-based predictor's performance is lower than the rest for both scenarios. On the other hand, the performance of AE varies with datasets: For Shoes, it is slightly worse or comparable with score-based and the two baselines, while for Dresses, it is equal to them but drops after turn 6 (GRU) or increases after turn 5 (EGE). We also note that when removing a portion of target images, AE performs better than the other predictors in all cases except for EGE Dresses, where it is comparable to all others but RV. Overall, we observe that it is not easy to distinguish the performance between the two scenarios, since the differences in accuracy levels between the scenarios for each predictor are very small; still, the general trend shows a slight increase compared to the base scenario. We observe a more marked increase for Dresses than Shoes: note that when removing targets for items found at rank = 1, we replace the easy items returned at rank 1 by the end of a conversation (turn 10) with a label = 0 (not found), and this number is larger for Shoes (15) than for Dresses (3). Therefore, removing more items based on model performance in Scenario 3 seems to blur the performance effect of Scenario 2 for the same amount of initial targets in the base scenario. To answer RQ6.1(a), CPP performance in Scenario 2 is comparable for the two scenarios for Shoes, and is slightly higher for Dresses.



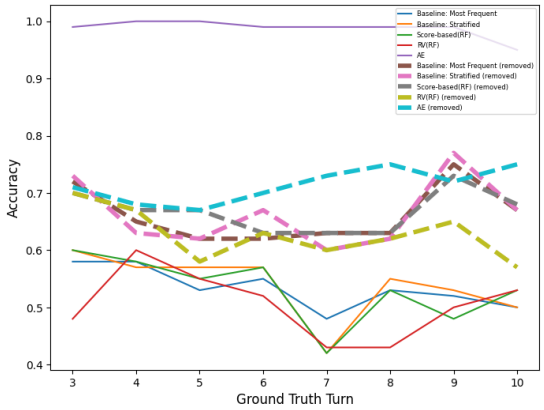

(a) Shoes rank=1




(b) Dresses rank=1



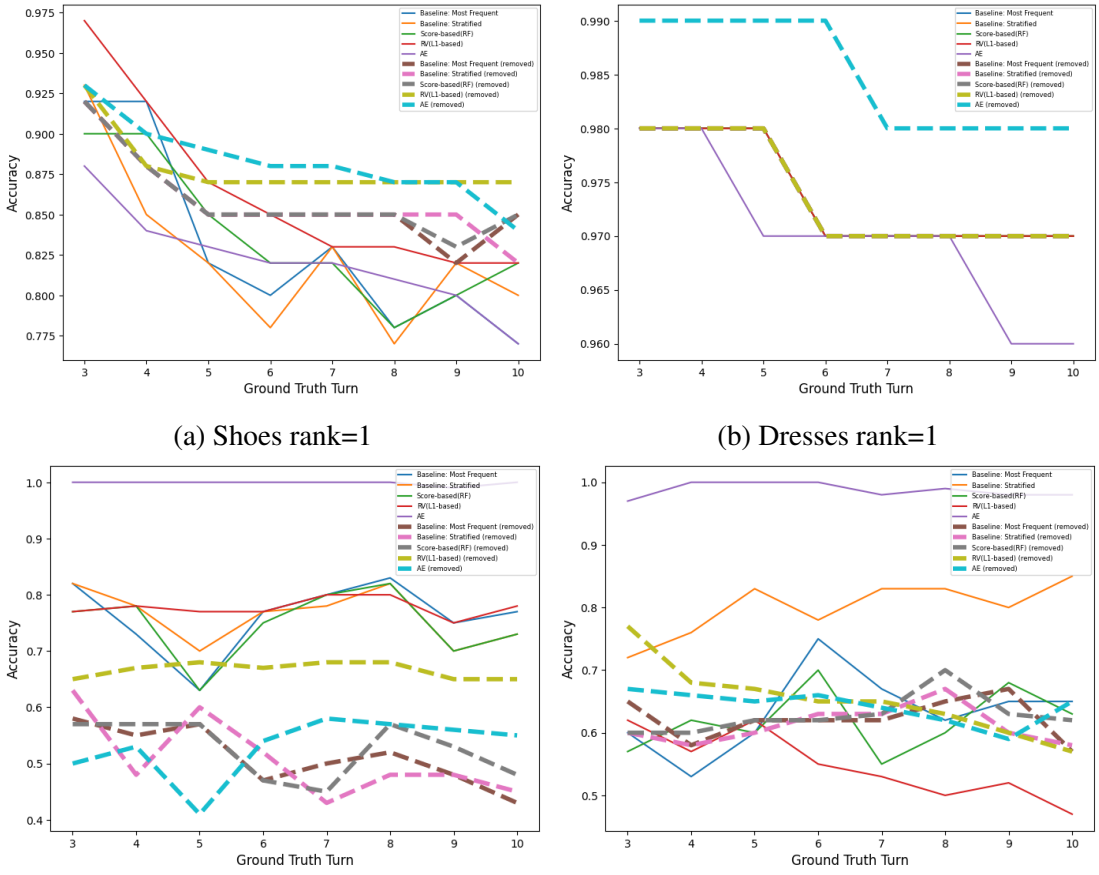

(c) Shoes rank=100




(d) Dresses rank=100

Figure 6.3: CPP Single-turn Results for Scenario 2 for the EGE model for a target item found at rank 1 (top figures) and rank 100 (bottom figures) for each dataset.

Next, we turn our attention to the items retrieved by rank 100, and therefore, predicting conversational success examining full rankings. For this, we look at plots 6.2 (c) and (d) and Figures 6.3 (c) and (d) for the GRU and the EGE model, respectively. Overall, we observe that going from Scenario 1 to Scenario 2 (after removing target items), the predictive accuracy of CPP predictors drops significantly (in the case of GRU Dresses this decrease is only marginal, and for AE there is a drop in performance instead). Also, we note that while in Scenario 1 AE performs notably better than the other predictors, in Scenario 2 this trend changes, and all predictors become comparable (AE is still the best predictor for EGE, but this difference is significantly decreased compared to the base scenario). Note that when removing selected target easy items for predicting items returned by rank 100, we replace 52 targets from Shoes and 27 from Dresses. This amount of removed targets is significantly larger than predicting the top-ranked item, and therefore, affects the accuracy of CPP more significantly than what we observed in RQ6.1(a), thus demonstrating the difficulty of the task of predicting a full ranking in the removed target case, thus answering RQ6.1(b).




(a) Shoes rank=1

(b) Dresses rank=1

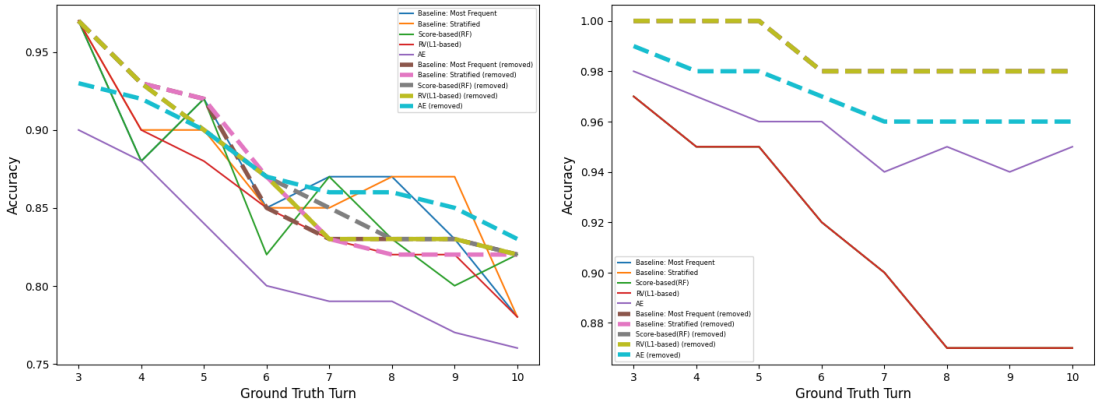
(c) Shoes rank=100

(d) Dresses rank=100

Figure 6.4: CPP Multi-turn Results for Scenario 2 for the GRU model for a target item found at rank 1 (top figures) and rank 100 (bottom figures) for each dataset.

### 6.3.2 Multi-turn CPP Results (Missing Target vs Base Scenario)

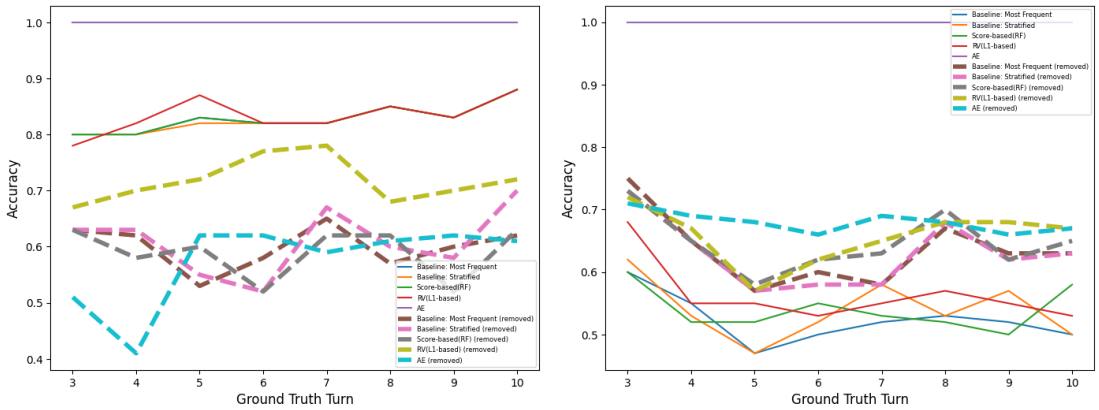
To examine the multi-turn CPP behaviour in the Removed Target Scenario, we focus on the multi-turn CPP predictors by examining the results in Figures 6.4 and 6.5, which display the CPP classification accuracy for the GRU and the EGE model, respectively. First, we examine the items found at rank 1 (Figure 6.4 (a) and (b), and 6.5 (a) and (b)) for both CRS models. In general, in all cases, we observe a similar pattern to the single-turn CPP results in RQ6.1(a); CPP performance slightly increases for all predictors after removing targets, and this is consistent across turns. One further observation is the reverse of the ordering of predictors: in most cases, while AE displayed a lower accuracy than RV (and the rest of predictors that mainly converge with RV) in Scenario 1, after removing targets (Scenario 2), AE becomes the best performing predictors (with a marked difference for Dresses and marginally for Shoes); on the other hand, for EGE Dresses, RV (together with score-based and the two baselines) outperforms AE, which was previously the best-performing predictor in Scenario 1. The differences among predictors are larger for Dresses than for Shoes. To answer RQ6.2(a), multi-turn CPP predictors change from Scenario 1 to Scenario 2 when predicting the top-ranked item, and in most cases AE




(a) Shoes rank=1




(b) Dresses rank=1




(c) Shoes rank=100




(d) Dresses rank=100

Figure 6.5: CPP Multi-turn Results for Scenario 2 for the EGE model for a target item found at rank 1 (top figures) and rank 100 (bottom figures) for each dataset.

increases when removing targets.

