



HAL
open science

Proactive models for open-domain conversational search

Pierre Erbacher

► **To cite this version:**

Pierre Erbacher. Proactive models for open-domain conversational search. Information Retrieval [cs.IR]. Sorbonne Université, 2024. English. NNT : 2024SORUS009 . tel-04562712

HAL Id: tel-04562712

<https://theses.hal.science/tel-04562712>

Submitted on 29 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

Spécialité **Informatique**

École Doctorale Informatique, Télécommunications et Électronique (Paris)

Proactive Models for Open-domain Conversational Search

Modèles Proactifs pour la Recherche d'Information Conversationnelle

Présentée par

Pierre ERBACHER

Dirigée par

Laure SOULIER

Pour obtenir le grade de

DOCTEUR de SORBONNE UNIVERSITÉ

Présentée et soutenue publiquement le 31 janvier 2024

Devant le jury composé de :

Lynda TAMINE-LECHANI

Professor, IRIT, Université Toulouse, Toulouse

Rapportrice

Alexandre ALLAUZEN

Professor, ESPCI/LAMSADE, Paris

Rapporteur

Sophie ROSSET

CNRS Research Director, LISN, Université Paris-Saclay, Paris

Examinatrice

Mohammad ALIANNEJADI

Assistant Professor, University of Amsterdam, Amsterdam

Examineur

François YVON

CNRS Research Director, ISIR, Sorbonne Université, Paris

Examineur

Laure SOULIER

Associate Professor HDR, ISIR, Sorbonne Université, Paris

Directrice de thèse

CONTENTS

CONTENTS	iii
LIST OF FIGURES	vii
LIST OF TABLES	ix
ABSTRACT	xi
RÉSUMÉ	xiii
REMERCIEMENTS	xv
ACRONYMS	xvii

I	CONTEXT AND MOTIVATIONS	1
1	INTRODUCTION	3
1.1	Context	3
1.2	Motivations	6
1.3	Contributions	7
1.4	Related publications	10
2	LANGUAGE MODELS	13
2.1	Historical Overview	14
2.2	Architecture Overview	16
2.3	Training	22
2.4	LLM Zoo	27
2.5	Emergent proprieties	28
2.6	Limitations of LLMs	29
2.7	Summary	33
3	TOWARD CONVERSATIONAL SEARCH SYSTEMS	35
3.1	Information Retrieval	36
3.2	Conversational search systems	41

II MULTI-TURN INTERACTIVE INFORMATION RETRIEVAL 49

4	MULTI-TURN INTERACTIVE MODEL WITH USER SIMULATION FOR QUERY REFINEMENT	51
4.1	Introduction	52
4.2	Question Clarification Simulation Framework	53
4.3	Evaluation Protocol	57
4.4	Results	59

4.5	Conclusion and perspectives	62
5	OPTIMIZING GENERATION FOR MULTI-TURN QUERY CLARIFICATION WITH REINFORCEMENT LEARNING	63
5.1	Introduction	64
5.2	A generative model for multi-turn query clarification with reinforcement learning	65
5.3	Evaluation protocol	69
5.4	Results	72
5.5	Conclusion and perspectives	75
III	DATASET AUGMENTATION FOR MIXED-INITIATIVE CONVERSATIONAL SEARCH	77
6	AUGMENTING AD-HOC IR DATASET FOR MIXED-INITIATIVE CONVERSATIONAL SEARCH	79
6.1	Introduction	80
6.2	Simulated interactions	81
6.3	Assessing the Quality of the Dataset Generation Methodology . . .	88
6.4	Evaluation on IR Task	93
6.5	Complementary Experiments	96
6.6	Conclusion and discussion	101
7	PAQA: TOWARD PROACTIVE OPEN-RETRIEVAL QUESTION ANSWERING	103
7.1	Introduction	104
7.2	Task Definition	105
7.3	PAQA dataset	105
7.4	Results	111
7.5	Conclusion and Discussion	114
IV	PROACTIVE INTERACTIONS WITH SEARCH ENGINE	117
8	NAVIGATING UNCERTAINTY: OPTIMIZING API DEPENDENCY FOR HALLUCINATION REDUCTION IN CLOSED-BOOK QA	119
8.1	Introduction	120
8.2	Learning when to search with LLMs	122
8.3	Evaluation protocol	123
8.4	Results	127
8.5	Conclusion	128
V	CONCLUSION AND PERSPECTIVE	129
9	CONCLUSION	131

9.1 Summary of Contributions 132

9.2 Promising Perspectives and Future work 133

9.3 Open discussion about LLMs and IR 136

BIBLIOGRAPHY 139

LIST OF FIGURES

CHAPTER 1: INTRODUCTION	3
CHAPTER 2: LANGUAGE MODELS	14
Figure 2.1 Transformer Architecture	17
Figure 2.2 Multi-Head Attention	19
Figure 2.3 Positional Encoding	21
Figure 2.4 Encoder-Decoder, Decoder Only, Encoder only	22
Figure 2.5 Pre-Training Strategies	24
Figure 2.6 LLMs Zoo	28
Figure 2.7 Emerging capabilities Examples	29
Figure 2.8 Performance against LLMs Scale	30
CHAPTER 3: TOWARD CONVERSATIONAL SEARCH SYSTEMS	36
Figure 3.1 Diagram of Information Retrieval System	37
Figure 3.2 Dense Retrieval/Ranking Models	39
Figure 3.3 Dagstuhl Conversational Search Typology	41
Figure 3.4 Example of conversational exploratory search	43
Figure 3.5 Interactivity Spectrum	44
CHAPTER 4: MULTI-TURN INTERACTIVE MODEL WITH USER SIMULATION FOR QUERY REFINEMENT	52
Figure 4.1 Query clarification simulation framework	54
Figure 4.2 Effectiveness score of query reformulation by rank	59
Figure 4.3 Maximal Score with increasing number of queries in the set	60
CHAPTER 5: OPTIMIZING GENERATION FOR MULTI-TURN QUERY CLARIFICATION WITH REINFORCEMENT LEARNING	64
Figure 5.1 Multi-turn Query clarification framework	65
Figure 5.2 Fine-tuning GPT2 with Proximal Policy Optimization (PPO)	69
Figure 5.3 Google and CIRCLE coop/turns HeatMap	75
CHAPTER 6: AUGMENTING AD-HOC IR DATASET FOR MIXED- INITIATIVE CONVERSATIONAL SEARCH	80
Figure 6.1 Online evaluation pipeline	87
Figure 6.2 Score Distribution over ranking	97

Figure 6.3	Passages ranking similarity	101
CHAPTER 7: PAQA: TOWARD PROACTIVE OPEN-RETRIEVAL		
QUESTION ANSWERING		104
Figure 7.1	Proactive question-answering task	105
Figure 7.2	PAQA dataset	107
Figure 7.3	Evolution of reconstruction metrics against threshold . . .	113
CHAPTER 8: NAVIGATING UNCERTAINTY: OPTIMIZING API		
DEPENDENCY FOR HALLUCINATION REDUCTION IN CLOSED-BOOK QA		120
Figure 8.1	Task Description	121
CHAPTER 9: CONCLUSION		
Figure 9.1	Meme of the current NLP research	134