Finally, we focus on the deeper ranking cutoff multi-turn results (Figure 6.4 (c) and (d), and 6.5 (c) and (d)). A general trend observed across CRS models and datasets is that our proposed AE-based predictor stops being effective when introducing the removed target setting and its performance becomes comparable with other predictors (Dresses) or is outperformed by RV (Shoes). Indeed, for the multi-turn prediction case in deeper rankings, we see the usefulness of introducing a shrinkage-based predictor, and note its contribution to the change of evaluation setting in Scenario 2. Another general observation is that similarly to RQ6.3(b), CPP accuracy values decrease significantly when we move from Scenario 1 to Scenario 2. Once again, in mos cases it is difficult to distinguish the performance of the two baselines from most other predictors, which shows the increased difficulty of predicting at the conversation level, especially in the new scenario. Overall, the lower accuracy values in Scenario 2 indicate the difficulty of the new task (CPP prediction under the new introduced scenario) and indicate that CPP the predictive accuracy values among predictors start to converge, thus answering RQ6.2(b).

Therefore, based on the results obtained in our experiments that compare CPP performance

of Scenario 2 (Missing Target) with Scenario 1 (Existing Target), we conclude that: (i) For predicting the effectiveness of a ranking by predicting items found by rank 1, predicting catalogue failure is slightly easier than system failure. (ii) For predicting the effectiveness of a ranking by predicting items found by rank 100, predicting catalogue failure is a more difficult task than predicting than system failure.

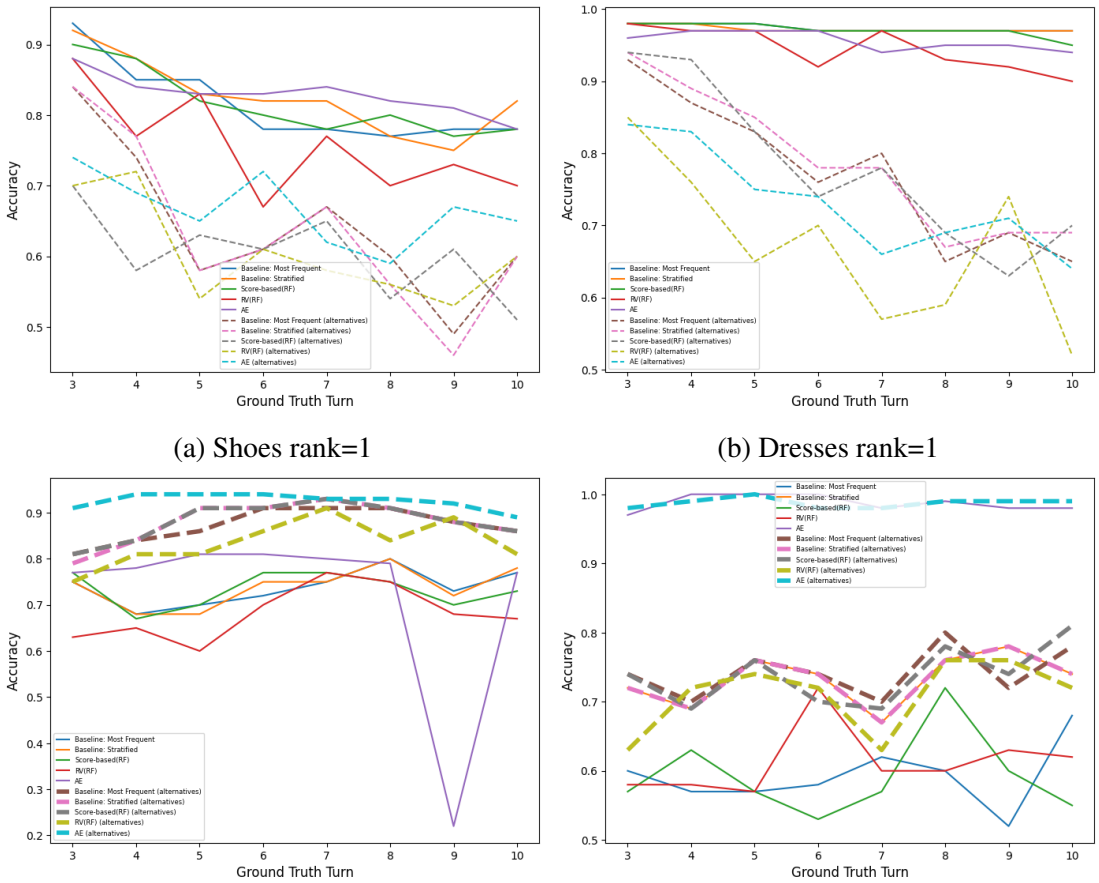
## 6.4 Results Alternatives (Scenario 3) vs Single Target (Scenario 1)

In this Section, we present the results for the Alternatives Scenario, where we used the alternative datasets collected in Chapter 5 and or new meta-simulator. In particular, we compare the CPP Results of the Alternatives Scenario (Scenario 3) with the base scenario with a single target (Scenario 1). First, we test the single-turn results in Section 6.4.1, and then we continue with the multi-turn results in Section 6.4.2.

### 6.4.1 Single-turn CPP Results (Alternatives vs Base Scenario)

We examine the single-turn CPP results in Figures 6.6 and 6.7, which display the CPP classification accuracy for the GRU and EGE models, respectively. While in Section 6.3 we examined single target items which were either available or missing from the catalogue, in this section, we compare the single and existing target of Scenario 1 with the Alternative Scenario or Scenario 3, which indicates the alternatives included in the target. In each plot, the x axis shows the ground truth turn of a conversation used for predictions, and the y axis is the predictive accuracy on the test set; each curve corresponds to a separate CPP predictor. The solid lines represent CPP predictors in Scenario 1 (base scenario), while the dashed lines correspond to the same predictors after introducing alternatives (Scenario 3) with a tolerance level after turn 2. The Shoes results are shown on the left and the Dresses results on the right. Also, in each figure, the top two plots show the results for predicting the success of a conversation at rank = 1, and the two bottom plots show predictions at rank = 100.

We start by describing the GRU results in Figure 6.6. First, we observe that when predicting the top-ranked item (plots 6.6 (a) and (b)), the results after introducing alternatives indicate a marked difference from Scenario 1. In particular, all solid lines show higher accuracy across turns than dashed lines. However, one issue for Scenario 1 CPP is that the baseline classifiers perform equally well and cannot be distinguished from both score-based and AE-based predictors up to turn 6; from turn 7, AE improves and (marginally) outperforms both baselines, and RV shows lower performance for Shoes, while for Dresses, there is no change. On the contrary, after introducing alternatives, for the Shoes dataset, while all predictors perform significantly lower, the two baselines significantly drop after turn 5, and they are outperformed by both AE




(a) Shoes rank=1

(b) Dresses rank=1

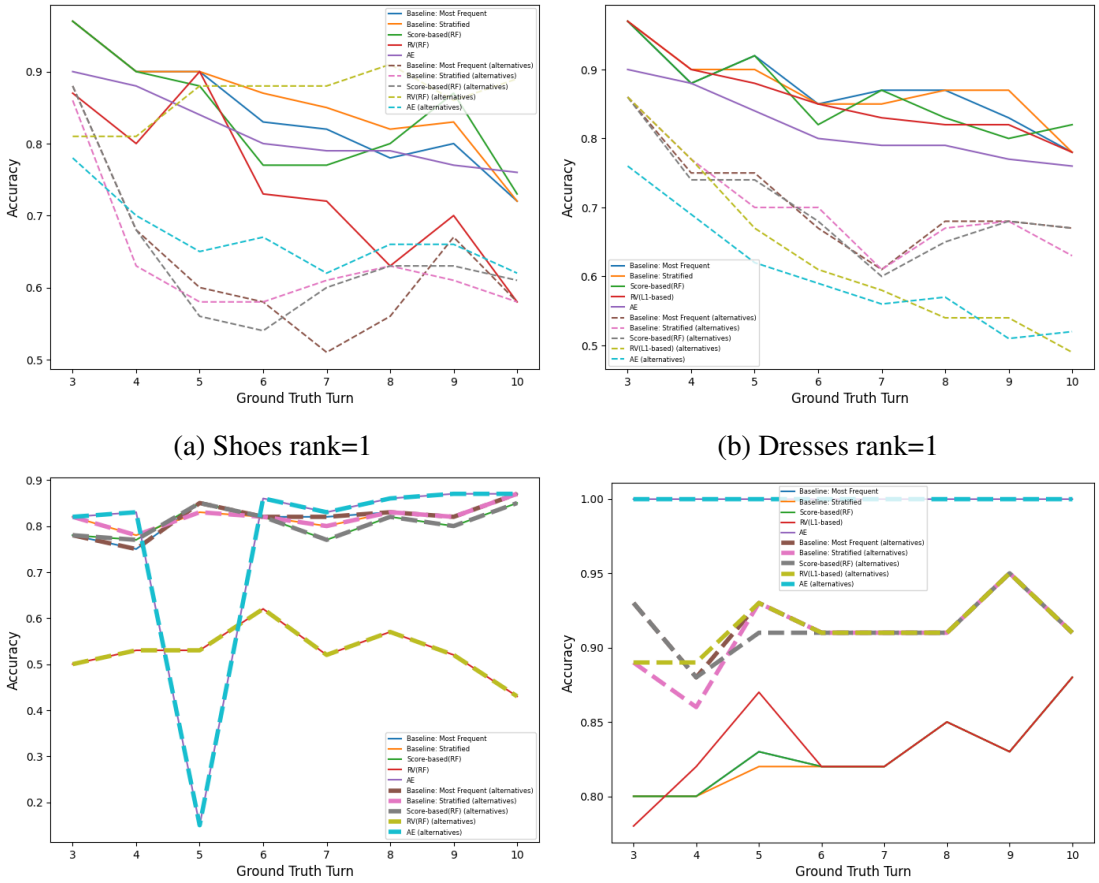
(c) Shoes rank=100

(d) Dresses rank=100

Figure 6.6: CPP Single-turn Results for Scenario 3 for the GRU model for a target item found at rank 1 (top figures) and rank 100 (bottom figures) for each dataset.

and score-based after turn 8. Also, for both scenarios, AE is generally the best performing predictor (especially after the first turns when system performance is becoming more stable), while the performance of RV(RF) is improving compared to the other predictors when introducing alternatives. On the other hand, the alternative-based setting affects Dresses differently; although the downward trend compared to the base scenario is similar to Shoes, both AE and RV perform lower than score-based, but still, the performance of baselines is competitive. The trend is only reversed towards the two final turns.