LIST OF TABLES

CHAPTER 2: LANGUAGE MODELS	14
Table 2.1 List of emergent abilities of LLMs	31
CHAPTER 3: TOWARD CONVERSATIONAL SEARCH SYSTEMS	36
Table 3.1 Taxonomy of clarifying questions	45
Table 3.2 Overview of Conversational Search Datasets	47
CHAPTER 4: MULTI-TURN INTERACTIVE MODEL WITH USER SIMULATION FOR QUERY REFINEMENT	52
Table 4.1 Results Multi-turn Test set MS MARCO	61
Table 4.2 Results Multi-turn Dev set MS MARCO	61
CHAPTER 5: OPTIMIZING GENERATION FOR MULTI-TURN QUERY CLARIFICATION WITH REINFORCEMENT LEARNING	64
Table 5.1 Effectiveness of CIRCLE methods	73
Table 5.2 Qualitative Examples against Google suggestions	73
Table 5.3 Ranking Diversity between turns (RBO)	74
CHAPTER 6: AUGMENTING AD-HOC IR DATASET FOR MIXED- INITIATIVE CONVERSATIONAL SEARCH	80
Table 6.1 Example of simulated interactions of MiMarco	83
Table 6.2 Statistics of MiMarco	89
Table 6.3 Evaluation of different clarifying models	92
Table 6.4 Clarifying Question/Passages Cosine Similarity	92
Table 6.5 Results of the human evaluation	93
Table 6.6 IR effectiveness on the MiMarco	95
Table 6.7 Examples of successful cases	98
Table 6.8 Examples of failure cases	99
Table 6.9 Effectiveness results on T interaction	99
Table 6.10 IR effectiveness on NQ	100
CHAPTER 7: PAQA: TOWARD PROACTIVE OPEN-RETRIEVAL QUESTION ANSWERING	104
Table 7.1 PAQA Statistics	108
Table 7.2 Examples clarifying questions	108

Table 7.3	Clarifying questions generation performances	110
Table 7.4	Ambiguous questions classification performances	111
Table 7.5	Naturalness metric description	111
Table 7.6	Relevance metric description	111
Table 7.7	Examples from PAQA dataset	114
Table 7.8	Qualitative examples of generated cq	115

**CHAPTER 8: NAVIGATING UNCERTAINTY: OPTIMIZING API
DEPENDENCY FOR HALLUCINATION REDUCTION IN CLOSED-
BOOK QA** 120

Table 8.1	Metric description	124
Table 8.2	Distribution of exact match and hallucinations	127
Table 8.3	Fluctuation of correct answers	127

CHAPTER 9: CONCLUSION 132

ABSTRACT

Access to information is a fundamental need for individuals, businesses, and governments in order to stay informed, share information, and make decisions. Thanks to various advancements in Deep Learning and NLP, conversational systems are increasingly becoming important gateways to information across a wide range of application areas, such as web search, especially since OpenAI’s demonstration to a broad audience. However, state-of-the-art conversational models cannot yet be considered as Conversational Search Systems (CSS). Beyond providing natural language responses, a key capability of CSS is their proactive participation in conversations with users. This enables CSS to better understand users’ needs and assist them during search sessions, particularly for exploratory searches or voice-only interactions. Understanding users’ needs remains a challenge, interacting with users is costly, and expanding supervised data sets remains a challenge. Moreover, these models are supposed to not only initiate conversations with users but also proactively interact with a diverse range of other systems or databases. This includes various tools (calendar, calculator), the internet (search engines), and various other APIs (weather, maps, e-commerce, bookings, etc.). For example, responding to complex user queries that require information from multiple sources. In this thesis, we are interested in system-initiated interactions that help understand users’ needs and system-to-system integrations. In the first part, we use user simulation methods in open-domain information search that mimic users’ responses to enable multi-turn system-user interactions at a minimal cost. In the second part, we propose a methodology to augment existing IR datasets with synthetic clarifying questions and user responses while maintaining document relevance judgments. We also propose an offline and online evaluation framework with user simulation. In the last part, we investigate hallucination detection in order to optimize the reliance on external search engines of tool-augmented language models.

RÉSUMÉ

L'accès à l'information est un besoin fondamental pour les individus, les entreprises et les gouvernements afin de rester informés, de partager des informations et de prendre des décisions. Grâce aux différentes avancées en Deep Learning et en NLP, les systèmes conversationnels deviennent de plus en plus des passerelles importantes vers l'information dans une majorité de domaines d'application, tels que la recherche sur le web, en particulier depuis la démonstration d'OpenAI à un large public. Cependant, ces modèles conversationnels ne peuvent pas encore être considérés comme des systèmes de recherche conversationnelle (CSS) à part entière. Au-delà de fournir des réponses en langage naturel, une capacité clé des CSS est leur participation proactive dans les conversations avec les utilisateurs. Cela permet aux CSS de mieux saisir les besoins des utilisateurs, de les assister et de les aider à naviguer virtuellement pendant les sessions de recherche, en particulier pour la recherche exploratoire ou avec des interactions uniquement orales. Interagir avec les utilisateurs est coûteux, et l'élargissement des jeux de données supervisés reste un défi dans le cadre de la recherche d'information conversationnelle. De plus, ces modèles sont censés non seulement être à l'initiative dans les conversations avec les utilisateurs, mais aussi interagir de manière proactive avec une gamme diversifiée d'autres systèmes ou bases de données. Cela inclut divers outils, moteurs de recherche et diverses autres API (météo, cartes, e-commerce, réservations, etc.). Par exemple, répondre à des requêtes utilisateur complexes nécessitant des informations provenant de multiples sources.

Dans cette thèse, nous nous intéressons aux interactions initiées par le système qui aident à comprendre les besoins des utilisateurs, ainsi qu'aux interactions système-système. Dans la première partie, nous exploitons des méthodes de simulation d'utilisateurs en recherche d'informations en domaine ouvert qui imitent les réponses des utilisateurs pour permettre des interactions système-utilisateur à plusieurs tours à moindre coût. Dans la deuxième partie, nous proposons une méthodologie pour augmenter les jeux de données IR existants avec des questions de clarification synthétiques et des réponses d'utilisateurs tout en maintenant les jugements de pertinence des documents. Nous proposons également un cadre d'évaluation en ligne et hors ligne avec simulation d'utilisateur. Dans la dernière partie, nous étudions la détection d'hallucinations dans les modèles de langue afin d'optimiser leur dépendance aux moteurs de recherche externes.

REMERCIEMENTS

Je souhaite exprimer ma profonde reconnaissance envers tous ceux qui ont joué un rôle essentiel dans l'accomplissement de cette thèse. En premier lieu, je tiens à exprimer ma gratitude envers ma directrice de thèse, Laure Soulier. Sa supervision attentive durant ces trois années, ses conseils avisés, sa disponibilité et ses encouragements ont été inestimables pour ce projet. Je remercie chaleureusement les membres de mon jury, pour avoir consacré du temps à l'examen de mon manuscrit et pour leurs retours constructifs. Je tiens également à remercier mes collègues de l'équipe MLIA pour les échanges stimulants, le partage de connaissances et les moments agréables passés ensemble au cours de ces trois années. Un remerciement particulier à Louis S, Louis Fa, Louis Fo, Lise, Léon, Etienne, Lucas, Folco, Téo, Marie, Guillaume, Paul, Alexandre, Yuan, Tanguy, Raphael, Florian, Adel, et Christophe. Un merci spécial à mes proches pour leur soutien, leur écoute et leurs encouragements constants. Je suis particulièrement reconnaissant envers Thibaut C, un ami infailible et à l'écoute depuis 24 ans, ainsi que Thibault H, Thibault R, Vincent et Antoine pour les aventures humaines. Et spécialement, Océane, pour son accompagnement, son soutien, sa joie et son amour. À ma famille, ma sœur Laura, mes frères Julien et Arthur, ainsi qu'à mes parents, je suis reconnaissant pour leur éducation, leur amour et leur soutien durant ces 29 années. Pour finir, je remercie toutes les personnes qui ont, de près ou de loin, contribué à cette thèse et m'ont soutenu au fil des ans. Ce travail est le fruit de leur soutien et de leur contribution.