Moving on to the EGE model, we look at Figure 6.7 (a) and (b). For both datasets, we observe a similar pattern with GRU when moving from the base scenario to alternatives. Indeed, the dashed lines correspond to lower accuracy values than the solid lines, which indicates that when introducing alternatives, CPP performance drops significantly compared to a single target in the case of predicting the top-ranked item (or items found at rank 1). The only exception to this overall pattern is the performance of RV, which improves when introducing alternatives, and is also higher than all predictors in the base scenario after turn 6. Still, there is a difference between datasets in Scenario 3: For Shoes, our proposed predictors (especially RV and AE)




(a) Shoes rank=1

(b) Dresses rank=1

(c) Shoes rank=100

(d) Dresses rank=100

Figure 6.7: CPP Single-turn Results for Scenario 3 for the EGE model for a target item found at rank 1 (top figures) and rank 100 (bottom figures) for each dataset.

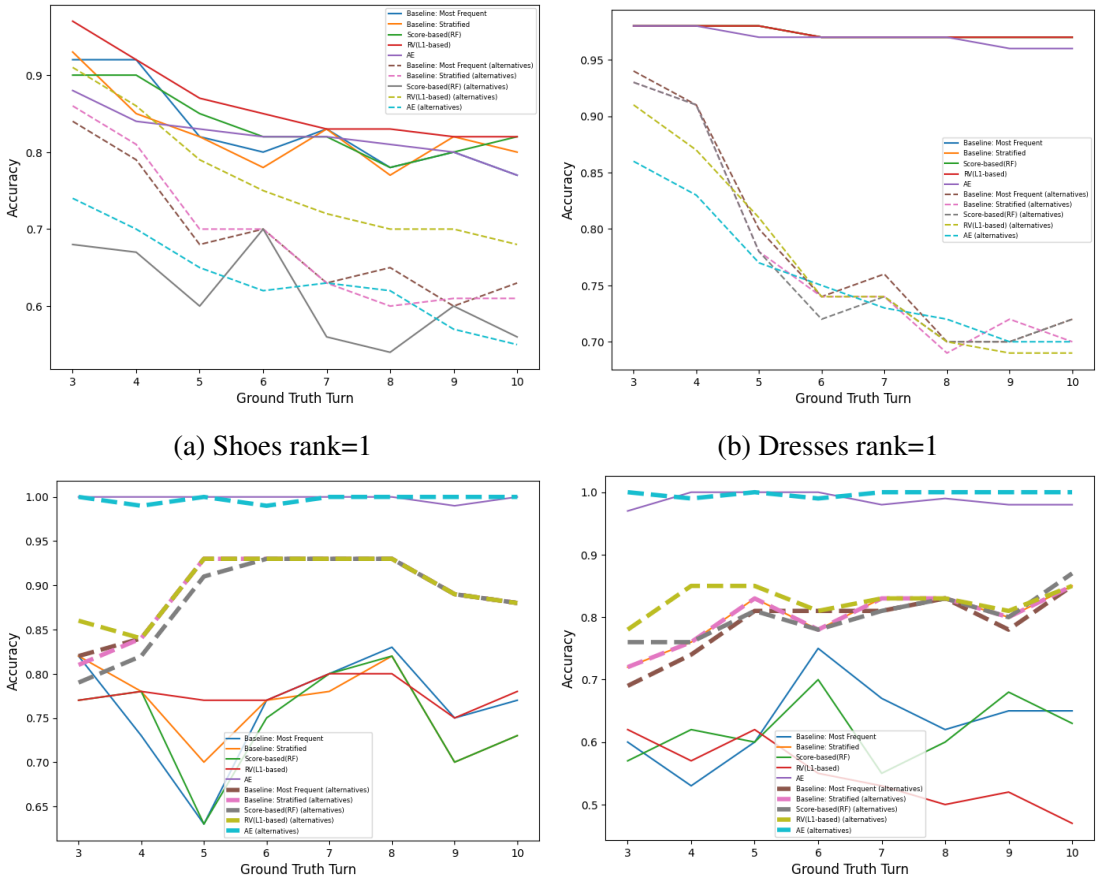
improve compared to the baselines when introducing alternatives, while for Dresses, RV and AE replicate the pattern of Scenario 1, which means they are lower than the baseline classifiers, while score-based perform equally well with them for both scenarios. Overall, the fact that in a lot of cases our baseline predictors do not outperform the baselines indicate the difficulty of predicting the top-ranked item, and indeed, this is an added difficulty compared to traditional QPP which focuses on predicting the effectiveness of a ranking by correlating QPP values with metrics at deeper rank cutoffs. Also, for both models, the decreased accuracy when moving from Scenario 1 to Scenario 3 indicates the difficulty of predicting conversational success in the alternative-based scenario, especially when we want to predict retrieved items at rank 1. We believe that part of this observation is due to the new evaluation setting that we introduced in Chapter 5. Specifically, each identified alternative for a given target item is now part of the new target space and is considered equally relevant. However, when we predict the top-returned item, only one of these alternatives is examined, and it might not be the alternative that was critiqued (because it was closer to the candidate at a given turn). Therefore, we believe that this is confusing for predicting items found at rank = 1, and therefore, results show reduced CPP

performance compared to the clearly defined target Scenario 1. This answers RQ6.3(a).

Next, we turn our attention to the items retrieved by rank 100, and therefore, predicting conversational success examining full rankings. First, we look at the GRU model (Figures 6.6 (c) and (d)), where we see that this time, introducing alternatives increases the accuracy of all CPP predictors across turns. Also, we note that AE is the only predictor that performs higher than the baseline classifiers for both scenarios (except for Scenario 1 at turns 9 and 10). In contrast, RV performs lower than the baselines in both cases (except for Scenario 1 Dresses turns 5-7), while score-based is equal to them. Moving to the EGE model (Figures 6.7 (c) and (d)), we observe that overall,introducing alternatives does not decrease CPP performance. Still, we observe differences between datasets. Specifically, for Shoes, each predictor performs equally for Scenario 1 and 3, and the only observed differences are the ones among predictors; in this case, score-based predictors are comparable with the baselines, RV is significantly lower, and AE, while initially much lower, significantly increases and marginally outperforms the rest after turn 6. In contrast, for Dresses, we see that introducing alternatives increases CPP performance across predictors, while the performance of most predictors is indistinguishable except for AE, which remains the highest in both scenarios (and is not increased, since was already high). We believe that the overall change of pattern from Scenario 1 to Scenario 3 (compared to predicting the top-ranked item) is observed because this time we are predicting effectiveness using a deeper rank cutoff, which contains the full set of target items. These results indicate that our proposed predictors are quite effective for predicting items found by rank 100, which is however an easier task compared to RQ6.3(a). Still, the increased accuracy is encouraging to show that we can effectively predict rankings in both scenarios, with an increased CPP effectiveness when adding alternatives, thus answering RQ6.3(b). In addition, we see that our proposed AE-based predictor is effective across scenarios.

### 6.4.2 Multi-turn CPP Results (Alternatives vs Base Scenario)

Here, we focus on the multi-turn CPP predictors by examining the results in Figures 6.8 and 6.9, which display the CPP classification accuracy for the GRU and the EGE model, respectively. First, we examine the items found at rank 1 (Figures 6.8 (a) and (b), and 6.9 (a) and (b)) for both CRS models. In general, in all cases, we observe a similar pattern to the single-turn CPP results in RQ6.3(a); CPP performance drops for all predictors when introducing alternatives, and this is consistent across turns. For GRU on Shoes, we observe that RV(L1-based) is the best-performing predictor for both scenarios, while the rest of the predictors are either comparable or slightly worse than the two baseline predictors. For GRU on Dresses, the picture is similar, but this time, AE and RV are equal for Scenario 1, while with alternatives, after turn 5 all predictors converge. Next, we turn to the EGE single-turn results (Figures 6.9 (a) and (b)). Again, we see a slight difference between datasets, while the overall pattern is similar (lower accuracy when introducing alternatives). More specifically, for Shoes, we see that apart from AE, which