ACRONYMS

NLP	Natural Language Processing
RL	Reinforcement Learning
PPO	Proximal Policy Optimization
MLM	Masked Language Model
PLM	Pretrained Language Model
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
IR	Information Retrieval
CS	Conversational Search
LLM	Large Language Model
LM	Language Model
QA	Question Answering
RLHF	Reinforcement Learning With Human Feedback
US	User Simulation
GPU	Graphics Processing Units
RNN	Recurrent Neural Network
TPU	Tensor Processing Units
FNN	Feed-Forward Neural Network
SERP	Search Engine Result Page

Part I

CONTEXT AND MOTIVATIONS

INTRODUCTION

"Information is the resolution of Uncertainty"

Claude Shannon - 1948

1.1 Context

The invention of reading and writing (between 3400 and 3100 BC) has given language a new power. A capacity to persist and be preserved in the form of a trace on a durable material support. This memory medium allowed language and information to travel through space and time. This has enabled mankind to preserve, organize, and share knowledge such as commercial, financial, and legal records as well as art, literature, techniques, and sciences. Several technical innovations (such as libraries, printing, and search engines) facilitate the sharing and accessibility of knowledge. With the significant increase in global literacy rates and internet accessibility, the number of people using the internet has surged over the past decade. The global internet user population grew from approximately 2.6 billion in 2013 to over 5 billion by 2023¹. This expansion of internet access has led to a dramatic increase in online activity, including web searches. In 2023, around 9.0 billion web searches were conducted daily, highlighting the integral role of search engines in the digital age². Access to digital information has become a fundamental right and a critical component of modern society for social, political, and economic sustainability.

The ability to access, analyze, and use information effectively has become essential for individuals and organizations to stay informed, share knowledge, make decisions, and solve problems. Retrieving, filtering, and analyzing effectively the ever-growing volume of data pose continuous challenges; particularly as the

1. <https://ourworldindata.org/internet>

2. <https://www.oberlo.com/blog/google-search-statistics>

amount of data in the world is doubling every two years (H. Guo et al. 2014). In the field of computer science and information science, the Information Retrieval (IR) community perpetually designs more effective systems enabling users to automatically search through vast amounts of data and access information that aligns with their specific needs. The recent advances in Natural Language Processing (NLP) and IR have significantly improved the performance of retrieval systems, inherently improving users' experience by reducing time, cognitive load, and frustration caused by irrelevant information (Barifah and Landoni 2020). However, the search paradigm has not evolved significantly, with user-system interactions consisting mainly of a few keywords entered into search bars, with results presented as a ranked list of documents on Search Engine Results Pages (SERPs). In this framework, accurately understanding the user's intent remains a challenge, especially when the information need is complex or the resulting query is vague (Culpepper et al. 2018). Understanding users' information needs is a long-standing challenge in IR (Cronen-Townsend and Croft 2002; Jansen et al. 2000; Sanderson 2008), often highlighted by the difficulty for users to formulate open-ended information needs into specific queries (Jansen et al. 2000). Users' queries can be influenced by context, cultural background, knowledge/expertise, and cognitive states (e.g. Tip of the Tongue Phenomenon (Arguello et al. 2021; Schwartz and Metcalfe 2011)), often leading to under-specified queries formulated with poor word choice (Bhavnani 2001; R. W. White et al. 2008; Wildemuth 2004; Kamvar and Baluja 2006). In addition to suboptimal results, users may also be subject to confirmation bias (R. White 2013) whereby their pre-existing personal beliefs are reinforced due to biased results stemming from skewed wording or subjective stances. On the other hand natural conversation is the most natural form of communication between humans and is also the oldest form of interaction for IR tasks since the invention of the Library, where seekers expressed their information needs through a conversation with the librarians. Conversational Search (CS) is regarded as one of the next research frontiers allowing more complex information needs to be expressed and refined through multiple sequential interactions and therefore more efficient search sessions

Conversational systems are increasingly becoming important gateways to information in a wide range of application domains such as customer service, health, education, office work, online shopping, and web search. Their adoption is being driven by social, financial, and technological factors. Advances in speech recognition and audio synthesis have enabled the democratization of virtual personal assistants and voice-only devices, with some 4.2 billion digital voice assistants

in use worldwide, such as Amazon’s Alexa, Google’s Assistant, Apple’s Siri³, encouraging the adoption and use of these systems. Recently, Large Language Model (LLM)s have attracted significant interest from both the general public and businesses, especially since OpenAI⁴ demonstrated the advanced capabilities of LLMs to a wide audience with ChatGPT⁵. LLMs have shown remarkable performance in both natural language understanding and natural language generation. When properly trained, LLMs are able to generate fluent text in response to natural language questions or instructions, making them ideal for searching and summarizing documents for IR applications (Brown et al. 2020a; Thoppilan et al. 2022a; Shuster et al. 2022a). This leads to large investments and therefore more resources required to scale, train, and deploy such models. The rapid and unprecedented massive adoption of these tools⁶ has forced major players to accelerate the development and deployment of their conversational large-scale Language Model (LM)s. Search engine companies such as Google⁷, Microsoft⁸, Baidu⁹ or Naver¹⁰ are competing to integrate and improve conversational LMs into their search engines (BARD/Gemini, NewBing, Ernie, CLOVA respectively). While these models are able to follow long conversations, answer questions, and summarize documents with impressive fluency, they cannot be considered as true CS systems (Radlinski and Craswell 2017; Zamani et al. 2023).

A CS system is a IR system that exploits the conversation to retrieve documents. Beyond providing natural language answers, a key capability of CS systems is their (pro)active participation in the conversation with users. This allows CS systems to better capture users’ needs but also guide, and assist them during search sessions (Zamani et al. 2023). In particular, when users cannot browse the list of documents to assess the relevance, as in pure speech interactions, the system needs to take the initiative to ask for additional context, ask for confirmation, or suggest more information to help the user navigate virtually and reduce his cognitive load (Shah and Bender 2022; Jeffrey Dalton et al. 2022; Zamani et al. 2022; Anand et al. 2020; Bender et al. 2021; Radlinski and Craswell 2017; Johanne R. Trippas et al. 2020; Aliannejadi et al. 2019; Keyvan and Huang 2022; Zamani et al. 2020c). Additionally, these models are expected not only to take the initiative in conversation with users but also to proactively interact with a diverse range of

3. <https://www.forbes.com/sites/joanverdon/2021/04/27/global-ecommerce-sales-to-hit-42-trillion->

4. <https://openai.com/>

5. <https://openai.com/blog/chatgpt>

6. <https://explodingtopics.com/blog/chatgpt-users>

7. <https://www.google.com>

8. <https://www.microsoft.com/>

9. <https://www.baidu.com/>

10. <https://www.naver.com/>

other systems or databases, including various tools (calendar, calculator), internet (search engines), and various other APIs (weather, maps, e-commerce, booking..). Large, diverse, and high-quality data remains the cornerstone of LLMs and other Deep Learning (DL) based methods. Ideally, CS should be trained and evaluated on datasets containing diverse examples of search-oriented and mixed-initiative conversations aligned with annotated relevant documents from open-domain collections. However, due to the high cost of collecting and annotating such data, available conversational datasets for information access are typically small, hand-crafted, and limited to domain-specific applications such as recommendation or conversational question-answering, which are typically user-initiated and contain simple or a series of contextualized questions. In addition, it is particularly challenging to properly evaluate CIS systems because of the nature of the interactions. The effectiveness or relevance of clarification questions also depends on the user's answers, resulting in high variability and trajectories. User Simulation (US) has been used extensively to train and evaluate conversational recommender systems for structured Knowledge Bases. In the open domain, IR, the use of user simulation is still limited and underused.

1.2 Motivations

With this context in mind, we tackle in the manuscript the following challenges that remain open.

Evaluating Multi-turn Interactive Search Systems with User Simulation. The benefit of interactive IR systems is that they actively involve both users and systems in the search process. Systems can probe users through multiple modalities (click, text ..) to better model their information need. Resulting in additional context that can be used to refine initial searches. Yet, training and evaluating systems with real user interactions has multiple shortcomings. Real-time user interaction is resource-intensive, burdensome, and runs the risk of compromising the user experience, particularly during the system's training phase. The challenge then arises: how can we leverage user simulations in open domain search? More specifically, how can simulations mimic real user behavior to both train and evaluate multi-turn interactive IR systems?