(a) Shoes rank=1

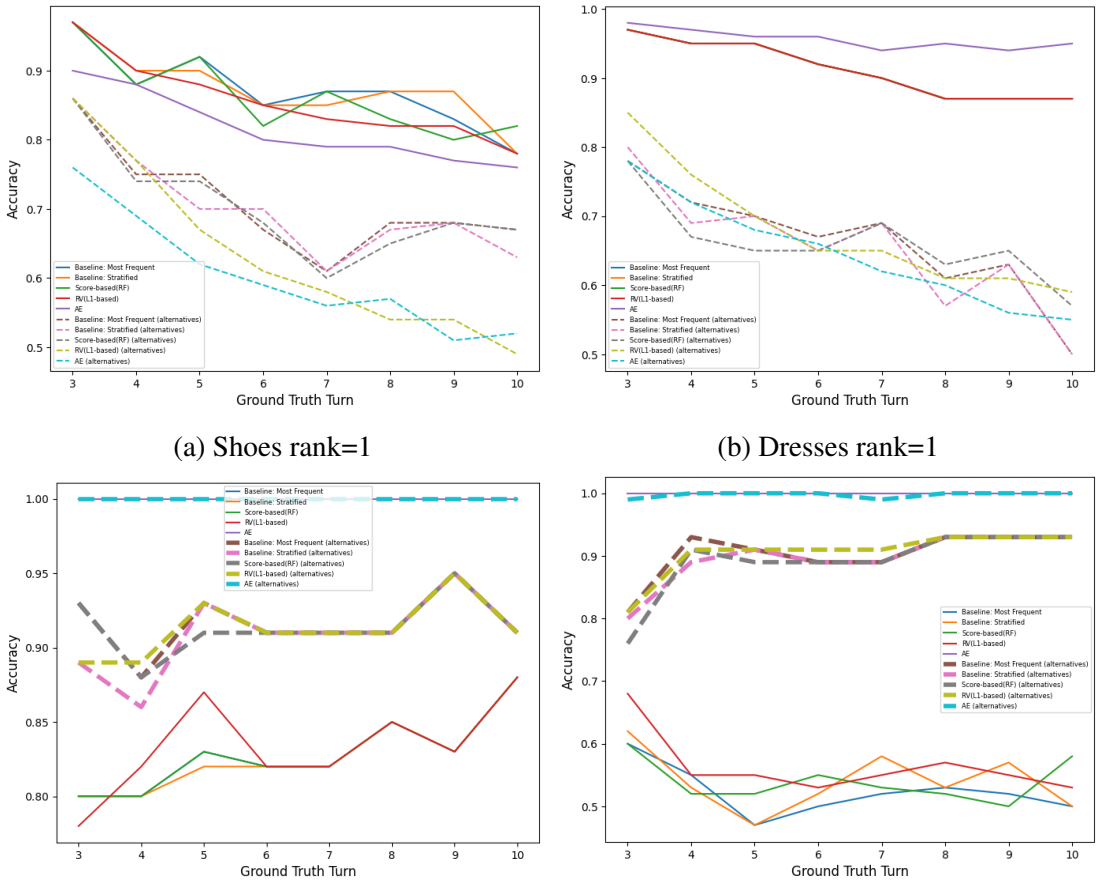
(b) Dresses rank=1

(c) Shoes rank=100

(d) Dresses rank=100

Figure 6.8: CPP Multi-turn Results for Scenario 3 for the GRU model for a target item found at rank 1 (top figures) and rank 100 (bottom figures) for each dataset.

performs slightly worse for both Scenarios, RV and score-based predictors display similar performance with the two baseline classifiers, especially for earlier turns. On the other hand, for Dresses in Scenario 1, we see that while RV, score-based, and the two baselines converge to the same accuracy across turns, AE is marginally better. Still, when we move to Scenario 3, its performance drops compared to the rest, and the different predictors start to converge. In general, the behaviour of the different classifiers for predicting conversation success at rank 1 mimics the results of RQ6.3(a) for the same cutoff, and we see that for both single-turn and multi-turn predictors, the task of predicting the top-ranked item is very demanding, not only when we introduce alternatives, but also in the base scenario. This highlights the difficulty of the CPP task in general, but some results need to be highlighted for specific cases: Our proposed shrinkage-based RV variant is quite effective especially for both models for the Shoes dataset in the base scenario, while AE is effective mainly for EGE Dresses. Also, we still believe that the decreased CPP accuracy when changing scenario (after introducing alternatives) in the case of predicting items returned at rank 1 is related to the new evaluation methodology that we proposed in Chapter 5. In particular, when we only examine rank = 1, it is less likely that the particular item of all




(a) Shoes rank=1

(b) Dresses rank=1

(c) Shoes rank=100

(d) Dresses rank=100

Figure 6.9: CPP Multi-turn Results for Scenario 3 for the EGE model for a target item found at rank 1 (top figures) and rank 100 (bottom figures) for each dataset.

set of alternatives that was actually critiqued will be returned, and therefore CPP performance is decreased, as it is confusing for the system when predicting items found at rank 1. This answers RQ6.4(a).

Finally, we focus on the deeper ranking cutoff multi-turn results. A general trend observed across CRS models and datasets is that for both scenarios, our proposed AE-based predictor is very effective and does not change when introducing alternatives. Another general observation is that similarly to RQ6.3(b), CPP accuracy values increase when we move from Scenario 1 to Scenario 3. Still, in all cases, except for AE, the rest of the predictors display very similar behaviour and their displayed accuracies are not clearly separated. Therefore, while predicting a deeper ranking (at the conversation level) seems like an easier task than predicting what is returned as the top item, the task of predicting the top-100 items also leads to the following result: Our proposed AE-based predictor is the one that converges results from both scenarios, both for single-turn predictors, but most importantly for multi-turn predictors. This result makes sense, as for the multi-turn predictor variant, we use the top-100 item representations, of which the predictor learns a compressed version. This answers RQ6.4(b).

Therefore, based on the results obtained in our experiments that compare CPP performance of Scenario 3 (Alternatives) with Scenario 1 (Single Target), we conclude that: (i) For predicting the effectiveness of a ranking by predicting items found by rank 1, predicting alternative failure is more difficult than system failure. (ii) For predicting the effectiveness of a ranking by predicting items found by rank 100, predicting alternative failure is an easier task than predicting than system failure.

## 6.5 Concluding Remarks

In this chapter, we have extended our Conversational Performance Prediction (CPP) framework (introduced in Chapter 4), which proposed predicting performance of CRS models at the conversation level using indicators of effective performance of a conversation under different evaluation settings. In particular, we introduced the concept of *recommendation scenarios* (Section 6.1.1), and transferred the CPP framework to a range of real-world applications. To achieve this, we introduced two parameters or variation factors, which influence the nature and flow of the Conversational Image Recommendation task and result in modified settings. More specifically, based on the availability of the user's target item in the item catalogue, we can have a Missing Target Scenario (or Scenario 2), which indicates the item is missing from the catalogue and cannot be returned to the user. Additionally, based on how clearly a target is defined, we can have the Alternative Scenario (or Scenario 3), where a number of equally reinforcing items can satisfy the user need, and therefore can be characterised as alternatives to the original target and join a common target space. Based on Scenarios 2 and 3, we defined two additional types of conversational failure different from the one introduced in Chapter 4 (system failure), namely catalogue failure and alternative failure. After redefining the CPP task for each recommendation scenario (Section 6.1.2), we tested both catalogue failure and alternative failure in comparison to the traditional system failure in Sections 6.3 and 6.4 using both the single-turn and the multi-turn variants of CPP predictors.

Starting with Scenario 2, the examination of single-turn predictors when predicting the top-ranked item (RQ6.1(a) in Section 6.3.1) showed that system and catalogue failures cannot easily be distinguished; there is only a marginal increase in CPP performance when moving to Scenario 2, and this is mainly due to the fact that very few items were removed as easy items. In contrast, when predicting conversation performance at deeper rank cutoffs (using items found at rank 100) (RQ6.1(b) in Section 6.3.1), we noted a noticeable decrease in CPP performance in most cases. This demonstrates the difficulty of predicting full rankings in the Removed Target Scenario, and these results are more representative of the actual effect of Scenario 2, since we removed a sufficient amount of items (based on the criterion of being returned by rank 100). The results of single-turn prediction are generally replicated when using multi-turn predictors to compare Scenario 2 to the base scenario; for predicting the top item (RQ6.2(a) in Section 6.3.2),

there is a slight increase in Scenario 2, while for deeper rankings (RQ6.2(b) in Section 6.3.2), we have not just a decrease in CPP performance, but also a change in the ordering of predictors. Contrary to the results of Scenario 2, the results for Scenario 3 display a different pattern overall. In particular, when predicting the top-ranked item, we observe a marked decrease in CPP performance after introducing alternatives, both for single-turn (RQ6.3(a) in Section 6.4.1) and for multi-turn (RQ6.4(a) in Section 6.4.2) predictors. We believe that this result is due to the confusion induced by predicting the top-ranked item while more target items are considered as equally relevant. On the other hand, when predicting items returned by rank 100, we note an increase in CPP performance across predictors for both single-turn (RQ6.3(b) in Section 6.4.1) and multi-turn (RQ6.4(b) in Section 6.4.2) prediction. This is is line with the fact that predicting a deeper ranking is likely to contain the entire new target space with alternatives.

Therefore, we see that while for Scenario 2 (Missing Target), predicting items found at deeper rank cutoffs such as rank 100 is an easier task than predicting the top-ranked item, for Scenario 3 (Alternatives), predicting target items using deeper rankings is an easier task than relying on predictions about target items returned at the top rank. This finding is reasonable, since for the alternatives case, predicting items found at rank 1 is more likely to lead to conversation failure, while for the removed target case, predicting a full ranking after removing items returned at a deeper rank (easier items) would likely result in a catalogue failure. Overall, our results indicate that predicting conversational failures under different evaluation settings (scenarios) is indeed insightful, and reveal that based on the nature of the task, CPP can differently affect predictions according to both the depth of the ranking and the specific CPP predictors. We further accomplished a connection of CPP task to the new evaluation setting introduced in Chapter 5, where we collected real user opinions about alternative options. Consequently, we showed that CPP is an evaluation framework that depends on the specific recommendation scenario, and therefore, can be extended to other QPP-based tasks, such as QPP in ad-hoc retrieval or Conversational Search. In any case, we showed that CPP, in the same way as QPP, is a task (in the supervised case) that can sufficiently be predicted by using the embedded representations already contained in the CRS models, thus linking back to Chapter 3. Finally, overall in this chapter, we have validated the fourth claim of the thesis statement which stated that *by using these alternatives datasets, and by predicting conversational performance under different Recommendation Scenarios, we detect different types of conversational failure, such as when a user cannot find an item, versus when the system's catalogue does not contain the relevant item.* For future work, we aim to further refine our CPP predictors with more complicated structures and test whether this would have an effect on better capturing the underlying complexity of the CRS model behaviour.

# Chapter 7

# Conclusions

## 7.1 Identified Challenges

This thesis addressed the problem of predicting the different types of retrieval failure that can occur during an interaction with a multi-turn and multi-modal Conversational Recommendation System (CRS). In particular, we argued that the performance of a Conversational Recommendation System can be predicted to detect when a conversation fails, under different scenarios, across different turns of a conversation. In Chapter 1, we posed a number of challenges for predicting conversational performance, which can be summarised as follows:

- **Challenge 1:** How to predict the effectiveness of ranking lists of items in multi-modal Conversational Recommendation Systems (CRS) that are composed of image-based result lists resulting from text-based user feedback (This is linked to **Limitation 3** in Section 2.3.3);

- **Challenge 2:** How to predict the degree of success of a multi-turn CRS conversation and to differentiate between predicting the current user satisfaction or the overall satisfaction of a conversation (This is linked to **Limitations 4a) and 4b)** in Section 2.3.3);

- **Challenge 3:** How to make fashion CRS system interactions more realistic by accounting for more flexible user needs and preferences, for example by examining more options as desired fashion items, to create the foundations for more accurate predictions (This is linked to **Limitations 1a) and 1b)** in Section 2.2.3);

- **Challenge 4:** How to account for the type of conversational failure (CRS responding differently when a user cannot find an item than when the underlying catalog does not contain the item) and therefore, make predictions under various conversational recommendation settings (This is linked to **Limitations 1a), 2a), and 2b)** in Section 2.2.3).

## 7.2 Contributions and Conclusions

For the purpose of addressing the aforementioned challenges, the contributions of this thesis are as follows:

- We proposed a number of embedding-based query performance predictors for text-based dense retrieval. As we explained in Chapter 3, Conversational Image Recommendation models can be seen as a subset of dense retrieval models that use image-based results lists rather than text.
- We contributed a new evaluation framework that considers the multi-turn aspect of the Conversational Image Recommendation task extends, and proposed a number of unsupervised and supervised predictors within the context of this framework.
- We collected user opinions for alternatives to existing target items, thus improving the completeness of CRS evaluation, and finally, we proposed two novel recommendation scenarios that more accurately capture the span of user needs and recommendation failures. We tested the ability to detect these scenarios using our alternative-based datasets.

In the following, we discuss in more detail our main conclusions in addressing these challenges, before we explain how we answer each claim of the thesis statement:

- **Conclusion 1: Coherence-based Query Performance Measures for Dense Retrieval:** To address the first challenge, we studied the query performance prediction (QPP) task in the text-based dense retrieval setting, before transferring it to a multi-turn CRS recommendation setting, with a purpose of detecting which factors affect query performance of the type of retrieval models that can easily be generalised to our multi-modal task of interest (see Chapter 3). In this regard, we studied QPP on two popular single-representation dense retrieval models, ANCE (Xiong et al., 2020) and TCT-ColBERT (Lin et al., 2020), and proposed a set of dense coherence-based predictors (Section 3.2.2), which are based on the intuition of average top vs. bottom rank pairwise similarities of top-ranked embeddings (higher pairwise similarity among documents of top ranks, and lower correlation for lower ranks). The results of Section 3.4 demonstrate that our proposed predictors can significantly improve performance compared to state-of-the-art supervised predictors, while they improved performance compared to score-based predictors for NDCG@10 and MRR@10. Moreover, in Section 3.5, we proposed a multi-level statistical approach to further explain why our predictors displayed lower correlations than score-based predictors for MAP@100. The results of Section 3.5.3 showed that the interplay of QPPs with *query types* (as these were proposed by Bolotova et al. (2022)), contributes to the unstable performance of QPPs only for MAP@100.

- **Conclusion 2: Conversational Performance Prediction (CPP)** To address the second challenge, we proposed a framework for *Conversational Performance Prediction (CPP)*, which predicts retrieval failures in a conversational recommendation setting by considering the recommendation ranking at different turns of a conversation, using both single-turn and consecutive-turn predictors (Section 4.2). In this framework, we predict performance in the context of recommendation models at the conversation level going further than the single ranking focus in QPP literature. In this regard, after focusing on simple score-based predictors to test the difference between prediction horizons in Sections 4.3.2 and 4.3.3, we tested our proposed predictors from Chapter 3 in the new setting (Sections 4.3.4 and 4.3.5). Overall, the findings of Section 4.2 indicated that long-term prediction does not work under this specified evaluation setting, but short-term predictions provide small to medium correlations, and in all cases, correlations were consistently lower than the ones in the QPP setting. To account for this, we also examined a supervised CPP evaluation methodology (Section 4.4), where we treated CPP as a binary classification task classifying if a given conversation will result in the user's target item being successfully retrieved or not. In this regard, we further proposed a new embedding-based supervised predictor (inspired by supervised QPP predictors) that learns a compressed representation of the retrieved item(s) of previous turn(s) up to the turn prior to the evaluation turn. In our experiments (Sections 4.4.3 and 4.4.4), we found that using classifier-based evaluation and the predictive accuracy of a predictor on the test set more effectively captures the underlying nature of a multi-turn conversation and shows high accuracy across both single-turn and multi-turn predictions.

- **Conclusion 3: Evaluating User Simulators with Alternatives** To address the third identified challenge, in Chapter 5, we addressed the issue of evaluation completeness in CRS models. In particular, we collected real user opinions in a dataset for fashion recommendation that contains labels about the presence of sufficient alternatives for a number of known target items on different fashion item categories, namely Shoes and Dresses (Section 5.3). This was achieved by using pooling from different CRS models with a process described in detail in Section 5.3.2. Next, using this new dataset with the updated target space, we modified the original user simulator using a new Meta-Simulator that provides feedback by considering the identified alternatives (Section 5.2) and reran the CRS models. In our experiments in Section 5.4, we found improved CRS performance compared to the non-alternative setup, indicating that evaluation with user simulators applying a single target, as has been used in all previous CRS literature for these datasets, were underestimating system performance.

- **Conclusion 4: Predicting Conversation Performance across Recommendations Scenarios** Finally, to address our fourth identified challenge, we introduced the concept two

novel recommendation scenarios in Chapter 6. Specifically, by considering two factors of variation, namely the definition and the availability of the target item, we define the *Missing Target or Scenario 2* and the *Alternative or Scenario 3* scenarios (Section 6.1). Scenario 3 uses the alternative relevance labels identified in Chapter 5. Consequently, we introduce different types of recommendation failures resulting from each scenario, and consequently, we differentiate between the system not being able to retrieve the correct item and the item not being available. For this purpose, we use the CPP predictors defined in Chapter 4. Our experiments in Sections 6.3 and 6.4 indicated that it is worth-exploring CPP under different recommendation scenarios, as the prediction accuracy differences are marked, both when we use Scenario 2 and Scenario 3.

Based on the results obtained in Chapters 3 to 6, we validate the thesis statement in Section 1.2, according to which the performance of a Conversational Recommendation System can be predicted to detect when a conversation fails, under different scenarios, across different turns of a conversation. Below, we discuss how the main experimental results obtained in each chapter validate each corresponding cliam of the thesis statement.

- **Claim 1:** *Initially, we can predict the effectiveness of a ranking of textual items for a textual query, by examining the coherence of the top-retrieved items based on their dense embedded representations.* Our experiments in Chapter 3 validated this claim by showing that our proposed dense coherence-based predictors on the TREC Deep Learning Track datasets demonstrate improved accuracy upon dense retrieval (up to 92% compared to sparse variants for TCT-ColBERT and 188% for ANCE when we correlate with NDCG@10) (Tables 3.2 and 3.3). We also showed that this pattern is unique to dense retrieval (Tables 3.9 and 3.10), a type of retrieval models that our CRS task of interest is a part of, due to the interaction of QPPs with the types of queries. This was particularly evident for one of the metrics, namely MAP@100 (Figure 3.3), while for metrics such as NDCG@10 and MRR@10, our proposed predictors more consistently outperform the baselines due to the reduced influence of query types.

- **Claim 2:** *Similarly, we can predict the effectiveness of a ranking of items in a Conversational Recommendation Systems (CRS), which are also based on learned embedded representation of images, where user feedback takes the place of a textual query. Indeed, by introducing a framework of Conversational Performance Prediction (CPP), we can predict the degree of success of a conversation by a CRS - such success can be predicted over a short or long time horizon, thereby predicting current user satisfaction or overall satisfaction of a conversation.* We validated this claim in Chapter 4, where we proposed a novel Conversational Performance Prediction (CPP) Framework, aiming to predict performance of CRS systems at the conversation level instead of at the query level in CRS systems. First, the results using all target image items contained in the relative caption-

ing train datasets, namely Shoes (Berg et al., 2010; Guo et al., 2018) and FashionIQ (Wu et al., 2021a), showed that it is possible to predict the performance of a conversation using score-based predictors in the short-term, especially for Shoes (Table 4.2 first group of rows reaching up to a 0.423 Spearman's correlation for early turns), while it is also sufficient for Shirts (Table 4.2 last group of rows reaching up to a 0.336 Spearman's correlation for early turns). Next, we examined an evaluation setting using sampled target items, corresponding to the QPP query sets and focusing on short-term prediction, and showed that mainly score-based CPP predictors perform well for GRU (Table 4.3 up to a Spearman's correlation of 0.339 for Shoes, NQC, MRR@10), while for EGE, the results are split; in some cases, score-based predictors are more effective (Table 4.4 up to 0.282 correlation for Shoes, NQC, MRR@10), while in others, embedding-based predictors win (Table 4.4 up to 0.291 correlation for Shoes, RV, NDCG@10). Finally, we empirically tested the supervised version of CPP as a classification task, which revealed the most promising results. Specifically, the results for both single-turn predictors (Tables 4.5 and 4.6) and the multi-turn results in Tables 4.7 and 4.8 indicate high accuracy values for the various CPP predictor supervised variants, and especially for multi-turn prediction (train turns up to a given turn and prediction at the next turn), supervised embedding-based predictors consistently performed with an accuracy above 95% for various rank cutoffs.

- **Claim 3:** *Furthermore, by obtaining user opinions about the relevance of items, we improve the completeness of the evaluation mechanism by identifying alternatives recommendations for existing target items, which could be used to both inform the user simulator and therefore improve the overall evaluation of CRS system.* We validated this claim in Chapter 5. Indeed, we collected pooled relevance judgments that reflect the simulator's knowledge of alternatives. We re-evaluated three CRS models before and after alternatives and as it can be seen in Tables 5.4 and 5.5, CRS performance is noticeably higher after introducing alternatives, reaching up to 99% improvement for Shoes EGE for SR@10, and up to 146% for Dresses EGE for MRR@10. We also observed that the specific tolerance level is not statistically significant for CRS performance (Figures 5.5 and 5.6, and Table 5.6). This means that while introducing alternatives changes the magnitude of CRS performance, the specific turn at which a user loses patience is not further particularly crucial. Still, the tolerance level is crucial for how rapid the increase in performance is (Figure 5.8).

- **Claim 4:** *Finally, using these alternatives datasets, and by predicting conversational performance under different Recommendation Scenarios, we detect different types of conversational failure, such as when a user cannot find an item, versus when the system's catalogue does not contain the relevant item.* Indeed, we validated this claim in Chapter 6, where we tested our proposed supervised CPP predictors from Chapter 4 under the

different recommendation scenarios. The main intuitions can be summarised as: (i) For both single-turn and multi-turn prediction, moving to Scenario 2 increases CPP performance when predicting the top item, while it decreases its performance when predicting item found at deeper rank cutoffs (Figures 6.2 to 6.5). On the other hand, for both single-turn and multi-turn prediction, moving to Scenario 3 decreases CPP performance when predicting the top item, while it increases its performance when predicting items found at deeper rank cutoffs (Figures 6.6 to 6.9). Overall, the CPP differences among scenarios indicate the presence of different cases of conversational failure under different scenarios and according to the depth of the predicted ranking. In addition, we linked the previously identified alternative items to a given target with CPP prediction in the corresponding novel scenario.