Mixed-initiative Conversational Search. CS systems should be able to take initiative in the conversation to ask clarifying questions. The immediate challenge

lies in the realm of data: how can we curate a dataset that's both extensive and relevant to train and evaluate these systems? Furthermore, how can the mixed-initiative serve document retrieval? And importantly, how can we generate data that teaches systems to discern ambiguity in queries and encourage pro-activity to clarify them?

Interactions with search engine CS systems are increasingly envisioned to not only interface with humans but to interact seamlessly with other machines and external tools, be it calculators, databases, or search engines. Because LLMs are pre-trained on large collections of documents, they are able to answer factual questions with a certain degree of correctness. Querying a search engine induces additional computational costs. We tackle the challenge of reducing the dependency on search engines while maintaining a high level of correct answers. How can we teach LLMs to self-evaluate their confidence in their ability to directly answer without the need for an external search engine?

These research directions lead to multiple contributions that are introduced and summarized in the following.

1.3 Contributions

1.3.1 Multi-turn query clarification

Query suggestion, an established technique in IR, assists users in refining their searches by offering a set of alternative, more comprehensive and precise queries, thereby facilitating faster access to relevant information (W. Chen et al. 2017; Aliannejadi et al. 2019; Zamani et al. 2020a; B. Wu et al. 2018; J. Guo et al. 2011; Santos et al. 2012). This method often draws from query logs of user search sessions (Zamani et al. 2020a; B. Wu et al. 2018; J. Guo et al. 2011; Santos et al. 2012; Ozertem et al. 2012; Zamani et al. 2020b; Jiafeng Guo et al. 2011; Garigliotti and Balog 2017), optimizing user experience by minimizing typing and discovery time. However, this approach poses challenges for smaller search engines and academic researchers due to the proprietary nature of query logs. Furthermore, current methods, limited to single-turn interactions, overlook the complexities of multi-turn settings where ongoing user-system interactions (trajectories) significantly influence query suggestions. Gathering real user trajectories for training

and evaluation is resource-intensive. A feasible alternative is user simulations, prevalent in recommendations but less so in open-domain IR due to the need for annotated intentions or document features. This thesis delves into a simulated multi-turn query clarification system, wherein an IR agent refines user queries through sequential interactions, as outlined in (Aliannejadi et al. 2019). Our focus is on simple information needs, deferring multi-faceted needs for later, as they may impact query ranking. Our framework is tested on the MS MARCO dataset, demonstrating the advantages of multi-turn interactions and various question selection methods on the IR effectiveness. Additionally, we enhance the multi-turn model by generating diverse query suggestions and employing reinforcement learning for improved query set creation. These findings are detailed in Chapter 4 and Chapter 5.

1.3.2 Dataset Augmentation for mixed-initiative conversational search

Conversational systems have made significant progress in generating natural language responses. However, their potential as CS systems is currently limited due to their passive role in the information-seeking process. Such systems should be able to interact with users to better understand their needs, by asking clarifying questions and then retrieving more relevant documents. However, asking clarifying questions in open domain search (Aliannejadi et al. 2019; Zamani et al. 2020a) remains a challenge as well as utilizing these interactions to better rank documents. Evaluating the retrieval capabilities of those systems at a large scale on IR task is very challenging, requiring adequate datasets containing such interactions to be aligned with relevant documents. However, collecting such large-scale conversational data is challenging because of the high annotation cost of user-systems interactions as well as annotating relevant documents in a collection.

With these limitations in mind, in Chapter 6 and Chapter 7 we explore how to augment existing datasets to train and evaluate CS systems.

In the first work, we propose a methodology to extend existing ad-hoc IR datasets (containing query/documents relevance pairs) to integrate mixed-initiative interactions, allowing us to train and evaluate retrieval models in the context of mixed-initiative conversations. We generate synthetic interactions using a clarification question generator and a user simulation mimicking the user’s response. We use them on the MS MARCO ad-hoc IR dataset and evaluate the augmentations using human raters and an IR downstream task.

In a second work, we tackle the challenge of generating relevant clarifying questions by taking into account the inherent ambiguities present in both user’s queries and documents. To achieve this, we propose PAQA, an extension to the existing AmbiNQ dataset, incorporating clarifying questions. This dataset allows training and evaluating pro-active question-answering systems that either directly answer non-ambiguous questions or ask clarifying questions when questions are ambiguous. We evaluated the dataset using human evaluators and trained various models for the pro-active QA task. We additionally assess how retrieval-augmented methods handle ambiguity detection and impact the quality of clarifying questions.

By addressing these gaps in CS systems, we aim to provide additional supervision to enhance the system’s active participation in the information-seeking process and provide more effective results.

1.3.3 Proactive Interactions with Search Engine

While other parts of the thesis focus on the system taking initiative with the user as a means of clarifying the query, the system may also take initiative with other systems to solve a given task. Especially when the system can access various specialized APIs such as calculator, calendar, and search engine but may induce additional computational cost. In the context of question answering, the literature usually heavily relies on LM coupled with external knowledge bases (Karpukhin et al. 2020; Lewis et al. 2020a; Borgeaud et al. 2022a; Schick et al. 2023; Shuster et al. 2022b; Nakano et al. 2021) where the LM extracts and generates an answer conditioned on retrieved documents. This method improves truthfulness and reduces the hallucinations. LLMs have accumulated a lot of knowledge during pre-training and are usually able to answer questions with parametric knowledge. State-of-the-art models do not take into account the additional cost of retrieving and processing documents. Proposed models tend to always rely on external databases for each question, without considering the model’s inherent ability to answer the question, with no real distinction between directly answerable questions and the real need for external knowledge. In this thesis, we investigate a more nuanced approach that uses external knowledge while also incorporating the parametric knowledge of LLMs. The aim is to teach LLMs to respond directly to the user when possible, or to invoke an external system when not. We, therefore, propose to study methods that generate either a natural language answer or an API call (e.g. $\langle search \rangle$) only when the model is uncertain about the answer,

minimizing the dependence on external resources helps to save inference time and computational cost while maintaining high performance. We focus on closed-book question-answering (CBQA) tasks, conducted on two datasets (Natural Questions (NQ) (Kwiatkowski et al. 2019b) and TriviaQA (TQA) (Joshi et al. 2017a)).

1.3.4 Outline of this Thesis

This thesis is divided into several chapters. Chapter 2 and Chapter 3 provide the context and technical background for understanding the following chapters. Chapter 2 covers elementary concepts of the principle behind LMs while Chapter 3 briefly introduce the principle of IR, focusing mainly on conversational information retrieval. Following Chapter 4, Chapter 5, Chapter 6, Chapter 7, and Chapter 8 detail the different contributions. Chapter 4 and Chapter 5 present our work on US for multi-turn query clarification in open domain IR. Chapter 6 and Chapter 7 focus on the generation of datasets for training and evaluation of mixed-initiative conversational IR systems. Chapter 8 details our study on hallucination detection for cost-effective system interaction with a search engine in question answering. The last Chapter 9 concludes with an overview, a discussion, and a perspective.

1.4 Related publications

This thesis is based on the material published in the following papers:

- Pierre Erbacher, Laure Soulier, and Ludovic Denoyer (2022b). *State of the Art of User Simulation approaches for conversational information retrieval*. arXiv: 2201.03435 [cs.IR]
- Pierre Erbacher, Ludovic Denoyer, and Laure Soulier (2022a). *Interactive Query Clarification and Refinement via User Simulation*. arXiv: 2205.15918 [cs.IR]
- Pierre Erbacher and Laure Soulier (2023). *CIRCLE: Multi-Turn Query Clarifications with Reinforcement Learning*. arXiv: 2311.02737 [cs.IR]
- Pierre Erbacher, Jian-Yun Nie, Philippe Preux, and Laure Soulier (2023b). *Augmenting Ad-Hoc IR Dataset for Interactive Conversational Search*. arXiv: 2311.06119 [cs.IR]

- Pierre Erbacher, Jian-Yun Nie, Philippe Preux, and Laure Soulier (2023c). *PAQA: Toward ProActive Open-Retrieval Question Answering*
- Pierre Erbacher, Louis Falissard, Vincent Guigue, and Laure Soulier (2023a). *Navigating Uncertainty: Optimizing API Dependency for Hallucination Reduction in Closed-Book QA*

LANGUAGE MODELS

Chapter abstract

In this chapter, we propose a general overview of the literature regarding Language Models. We first propose a deep dive into transformers based Language Model (LM)s. We then briefly present capabilities and limitations.