We have validated all the claims, and therefore, the thesis statement (Section 1.2) has been validated. Next, we describe some future directions that arise from the insights obtained in the experiments of this thesis.

## 7.3 Future Directions

Here, we propose some possible directions for future steps to enhance both our CPP framework and also incorporate the insights obtained in this thesis to improve CRS models.

**Probabilistic Interpretation of CPP:** In Chapter 4, we posed the argument that in the multi-turn recommendation setting, it is difficult to distinguish what pre-retrieval and post-retrieval predictors mean. In this regard, a probabilistic framework was originally proposed for QPP (Shtok et al., 2016), which assumes pre-retrieval predictors can be considered as prior information and post-retrieval predictors can be considered as posterior information, according to a Bayesian interpretation. One possible direction is to extend this approach into a multi-turn recommendation setting with the possibility of combining textual and image representations to predict performance. In particular, a CPP approach needs to show how the textual embedded representations (resulting from user feedback) can be used in parallel with image embedded representations (resulting from retrieved images at a given turn) in order to predict the performance of CRS systems. For example, a CPP predictor that can address the multi-turn aspect in CPP should be able to quantify the predictions at each turn, update the predictive probability of success to the next turn, and result in a final probability at the end of a conversation. Also, it should address the nature of CRS that include multiple modalities (i.e., both text and images), thus accounting for the interplay between these modalities and their link across turns. In this way, any new turn could be added and evaluated as new test input data, and the final probability of conversational success could be measured at a given "final" turn.

**Interventions with CPP predictions and Alternatives:** In Chapters 4 and 6, we showed that conversational performance in CRS models can be predicted across different scenarios. Still,

currently, our CPP insights are not used to enhance the performance of CRS models by incorporating these predictions. Existing Conversational Image Recommendation models (Guo et al., 2018; Wu et al., 2021a,b) recommend items from the beginning until the end of a dialogue, but do not ask clarifying questions or update user preferences systematically. This delays the target item, while informative statements by the system would result in its faster identification (earlier turn). For example, using specific insights resulting form QPP or CPP indicators when the system informs the user or when it makes a new recommendation would result in improved CRS performance. In this regard, CPP predictors would be indicative of when to make an intervention, which is more reasonable than relying on a static number of turns. Also, we could modify the system to provide instructions about what the catalogue contains, and what differentiates the ranked results from the rest. This would also require validation with a user study.

**Prediction Differences of CRS from Conversational Search:** While some predictors were found to be effective for QPP in a Conversational Search setting (Faggioli et al., 2023a; Meng et al., 2023), they do not seem to predict equally well in a CRS setting, as indicated by our CPP experiments in Chapters 4 and 6. Instead, for CPP, we need to take into account the learned representations in a sequential way throughout the conversation. At the same time, it would be useful to examine the differences between the queries Conversational Search, and the captions in Conversational Image Recommendation. For example, a feedback utterance might contain certain aspects that lead a conversation to be more or less effective, and this in turn, influences predictions. We avoided this due to only having user simulator feedback in this thesis. Therefore, it would be useful to also examine and compare the content of the critiques between the tasks.

## 7.4 Concluding Remarks

This thesis has addressed the topic of evaluation in Conversational Image Recommendation models in the fashion domain. In particular, this thesis has proposed a novel evaluation framework for predicting different types of failure in Conversational Recommendation Systems (CRSs). We have demonstrated that performance of a CRS can be predicted to detect when a conversation fails, under different scenarios, across different turns of a conversation. In addition, we have shown that using multi-turn predictions for our prediction framework works best when we treat the problem as a classification task that predicts whether a target item is found at a given turn. In Section 7.3, we have proposed a new directions for the methodology proposed in this thesis. While we have shown how we can detect retrieval failure under different recommendation settings, we believe that incorporating these insights into current CRS models and further developing the evaluation methodology interactively across turns is a promising direction for future work in conversational recommendation.

# Bibliography

M. Aliannejadi, H. Zamani, F. Crestani, and W. B. Croft. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm SIGIR conference on research and development in information retrieval*, pages 475–484, 2019.

A. Anand, L. Cavedon, H. Joho, M. Sanderson, and B. Stein. Conversational search (dagstuhl seminar 19461). In *Dagstuhl Reports*, volume 9. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

S. Andolina, V. Orso, H. Schneider, K. Klouche, T. Ruotsalo, L. Gamberini, and G. Jacucci. Investigating proactive search support in conversations. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 1295–1307, 2018.

N. Arabzadeh, F. Zarrinkalam, J. Jovanovic, F. Al-Obeidat, and E. Bagheri. Neural embedding-based specificity metrics for pre-retrieval query performance prediction. *Information Processing & Management*, 57(4):102248, 2020.

N. Arabzadeh, A. Bigdeli, M. Zihayat, and E. Bagheri. Query performance prediction through retrieval coherency. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, pages 193–200. Springer, 2021a.

N. Arabzadeh, M. Khodabakhsh, and E. Bagheri. Bert-qpp: Contextualized pre-trained transformers for query performance prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2857–2861, 2021b.

N. Arabzadeh, M. Seifikar, and C. L. Clarke. Unsupervised question clarity prediction through retrieved item coherency. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3811–3816, 2022.

A. Argal, S. Gupta, A. Modi, P. Pandey, S. Shim, and C. Choo. Intelligent travel chatbot for predictive recommendation in echo platform. In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pages 176–183. IEEE, 2018.

K. Balog. Conversational ai from an information retrieval perspective: Remaining challenges and a case for user simulation. *DESIRES*, 2021.

D. Bates, M. Maechler, B. Bolker, S. Walker, R. H. B. Christensen, H. Singmann, B. Dai, F. Scheipl, and G. Grothendieck. Package 'lme4'. 2009.

N. J. Belkin and W. B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

N. J. Belkin, C. Cool, D. Kelly, S.-J. Lin, S. Park, J. Perez-Carballo, and C. Sikora. Iterative exploration, design and evaluation of support for query reformulation in interactive information retrieval. *Information Processing & Management*, 37(3):403–434, 2001.

M. S. Ben-Shachar, D. Makowski, D. Lüdecke, I. Patil, B. Wiernik, K. Kelley, D. Stanley, J. Burnett, and J. Karreth. Package 'effectsize', 2022.

T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 663–676. Springer, 2010.

V. Bolotova, V. Blinov, F. Scholer, W. B. Croft, and M. Sanderson. A non-factoid question-answering taxonomy. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1196–1207, 2022.

T. M. Brill, L. Munoz, and R. J. Miller. Siri, Alexa, and other digital assistants: a study of customer satisfaction with artificial intelligence applications. *Journal of Marketing Management*, 35(15-16):1401–1436, 2019.

A. Broder. A taxonomy of web search. In *ACM SIGIR forum*, volume 36, pages 3–10, 2002.

C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, 2004.

C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. Bias and the limits of pooling for large collections. *Information retrieval*, 10:491–508, 2007.

V. S. Bursztyn, J. Healey, E. Koh, N. Lipka, and L. Birnbaum. Developing a conversational recommendation systemfor navigating limited options. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2021.

W. Cai, Y. Jin, and L. Chen. Critiquing for music exploration in conversational recommender systems. In *26th International Conference on Intelligent User Interfaces*, pages 480–490, 2021.

D. Carmel and E. Yom-Tov. Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2(1):1–89, 2010.

D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. What makes a query difficult? In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 390–397, 2006.

S. Champely, C. Ekstrom, P. Dalgaard, J. Gill, S. Weibelzahl, A. Anandkumar, C. Ford, R. Volcic, H. De Rosario, and M. H. De Rosario. Package 'pwr'. *R package version*, 1(2), 2018.

A. J. B. Chaney, B. M. Stewart, and B. E. Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 224–232, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359016. doi: 10.1145/3240323.3240370. URL https://doi.org/10.1145/3240323.3240370.

L. Chen and P. Pu. Hybrid critiquing-based recommender systems. In *Proc. IUI*, pages 22–31, 2007.

L. Chen and P. Pu. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction*, 22(1):125–150, 2012.

M. Chen. Exploration in recommender systems. In *Fifteenth ACM Conference on Recommender Systems*, pages 551–553, 2021.

Q. Chen, J. Lin, Y. Zhang, M. Ding, Y. Cen, H. Yang, and J. Tang. Towards knowledge-based recommender dialog system. *arXiv preprint arXiv:1908.05391*, 2019.

A.-G. Chifu, L. Laporte, J. Mothe, and M. Z. Ullah. Query performance prediction focused on summarized letor features. In *Proc. SIGIR*, 2018.

K. Cho. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

K. Christakopoulou, F. Radlinski, and K. Hofmann. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 815–824, 2016.

K. Christakopoulou, A. Beutel, R. Li, S. Jain, and E. H. Chi. Q&r: A two-stage approach toward interactive recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 139–148, 2018.

V. Christophides, V. Efthymiou, and K. Stefanidis. Entity resolution in the web of data. *Synthesis Lectures on the Semantic Web*, 5(3):1–122, 2015.

G. Chung. Developing a flexible spoken dialog system using simulation. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pages 63–70, 2004.

C. Cleverdon, J. Mills, and M. Keen. Factors determining the performance of indexing systems. 1966.

N. Craswell and D. Hawking. Overview of the trec-2002 web track. In *Proceedings of TREC-2002*, 2002.

N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pages 87–94, 2008.

N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*, 2020.

N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. Overview of the trec 2020 deep learning track. *arXiv e-prints*, pages arXiv–2102, 2021.

W. B. Croft and R. H. Thompson. I3r: A new approach to the design of document retrieval systems. *Journal of the american society for information science*, 38(6):389–404, 1987.

S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, 2002.

R. Cummins. Document score distribution models for query performance inference and prediction. *ACM Transactions on Information Systems (TOIS)*, 32(1):1–28, 2014.

R. Cummins, J. Jose, and C. O'Riordan. Improved query performance prediction using standard deviation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1089–1090, 2011.