Contents

2.1	Historical Overview	14
2.2	Architecture Overview	16
2.2.1	Self-Attention Mechanism	18
2.2.2	Multi-Head Attention	20
2.2.3	Positional Encoding	21
2.2.4	Encoder/Decoder Cross-attention	22
2.3	Training	22
2.3.1	Pre-Training Objectives	23
2.3.2	Fine-tuning	26
2.3.3	Reinforcement learning	26
2.4	LLM Zoo	27
2.5	Emergent proprieties	28
2.6	Limitations of LLMs	29
2.7	Summary	33

2.1 Historical Overview

Language modeling is a fundamental task in the domain of Natural Language Processing (NLP) and Computational Linguistics. It involves modeling statistical patterns and hierarchical representations within languages such as the development of probabilistic models to predict the likelihood of a sequence of words (Y. Bengio et al. 2000), and is crucial for various applications including for instance machine translation and text generation. A statistical model for language predicts the probability of the next word based on the previous words, with the following joint probability of the sequence w_1, w_2, \dots, w_T in Equation 2.1

$$P(w_1, w_2, \dots, w_T) = \prod_{t=1}^T P(w_t | w_{t-1}) \quad (2.1)$$

The essence of the challenge in language modeling lies in the curse of dimensionality, especially when handling the joint distribution of discrete variables like consecutive words in a sentence. For instance, modeling 10 consecutive words from a 100,000-word vocabulary leads to an exponentially large number of potential parameters. Early attempts at language modeling rely on n-gram models (Goodman 2001), typically utilizing Hidden Markov Models with trigrams, simplify the

problem by considering only a few preceding words to estimate probabilities for the next word. Despite the practical applications of n-gram models, they have limitations in scalability and contextual understanding, leading to the emergence of more sophisticated techniques. The advent of Deep Learning (DL) has significantly advanced the field of language modeling by addressing these limitations and providing more nuanced ways to capture the complexities of language.

DL is a subfield of Machine Learning (ML), which itself is a branch of Artificial Intelligence (AI). While AI regroups all types of systems that automatically improve from observations, at the core of DL are artificial neural networks, which are composed of layers of interconnected nodes (neurons). These networks are inspired by the structure and function of the human brain, where neurons are interconnected and work together to process information and learn useful and dense representations (Mikolov et al. 2013). An early attempt at language modeling with neural networks was introduced by Y. Bengio et al. 2000 and managed to learn dense representations of words using a larger context. Further improvement with as Feed-Forward Neural Network (FNN) and Recurrent Neural Network (RNN), markedly eclipsed the performance of conventional statistical models. DL methods bring considerable enhancements in language modeling, enabling the computation of dense representations of words in latent spaces (word embeddings). Techniques such as Word2Vec (Mikolov et al. 2013) and GloVe (Pennington et al. 2014) emerged as pioneering approaches in learning dense vector representations of words. Unlike sparse, high-dimensional, and binary bag-of-words representations, these embeddings encapsulate semantic relationships between words in lower-dimensional space. Word2Vec employs shallow neural networks to learn word representations by predicting a word based on its context (Continuous Bag of Words, CBOW) or predicting context words given a target word (Skip-Gram). GloVe, on the other hand, leverages global co-occurrence statistics from a corpus to learn word vectors, capturing both local and global semantics. Traditional RNNs were hindered by the vanishing gradient problem, which made it difficult to capture long-term dependencies in sequences. The development and widespread adoption of more sophisticated RNN variants like Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) networks and Gated Recurrent Units (GRUs) (J. Chung et al. 2014) allowed language model to capture long dependency. Another milestone in the language modeling arena was achieved with the introduction of ELMo (Embeddings from Language Model (LM)) by Peters et al. 2018. ELMo addresses the limitation of traditional word embeddings, like Word2Vec and GloVe which assign a fixed vector to each word regardless of the context in which it appears. Instead, ELMo generates contextualized word embeddings, meaning that the representation of each word is influenced by its

surrounding words, thus capturing polysemy and nuanced semantic differences. ELMo utilizes a multi-layer bidirectional LSTM trained on a large text corpus to derive these contextual embeddings, significantly enhancing the model's understanding of semantic and syntactic context. The ELMo approach demonstrated substantial improvements in several NLP tasks, marking a step towards more sophisticated and context-aware representations in language modeling. The next significant leap was the introduction of the Transformer architecture by (Vaswani et al. 2017). Transformers abandon recurrence and instead utilize self-attention mechanisms to process input sequences in parallel, thereby drastically improving computational efficiency while still effectively capturing long-term dependencies in text. This architecture became the foundation for subsequent state-of-the-art LMs like BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al. 2019) and GPT (Generative Pre-trained Transformer) (Radford et al. 2018), which further pushed the boundaries of performance across a wide array of NLP tasks, including conversational agents, summarization, translation, and question-answering (Brown et al. 2020b; Bubeck et al. 2023; Wei et al. 2022a). Transformers are now ubiquitous to DL because they have demonstrated remarkable performance and scaling capabilities for various domains such as NLP (Brown et al. 2020a), Computer-Vision (Dosovitskiy et al. 2021) and optimal control (Brohan et al. 2023; L. Chen et al. 2021). The Transformer's parallel processing and self-attention mechanism make it highly efficient for training on Graphics Processing Units (GPU)s or Tensor Processing Units (TPU)s, allowing for faster training compared to traditional sequential models like RNNs. This catalyzed a scaling trend in Transformer models, both in terms of model size and data, which not only improved performance across various NLP tasks but also unveiled new capabilities (Brown et al. 2020b; OpenAI 2023).

In the following section, we delve into the fundamental concepts of transformers such as self-attention, multi-head attention, or positional encoding. Then we elaborate on the different training paradigms of these models. Finally, we address the proprieties of language models by contouring their capabilities and limitations.

2.2 Architecture Overview

The main idea behind the transformer architecture is to use self-attention to learn long-range dependencies in sequences. Self-attention is a mechanism that al-

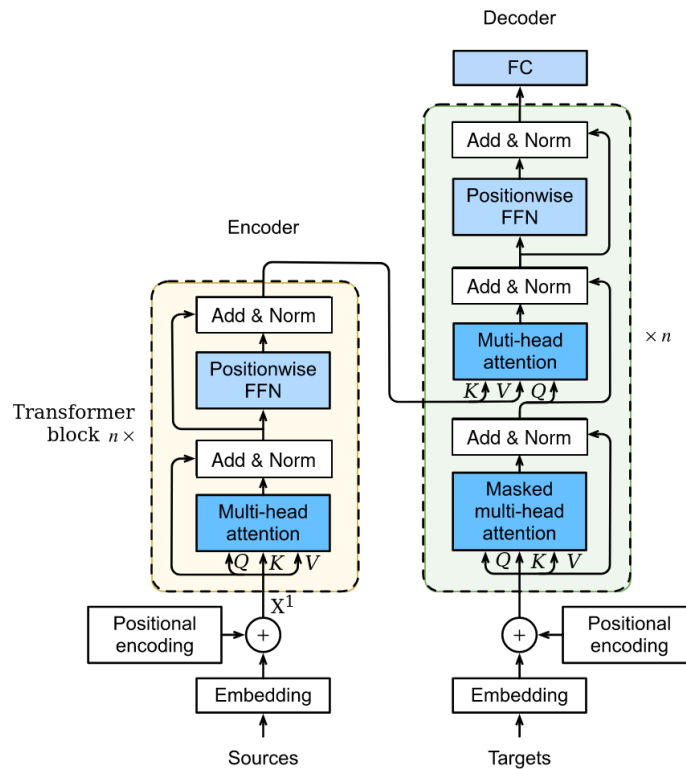


Figure 2.1 – **Transformer Architecture.** with Encoder and Decoder (source:(Vaswani et al. 2017))

allows a model to attend to different parts of the input sequence and learn how they are related to each other. This is different from RNN, which typically processes sequences one token at a time, to update hidden variables used for the next token, making it difficult to learn long-range dependencies and difficult to parallelize. Transformers only operate through matrix multiplications on the entire sequence, a highly efficient and parallelizable approach. Words are converted into tokens which are indices representing words or sub-words. Given an input sequence of N tokens, we consider the matrix $X = x_1 \dots x_N$ where each column x_N is a vector of features of dimension D . Transformers are divided into successive blocks and output a matrix X^m with the same dimensions $D \times N$. Successive transformations are performed at each layer m with $0 < m < M$ in the transformers. These transformations tend to refine representations of each token regarding the whole sequence. Several operations are performed on the matrix X at each level in the transformer. "Line-wise" operations refine features independently by considering the same feature across the whole sequence. This is called the self-attention mechanism. "Column-wise" operations refine token representations independently of other tokens in the sequence. This operation is performed with FNN. These pro-

cedures collectively constitute what is known as transformer blocks, represented as in Equation 2.2 and pictured as grey boxes in the Figure 2.1. X^m is the matrix output of the transformer block $m - 1$.

$$X^m = \text{transformerBlock}^{m-1}(X^{m-1}) \quad (2.2)$$

Furthermore, Transformer blocks use residual connections after the Multi-Head attention and the FNN as introduced in (K. He et al. 2016). Residual connections help propagate gradients more effectively through the network, allowing the gradients to flow directly to earlier layers without significant degradation (vanishing gradient). As a result, deeper networks can be trained more easily with residual connections.

$$x' = x + \text{sublayer}(x) \quad (2.3)$$

With the sublayer being multihead attention or a fully-connected feed-forward network.

A transformer architecture is a concatenation of several transformer blocks that iteratively refine token representations. Each representation x_i is used for a classification task where the last layer predicts a score for each token in the vocabulary.

In the section below, we describe the self-attention, the multi-head attention.

2.2.1 Self-Attention Mechanism

The attention mechanism was introduced by Bahdanau et al. 2014 in RNN encoder-decoder architecture as a way to compute a new context vector c_i as the weighted sum of hidden states h_i . This context vector is then used for computing the current vector $s_i = f(s_{i-1}, y_{i-1}, c_i)$ in the decoder. The goal is to align encoder representations h_i and decoder representations s_i . This can be interpreted as how important each hidden state of the encoder should be in the computation of the new context vector in the decoder. Given the input matrix X^m (where each column is a hidden state), these are weighted by an Attention matrix A of dimension $N \times N$ using matrix multiplication as shown in the Equation 2.4. Attention values are normalized over columns ensuring that $\sum_n A_{n,n'} = 1$

$$\text{attention}(X, A) = X^T A \quad (2.4)$$

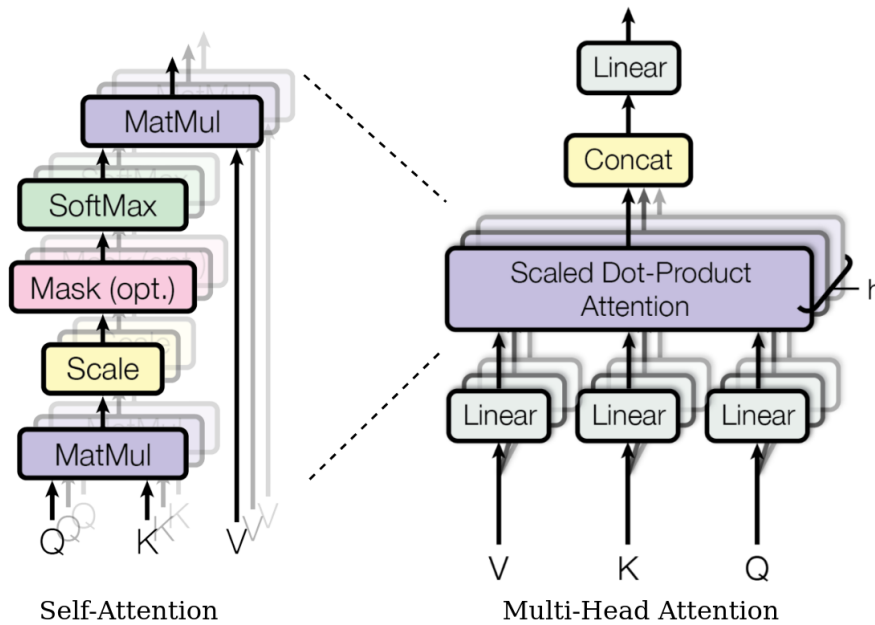


Figure 2.2 – **Multi-Head Attention**. Diagram showing the Self-attention on the left and multi-head attention on the right. (source: (Vaswani et al. 2017))

In transformers, attention is used to compute a weighted sum that contextualizes the input tokens, where the weights are determined by how much each token attends to the other tokens. This means that the attention matrix is computed by comparing the input against itself, thus the name self-attention. This is the main mechanism that allows transformers to learn long-range dependencies in sequences.

The self-attention mechanism can be implemented in a variety of ways, but the most common approach is to use the scaled dot-product attention. In this approach, the attention weights are computed as follows: Queries Q and Keys K are computed with two different linear transformations to the original sequence X , using a linear transformation respectively using the matrices $W_q \in \mathbb{R}^{K \times D}$ and $W_k \in \mathbb{R}^{K \times D}$ such as Equation 2.5 and Equation 2.6. Using different transformation on keys and queries ensure that the resulting self-attention matrix is asymmetrical, $A_{n,n'} \neq A_{n',n}$. In other words, this allows a value for the relation (token 1, token 2) to be different from (token 2, token 1). The word "wine" may have a high value toward the word "red" while, "red" may have a lower value toward "wine", because it is more common in the text to have "wine" associated with the color

"red" than the word "red" being associated with "wine". This flexibility allows for learning fine-grained contextual representations for each token.

$$Q = X^T W_q$$

with $Q \in \mathbb{R}^{D \times N}$, $W_q \in \mathbb{R}^{K \times D}$ and $X \in \mathbb{R}^{D \times N}$ (2.5)

$$K = X^T W_k$$

with $K \in \mathbb{R}^{D \times N}$, $W_k \in \mathbb{R}^{K \times D}$ and $X \in \mathbb{R}^{D \times N}$ (2.6)

The overall Self-attention matrix $A \in \mathbb{R}^{N \times N}$ is computed using Q and K matrices as a dot-product scaled by $\frac{1}{\sqrt{D}}$ and with a softmax as described in the Equation 2.7 and applied to the value matrix:

$$A = \text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$
 (2.7)

2.2.2 Multi-Head Attention

Given one triplet (Q, K, V), the attention between tokens is one-dimensional. To allow richer representations, transformers usually have multiple attention maps computed independently with a multi-attention Head. In other words, given h head, h different self-attention maps are computed for the same input, allowing to consider similarities between 2 tokens on h dimensions as depicted in the Figure 2.2.