P. J. Curran, E. Stice, and L. Chassin. The relation between adolescent alcohol use and peer alcohol use: a longitudinal random coefficients model. *Journal of consulting and clinical psychology*, 65(1):130, 1997.

J. Dalton, V. Ajayi, and R. Main. Vote goat: Conversational movie recommendation. In *The 41st international acm sigir conference on research & development in information retrieval*, pages 1285–1288, 2018.

J. Dalton, C. Xiong, and J. Callan. Trec cast 2019: The conversational assistance track overview. *arXiv preprint arXiv:2003.13624*, 2020a.

J. Dalton, C. Xiong, V. Kumar, and J. Callan. Cast-19: A dataset for conversational information seeking. In *Proc. SIGIR*, 2020b.

S. Datta, D. Ganguly, D. Greene, and M. Mitra. Deep-qpp: A pairwise interaction-based deep learning model for supervised query performance prediction. In *Proceedings of the fifteenth ACM international conference on Web Search and Data Mining*, pages 201–209, 2022a.

S. Datta, S. MacAvaney, D. Ganguly, and D. Greene. A'pointwise-query, listwise-document'based query performance prediction approach. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2148–2153, 2022b.

S. Datta, D. Ganguly, J. Mothe, and M. Z. Ullah. Combining word embedding interactions and letor feature evidences for supervised qpp. *QPP++@ECIR*, 2023.

J. C. De Winter, S. D. Gosling, and J. Potter. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods*, 21(3):273, 2016.

Y. Deldjoo, J. R. Trippas, and H. Zamani. Towards multi-modal conversational information seeking. In *Proceedings of the 44th International ACM SIGIR conference on research and development in Information Retrieval*, pages 1577–1587, 2021.

J. Devlin, M.-W. Chang, K. Lee, and K. 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 (Long and Short Papers)*, pages 4171–4186, 2019.

F. Diaz. Regularizing ad hoc retrieval scores. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 672–679, 2005.

F. Diaz. Performance prediction using spatial autocorrelation. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 583–590, 2007.

G. Faggioli, S. Marchesin, et al. What makes a query semantically hard? In *CEUR WORKSHOP PROCEEDINGS*, volume 2950, pages 61–69. CEUR-WS, 2021a.

G. Faggioli, O. Zendel, J. S. Culpepper, N. Ferro, and F. Scholer. An enhanced evaluation framework for query performance prediction. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 115–129. Springer, 2021b.

G. Faggioli, N. Ferro, C. I. Muntean, R. Perego, and N. Tonellotto. A geometric framework for query performance prediction in conversational search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1355–1365, 2023a.

G. Faggioli, T. Formal, S. Marchesin, S. Clinchant, N. Ferro, and B. Piwowarski. Query performance prediction for neural ir: Are we there yet? In *European Conference on Information Retrieval*, pages 232–248. Springer, 2023b.

Z. Field, J. Miles, and A. Field. Discovering statistics using r. *Discovering Statistics Using R*, pages 1–992, 2012.

C. Gao, W. Lei, X. He, M. de Rijke, and T.-S. Chua. Advances and challenges in conversational recommender systems: A survey. *AI open*, 2:100–126, 2021.

J. Gao, M. Galley, and L. Li. Neural approaches to conversational ai. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1371–1374, 2018.

A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 241–257. Springer, 2016.

D. Griol, J. Carbó, and J. M. Molina. An automatic dialog simulation technique to develop and evaluate interactive conversational agents. *Applied Artificial Intelligence*, 27(9):759–780, 2013.

X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. Feris. Dialog-based interactive image retrieval. In *Proc. NeurIPS*, pages 678–688, 2018.

D. Harman. Overview of the second text retrieval conference (trec-2). *Information Processing & Management*, 31(3):271–289, 1995.

D. Harman and C. Buckley. The nrrc reliable information access (ria) workshop . In *Proc. SIGIR*, 2004.

H. Hashemi, H. Zamani, and W. B. Croft. Performance prediction for non-factoid question answering. In *Proc. ICTIR*, 2019.

C. Hauff, D. Hiemstra, and F. de Jong. A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1419–1420, 2008.

A. Hauptmann, J. Magalhaes, R. G. Sousa, and J. P. Costeira. Mucai'20: 1st international workshop on multimodal conversational ai. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4767–4768, 2020.

B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *String Processing and Information Retrieval: 11th International Conference, SPIRE 2004, Padova, Italy, October 5-8, 2004. Proceedings 11*, pages 43–54. Springer, 2004.

B. He and I. Ounis. Query performance prediction. *Information Systems*, 31(7):585–594, 2006.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016.

J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, 2000.

B. Hidasi and A. Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 843–852, 2018.

B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

J. Huang, R. S. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Proceedings of the IEEE international conference on computer vision*, pages 1062–1070, 2015.

A. H. Jadidinejad, C. Macdonald, and I. Ounis. Using exploration to alleviate closed loop effects in recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2025–2028, 2020. ISBN 9781450380164.

D. Jannach and M. Jugovac. Measuring the business value of recommender systems. *ACM Transactions on Management Information Systems (TMIS)*, 10(4):1–23, 2019.

D. Jannach, A. Manzoor, W. Cai, and L. Chen. A survey on conversational recommender systems. *ACM Computing Surveys*, 54(5):1–36, 2021. ISSN 1557-7341.

B. J. Jansen, D. L. Booth, and A. Spink. Determining the informational, navigational, and transactional intent of web queries. *Information Processing & Management*, 44(3):1251–1266, 2008.

K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2011.

Y. Jin, W. Cai, L. Chen, N. N. Htun, and K. Verbert. Musicbot: Evaluating critiquing-based music recommenders with conversational interaction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 951–960, 2019.

K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information processing & management*, 36(6):809–840, 2000.

F. Jurcıcek, S. Keizer, M. Gašic, F. Mairesse, B. Thomson, K. Yu, and S. Young. Real user evaluation of spoken dialogue systems using amazon mechanical turk. In *Proceedings of INTERSPEECH*, volume 11, 2011.

W.-C. Kang and J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. In *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, 2020.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

J. Kiesel, A. Bahrami, B. Stein, A. Anand, and M. Hagen. Toward voice query clarification. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1257–1260, 2018.

J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2973–2980. IEEE, 2012.

E. Krikon, D. Carmel, and O. Kurland. Predicting the performance of passage retrieval for question answering. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Managemen*, 2012.

J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, 2001.

V. Lavrenko and W. B. Croft. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267, 2017.

W. Lei, X. He, Y. Miao, Q. Wu, R. Hong, M.-Y. Kan, and T.-S. Chua. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 304–312, 2020a.

W. Lei, G. Zhang, X. He, Y. Miao, X. Wang, L. Chen, and T.-S. Chua. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2073–2083, 2020b.

R. Li, S. Ebrahimi Kahou, H. Schulz, V. Michalski, L. Charlin, and C. Pal. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31, 2018.

X. Li, Z. C. Lipton, B. Dhingra, L. Li, J. Gao, and Y.-N. Chen. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*, 2016.

S.-C. Lin, J.-H. Yang, and J. Lin. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386*, 2020.

C. Lioma and I. Ounis. A syntactically-based query reformulation technique for information retrieval. *Information processing & management*, 44(1):143–162, 2008.

A. Lipani, B. Carterette, and E. Yilmaz. How am i doing?: Evaluating conversational search systems offline. *ACM Transactions on Information Systems (TOIS)*, 39(4):1–22, 2021.

Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016.

Z. Liu, H. Wang, Z.-Y. Niu, H. Wu, W. Che, and T. Liu. Towards conversational recommendation over multi-type dialogs. *arXiv preprint arXiv:2005.03954*, 2020.

X. Lu, S. Pramanik, R. Saha Roy, A. Abujabal, Y. Wang, and G. Weikum. Answering complex questions by joining multi-document evidence with quasi knowledge graphs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 105–114, 2019.

K. Luo, H. Yang, G. Wu, and S. Sanner. Deep critiquing for vae-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1269–1278, 2020.

W. Ma, R. Takanobu, and M. Huang. Cr-walker: Tree-structured graph reasoning and dialog acts for conversational recommendation. *arXiv preprint arXiv:2010.10333*, 2020.

S. MacAvaney and L. Soldaini. One-shot labeling for automatic relevance estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2230–2235, 2023.

C. Macdonald, N. Tonellotto, S. MacAvaney, and I. Ounis. Pyterrier: Declarative experimentation in python from bm25 to dense retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4526–4533, 2021.

H. Madsen and P. Thyregod. *Introduction to general and generalized linear models*. CRC Press, 2010.

C. D. Manning. *An introduction to information retrieval*. Cambridge university press, 2009.

S. E. Maxwell, H. D. Delaney, and K. Kelley. *Designing experiments and analyzing data: A model comparison perspective*. Routledge, 2017.

K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. On the dynamic generation of compound critiques in conversational recommender systems. In *Proc. AH*, pages 176–184, 2004.

R. Mehta and K. Rana. A review on matrix factorization techniques in recommender systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pages 269–274. IEEE, 2017.

C. Meng, N. Arabzadeh, M. Aliannejadi, and M. de Rijke. Query performance prediction: From ad-hoc to conversational search. *arXiv preprint arXiv:2305.10923*, 2023.

M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–214, 1998.

J. Mothe and L. Tanguy. Linguistic features to predict query difficulty. In *ACM Conference on research and Development in Information Retrieval, SIGIR, Predicting query difficulty-methods and applications workshop*, pages 7–10, 2005.

J. A. Nelder and R. W. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human-generated machine reading comprehension dataset. 2016.

R. Nogueira and K. Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

R. Nogueira, Z. Jiang, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*, 2020.

R. N. Oddy. Information retrieval through man-machine dialogue. *Journal of documentation*, 33(1):1–14, 1977.

I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

P. Owoicho, I. Sekulic, M. Aliannejadi, J. Dalton, and F. Crestani. Exploiting simulated user feedback for conversational search: Ranking, rewriting, and beyond. In *Proc. SIGIR*, pages 632–642, 2023.

J. M. O'Brien. The race to create a 'smart'google. *Fortune Magazine*, 2006.

K. Pearson. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318, 1896.

J. Pérez-Iglesias and L. Araujo. Standard deviation as a query hardness estimator. In *String Processing and Information Retrieval: 17th International Symposium, SPIRE 2010, Los Cabos, Mexico, October 11-13, 2010. Proceedings 17*, pages 207–212. Springer, 2010.