The h outputs are concatenated into a $N \times h \cdot d_v$ matrix. Therefore, we can write the multihead-attention as in Equation 2.8.

$$\begin{aligned} X' &= \text{MultiHead}(Q, V, K) \\ &= \text{Concat}(\text{head}_1, \dots, \text{head}_h) W_b \\ &= \text{Concat}(\text{Attention}(Q_1, K_1, V_1), \dots, \text{Attention}(Q_h, K_h, V_h)) W_b \\ &= \text{Concat}(\text{Attention}(X^T W_q^1, X^T W_k^1, X^T W_v^1), \dots, \text{Attention}(X^T W_q^h, X^T W_k^h, X^T W_v^h)) W_b \end{aligned}$$
 (2.8)

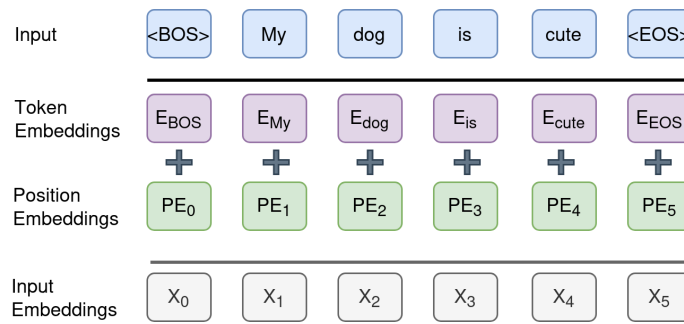


Figure 2.3 – **Positional Encoding.** Input representation. The final embedding X is a sum of token and positional embeddings

With X' the output of the Multi-Head attention, $X \in \mathbb{R}^{N \times D}$ the input matrix, $W_q \in \mathbb{R}^{d_{model} \times d_k}$, $W_k \in \mathbb{R}^{d_{model} \times d_k}$, $W_v \in \mathbb{R}^{d_{model} \times d_v}$, $W_b \in \mathbb{R}^{hd_v \times d_{model}}$ for projection matrices and learnable parameters.

2.2.3 Positional Encoding

As described in the previous paragraph, the attention mechanism and the position-wise FNN are operations independent of the token position in the sequence. This means that the sequence is seen as a bag-of-words by the model as it does not use convolution or RNN layers. To inject information about positions of tokens in the sequence (Vaswani et al. 2017) proposed to use positional encoding. To encode the position, each column x_i is added element-wise to a unique positional vector $pe_i \in \mathbb{R}^D$ as depicted in the Figure 2.3. This addition operation encodes the positional information of tokens in the sentences and allows more refined attention maps. The matrix PE can be either learned from linear operations or predefined using a function such as \sin and \cos like in the original work (Vaswani et al. 2017). Additional encoding can be done to inject additional information about the sequence. For example, authors of the Bert model (Devlin et al. 2019) used sentence types embeddings in addition to positional embeddings. Other positional encoders exist such as Rotary Positional Embedding (RoPE), which consists of rotating token embeddings (Su et al. 2022), and has proven to generalize better to positional encoding unseen during training.

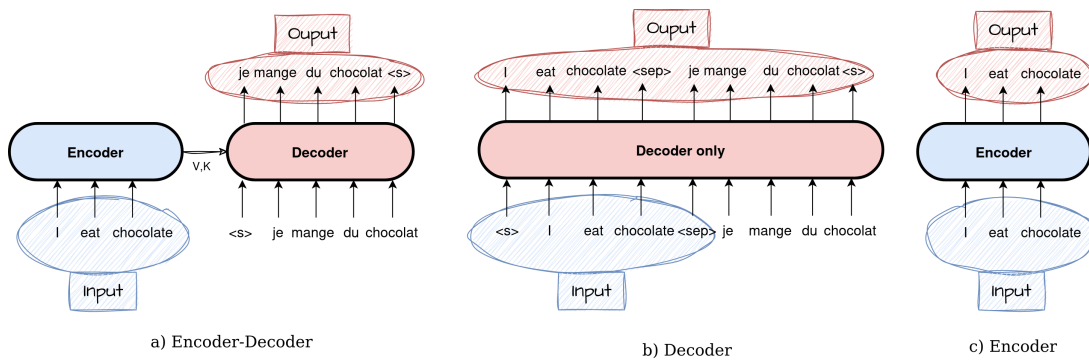


Figure 2.4 – **Encoder-Decoder, Decoder Only, Encoder only.** Encoder-Decoder architecture, Encoder is used to compute latent representations while decoder is used to generate new tokens.

2.2.4 Encoder/Decoder Cross-attention

So far, we described the multi-head attention and the position-wise FNN. These line-wise and column-wise operations form a TransformerBlock as shown in Equation 2.2. Multiple architectures can be built using these transformer blocks. Encoder, Decoder, or Encoder-Decoder, also called Sequence-to-Sequence model (Sutskever et al. 2014b). The original architecture proposed by (Vaswani et al. 2017) as shown in the Figure 2.1 is an encoder-decoder architecture. The encoder processes the input sequence and fixes its representations in a dense latent space. These latent representations are later fed into the decoder which generates autoregressively the output sequence. These are injected using a cross-attention mechanism. The resulting Keys and Values computed in the encoder are used in multi-head attention of the decoder as pictured in the Figure 2.1. Decoder-only architecture directly predicts the next token in a sequence as shown in the Figure 2.4. For sequence-to-sequence tasks, such as machine translation or question answering, the community relies on special tokens to distinguish input from output, otherwise, the model may just continue the input sequence.

2.3 Training

There are several training strategies to build language models as well as multiple training stages. First pre-training allows the model to learn useful representations. The finetuning steps are then used to encourage the model toward a wanted distribution. In what follows, we describe the different training steps and training objectives.

2.3.1 Pre-Training Objectives

Pre-training is the main training stage behind language modeling. The largest models' pre-training may take several months and cost up to 100 million dollars¹ on trillions of tokens (Zhao et al. 2023). It is performed on massive collections regrouping all sorts of unlabeled texts: books, news, webpages, forums, encyclopedias, scientific data, and code repositories. Models are trained under a self-supervised paradigm, with training objectives usually consisting of filling/predicting missing/masked words in texts. Multiple pre-training objectives are used in the literature for self-supervised training and have various data efficiency and performance trade-offs. All self-supervised training objectives for LM rely on predicting hidden words from texts. These objectives differ in how the words are masked and how the text is fed into the model.

The Figure 2.5 depicts attention map regarding multiple objectives. The grid structure represents how tokens attend to each other (attention map). The blue cells in the encoder indicate the input sequence, while the red cells in the decoder represent the target sequence. The what follows, we describe the depicted training objectives.

Masked Language Modelling A popular and efficient approach is called Masked Language Model (MLM) and consists in hiding a proportion of the tokens with a special 'mask' token (Devlin et al. 2019). The model considers the whole sequence of tokens to estimate missing tokens denoising the sequence. This is widely used for non-autoregressive models because MLM considers all other tokens with bi-directional attention. The bi-directional attention makes the training data efficient and allows for learning good token and sequence representations. The Figure 2.5 shows which tokens the model can consider in the attention to estimate a particular output y_i . Tokens x_1, x_2, x_3, x_4, x_5 are provided as input. The target consists of the same set of tokens but with one or more tokens masked (or hidden), $(x_1, x_2, x_{mask}, x_4, x_5)$.