E. Poesina, R. T. Ionescu, and J. Mothe. iqpp: A benchmark for image query performance prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

F. Radlinski and N. Craswell. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*, pages 117–126, 2017.

Z. Ren, Z. Tian, D. Li, P. Ren, L. Yang, X. Xin, H. Liang, M. de Rijke, and Z. Chen. Variational reasoning about user preferences for conversational recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–175, 2022.

F. Ricci, L. Rokach, and B. Shapira. Recommender systems: introduction and challenges. In *Recommender systems handbook*, pages 1–34. 2015.

S. Y. Rieh et al. Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Information Processing & Management*, 42(3):751–768, 2006.

S. Robertson. On the optimisation of evaluation metrics. In *Keynote, SIGIR 2008 workshop learning to rank for information retrieval (LR4IR)*, 2008.

S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241. Springer, 1994.

S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

H. Roitman, S. Erera, O. Sar-Shalom, and B. Weiner. Enhanced mean retrieval score estimation for query performance prediction. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 35–42, 2017a.

H. Roitman, S. Erera, and B. Weiner. Robust standard deviation estimation for query performance prediction. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 245–248, 2017b.

H. Roitman, S. Erera, and G. Feigenblat. A study of query performance prediction for answer quality determination. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 43–46, 2019.

D. Roy, D. Ganguly, M. Mitra, and G. J. Jones. Estimating gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. *Information processing & management*, 56(3):1026–1045, 2019.

Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655, 1998.

A. Saleminezhad, N. Arabzadeh, S. Beheshti, and E. Bagheri. Context-aware query term difficulty estimation for performance prediction. In *European Conference on Information Retrieval*, pages 30–39. Springer, 2024.

G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

M. Sanderson et al. Test collection based evaluation of information retrieval systems. *Foundations and Trends® in Information Retrieval*, 4(4):247–375, 2010.

J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*, pages 291–324. Springer, 2007.

J. Schatzmann and S. Young. The hidden agenda user simulation model. *IEEE transactions on audio, speech, and language processing*, 17(4):733–747, 2009.

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, 2007.

F. Scholer and S. Garcia. A case for improved evaluation of query difficulty prediction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 640–641, 2009.

F. Scholer, H. E. Williams, J. Yiannis, and J. Zobel. Compression of inverted indexes for fast query evaluation. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229, 2002.

S. Sedhain, A. K. Menon, S. Sanner, and L. Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*, pages 111–112, 2015.

I. Sekulić, M. Aliannejadi, and F. Crestani. Exploiting document-based features for clarification in conversational search. In *European Conference on Information Retrieval*, pages 413–427. Springer, 2022.

I. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

W. Shi, K. Qian, X. Wang, and Z. Yu. How to build user simulators to train rl-based dialog systems. *arXiv preprint arXiv:1909.01388*, 2019.

B. Shneiderman, D. Byrd, and W. B. Croft. Clarifying search. *D-lib magazine*, 3(1), 1997.

A. Shtok, O. Kurland, and D. Carmel. Predicting query performance by query-drift estimation. In *Conference on the Theory of Information Retrieval*, pages 305–312. Springer, 2009.

A. Shtok, O. Kurland, and D. Carmel. Using statistical decision theory and relevance models for query-performance prediction. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 259–266, 2010.

A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems (TOIS)*, 30(2):1–35, 2012.

A. Shtok, O. Kurland, and D. Carmel. Query performance prediction using reference lists. *ACM Transactions on Information Systems (TOIS)*, 34(4):1–34, 2016.

J. D. Singer and J. B. Willett. *Applied longitudinal data analysis: Modeling change and event occurrence*. Oxford University Press, 2003.

A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management*, pages 553–562, 2015.

K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471, 1987.

X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*, 2018.

F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

W. Sun, S. Zhang, K. Balog, Z. Ren, P. Ren, Z. Chen, and M. de Rijke. Simulating user satisfaction for the evaluation of task-oriented dialogue systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2499–2506, 2021.

W. Sun, S. Guo, S. Zhang, P. Ren, Z. Chen, M. de Rijke, and Z. Ren. Metaphorical user simulators for evaluating task-oriented dialogue systems. *ACM Transactions on Information Systems*, 2023.

Y. Sun and Y. Zhang. Conversational recommender system. In *Proceedings of the 41st international ACM SIGIR Conference on Research & Revelopment in Information Retrieval*, pages 235–244, 2018.

Y. Tao and S. Wu. Query performance prediction by considering score magnitude and variance together. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1891–1894, 2014.

W. L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

R. C. Team. R: A language and environment for statistical computing. published online 2020, 2021.

E. Temizhan, H. Mirtagioglu, M. Mendes, et al. Which correlation coefficient should be used for investigating relations between quantitative variables. *Acad. Sci. Res. J. Eng. Technol. Sci*, 85:265–277, 2022.

F. N. Tou, M. D. Williams, R. Fikes, D. A. Henderson Jr, and T. W. Malone. Rabbit: An intelligent database assistant. In *AAAI*, pages 314–318, 1982.

J. R. Trippas, D. Spina, L. Cavedon, H. Joho, and M. Sanderson. Informing the design of spoken conversational search: Perspective paper. In *Proceedings of the 2018 conference on human information interaction & retrieval*, pages 32–41, 2018.

Q. Tu, S. Gao, Y. Li, J. Cui, B. Wang, and R. Yan. Conversational recommendation via hierarchical information modeling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2201–2205, 2022.

L. H. Ungar and D. P. Foster. Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, volume 1, pages 114–129. Menlo Park, CA, 1998.

S. Vakulenko, K. Revoredo, C. Di Ciccio, and M. de Rijke. Qrfa: A data-driven model of information-seeking dialogues. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41*, pages 541–557. Springer, 2019.

S. Vakulenko, S. Longpre, Z. Tu, and R. Anantha. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 355–363, 2021.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

R. Vedantam, S. Bengio, K. Murphy, D. Parikh, and G. Chechik. Context-aware captions from context-agnostic supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 251–260, 2017.

S. Verberne, M. Sappelli, K. Järvelin, and W. Kraaij. User simulations for interactive search: Evaluating personalized query suggestion. In *Advances in Information Retrieval: 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29-April 2, 2015. Proceedings 37*, pages 678–690. Springer, 2015.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

M. Vlachou and C. Macdonald. Performance predictors for conversational fashion recommendation. In *4th Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) Workshop @ RecSys 2022, September 18–23 2022, Seattle, WA, USA*, 2022.

M. Vlachou and C. Macdonald. Coherence-based query performance measures for dense retrieval. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 15–24, 2024a.

M. Vlachou and C. Macdonald. What else would i like? a user simulator using alternatives for improved evaluation of fashion conversational recommendation systems. *arXiv preprint arXiv:2401.05783*, 2024b.

E. M. Voorhees et al. The trec-8 question answering track report. In *TREC*, volume 99, pages 77–82, 1999.

E. M. Voorhees et al. Overview of the trec 2003 robust retrieval track. In *TREC*, pages 69–77, 2003.

W. Wang and I. Benbasat. Research note—a contingency approach to investigating the effects of user-system interaction modes of online decision aids. *Information Systems Research*, 24 (3):861–876, 2013.

X. Wang, C. Macdonald, and I. Ounis. Deep reinforced query reformulation for information retrieval. *arXiv preprint arXiv:2007.07987*, 2020.

X. Wang, C. Macdonald, N. Tonellotto, and I. Ounis. Pseudo-relevance feedback for multiple representation dense retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 297–306, 2021.

X. Wang, C. Macdonald, N. Tonellotto, and I. Ounis. Colbert-prf: Semantic pseudo-relevance feedback for dense passage and document retrieval. *ACM Transactions on the Web*, 17(1): 1–39, 2023.

P. Wärnestål. User evaluation of a conversational recommender system. In *Proceedings of the 4th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2005.

W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38, 2010.

H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris. Fashion iq: A new dataset towards retrieving images by natural language feedback, 2020.

H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris. Fashion iq: A new dataset towards retrieving images by natural language feedback. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11307–11317, 2021a.

Y. Wu, C. Macdonald, and I. Ounis. Partially observable reinforcement learning for dialog-based interactive recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 241–251, 2021b.

Y. Wu, C. Macdonald, and I. Ounis. Goal-oriented multi-modal interactive recommendation with verbal and non-verbal relevance feedback. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 362–373, 2023.

L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. Bennett, J. Ahmed, and A. Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

A. Yu and K. Grauman. Fine-grained comparisons with attributes. In *Visual Attributes*, pages 119–154. Springer, 2017.

H. Yu, C. Xiong, and J. Callan. Improving query representations for dense retrieval with pseudo relevance feedback. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3592–3596, 2021.

T. Yu, Y. Shen, and H. Jin. A visual dialog augmented interactive recommender system. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 157–165, 2019.

Y. Yuan and W. Lam. Conversational fashion image retrieval via multiturn natural language feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 839–848, 2021. ISBN 9781450380379.

H. Zamani, W. B. Croft, and J. S. Culpepper. Neural query performance prediction using weak supervision from multiple signals. In *Proc. SIGIR*, 2018.

H. Zamani, S. Dumais, N. Craswell, P. Bennett, and G. Lueck. Generating clarifying questions for information retrieval. In *Proceedings of the Web conference 2020*, pages 418–428, 2020.

H. Zamani, J. R. Trippas, J. Dalton, and F. Radlinski. Conversational information seeking. *arXiv preprint arXiv:2201.08808*, 2022.

H. Zamani, J. R. Trippas, J. Dalton, F. Radlinski, et al. Conversational information seeking. *Foundations and Trends® in Information Retrieval*, 17(3-4):244–456, 2023.

E. Zangerle and C. Bauer. Evaluating recommender systems: survey and framework. *ACM Computing Surveys*, 55(8):1–38, 2022.

S. Zhang and K. Balog. Evaluating conversational recommender systems via user simulation. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, pages 1512–1520, 2020.

S. Zhang, M.-C. Wang, and K. Balog. Analyzing and simulating user utterance reformulation in conversational recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 133–143, 2022.

Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. B. 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, 2018.

Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Advances in Information Retrieval: 30th European*

*Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings 30*, pages 52–64. Springer, 2008.

K. Zhou, Y. Zhou, W. X. Zhao, X. Wang, and J.-R. Wen. Towards topic-guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4128–4139, 2020.

Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 543–550, 2007.

J. Zou and E. Kanoulas. Learning to ask: Question-based sequential bayesian product search. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 369–378, 2019.

J. Zou, Y. Chen, and E. Kanoulas. Towards question-based recommender systems. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 881–890, 2020.