Given x_{mask} denoting the set of masked tokens in the sequence, and $x_{\neg mask}$ the set of non-masked tokens, the objective is to maximize the likelihood L given the model's parameters θ Equation 2.9

$$L = \frac{1}{|x_{mask}|} \sum_{i \in x_{mask}} \log P(x_{mask_i} | x_{\neg mask}; \theta) \quad (2.9)$$

1. <https://en.wikipedia.org/wiki/GPT-4>

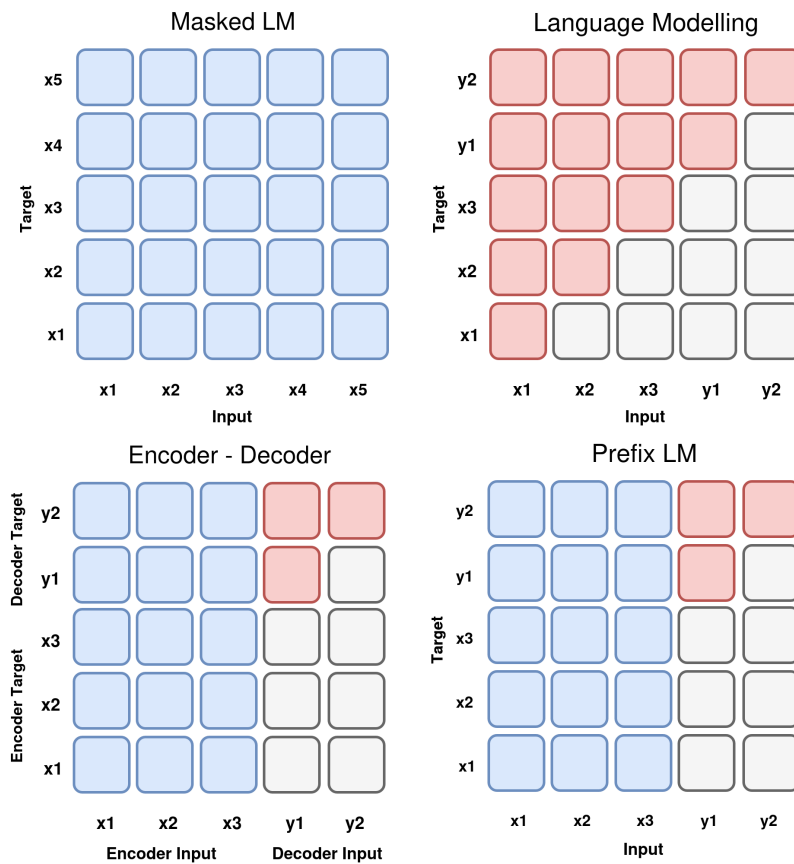


Figure 2.5 – **Pre-Training Strategies.** Various language modelling approaches. Blue tiles show attention with input tokens. Red tiles show attention with target tokens. Grey tiles show masked attention. MLM attention is computed considering all the tokens while autoregressive lm can only attend to previously decoded tokens.

Language Modelling Another commonly used training objective for preparing models that excel in text generation (Radford et al. 2018) involves Language Modeling, also referred to as the next token prediction task. In this approach, models are presented with sequences of tokens in a step-by-step manner and tasked with predicting the next token in the sequence. The Figure 2.5 shows that the model only attends to previous tokens in the sequence (uni-directional attention) to estimate the next token. For instance, tokens x_1, x_2, x_3 are given, and the model is trained to predict the next tokens in the sequence, here represented as y_1, y_2 . These can only attend to previously decoded tokens in the sequence. While being suitable for natural language generation (Radford et al. 2018), the training objective is not data efficient. Given the sequence of tokens, the training objective aims at maximizing the following likelihood L :

$$L(x) = \sum_{i=1}^N \log P(x_i | x_1, \dots, x_{i-1}; \theta) \quad (2.10)$$

Prefix Language Modeling Prefix Language Modeling serves as an intermediary approach bridging the gap between MLM and the training objective of traditional language modeling, as outlined in the work by Raffel et al. 2020a. In the context of Prefix Language Modeling, a sequence of tokens at the beginning, referred to as a "prefix sequence," is treated bidirectionally, with each token attending to both preceding and subsequent tokens in the sequence. However, the language modeling task is performed on the remaining portion of the sequence (see Figure 2.5). Tokens x_1, x_2, x_3 are provided as the prefix or starting sequence, based on the prefix, and the model predicts the next tokens in the sequence, depicted as y_1, y_2 in Figure 2.5.

Encoder-Decoder In the encoder-decoder architecture, the encoder is trained using MLM, where certain tokens are masked out, and the decoder is trained using traditional Language Modeling. This enables the encoder to acquire efficient representations and then to employ these fixed representations to guide the decoder during the Language Modeling process.

Teacher Forcing When training on language modeling, the model is fed a sequence of input tokens and is expected to generate the corresponding output sequence. In the absence of teacher forcing, the model's own generated output at each step is used as the input for the subsequent step. This means that the model's predictions are used as the input for generating the next token, leading to an autoregressive process where the model relies on its previous predictions. This is not suited for training because the prediction error may propagate in the following token prediction steps as the context diverges from the true output. To alleviate such a problem a common practice introduced by Williams and Zipser 1989a for recurrent Neural Net is to feed the true target at each time step. Teacher Forcing increases the training stability and convergence rate of neural networks and is commonly used for training.

2.3.2 Fine-tuning

During pre-training, the model learns to capture general features and patterns present in the data but is unable to perform any specific tasks. Fine-tuning a Pretrained Language Model (PLM) (Brown et al. 2020b) is a pragmatic and effective approach that capitalizes on the knowledge and the representations captured during pre-training while adapting the model's capabilities to the peculiarities of a target task. This allows to adjust parameters of the LM further and has proven to be much more efficient than training models from scratch. The fine-tuning stage usually requires examples of annotated data containing aligned input labels on specific tasks. Encoders can be trained to align various representations (eg: queries and documents (Karpukhin et al. 2020)), perform classifications (eg: sentiment classification (Devlin et al. 2019)) while decoder and encoder-decoder models are usually finetuned on a generative task such as conversational, machine translation (Raffel et al. 2020b)... Multiple studies show that scaling LMs sizes improve their performances. Large Language Model (LLM)s are both deep (number of cascading transformer blocks) and wide (number of neurons per layer) with several billions of parameters trained on massive corpora such as mC4 (6.4T tokens), MassiveText (2.4T tokens) or RedPajama (1.2T tokens). These models have achieved state-of-the-art performance on a lot of tasks. C. Zhou et al. 2023 and Brown et al. 2020b suggest that knowledge and capabilities are learned almost entirely during pre-training while the fine-tuning only shows how LLMs should behave. Authors were able to finetune a 65B parameter LLaMa language model on only 1,000 carefully curated prompts and responses for conversational tasks and managed to produce very high-quality output.

2.3.3 Reinforcement learning

During Pre-training and fine-tuning the loss is computed at the token level using cross-entropy between the predicted distribution over the vocabulary and the ground truth tokens. This poses several issues, during generation, the model predicts the most probable tokens individually without planning at the sequence level. Additionally, LMs are restrained to label distributions that are not always optimal regarding the tasks. Sequence-level training methods are used to enable the LM to explore parts of sentence space and prioritize tokens that are part of effective sentences. For example, an RL approach might involve the model generating an entire sentence, and then receiving a reward based on how well the

sentence meets certain criteria (like fluency, coherence, relevance, etc.). This helps the model learn to generate sequences that are effective as a whole, rather than just focusing on getting the next token right. Because tokens are sequentially sampled from predicted distributions to build sequences, the loss is not differentiable. Reinforcement Learning (RL) algorithms are used for various NLP tasks with non-differentiable automatic reward functions to finetune further LM at the sentence level. One of the main difficulties in training a language model from scratch with RL is the large action space, this is why previous works used PLM to constrain the exploration (Bahdanau et al. 2016; Ranzato et al. 2016; Donati et al. 2021; Ziegler et al. 2019; R. Nogueira et al. 2019a; R. F. Nogueira and Cho 2017; Ouyang et al. 2022b). In the context of language generation, RL can be used to adjust the model's output distribution to improve its performance on a specific task. Sentence performances are measured using a reward function adapted to the tasks. Automatic metrics such as BLEU/ROUGE (Papineni et al. 2002a) were often used for translation or summarization task (Ranzato et al. 2016; M. Liu et al. 2020). Some work also uses human feedback to fine-tune their model (Ouyang et al. 2022b; Ziegler et al. 2019) and align generations with human values. Human preferences are embedded into reward models that estimate a reward given a sentence. By using Proximal Policy Optimization (PPO) to maximize the reward, LMs can generate more fluent, comprehensive, and less toxic responses (Ziegler et al. 2019; Ouyang et al. 2022a; Roit et al. 2023; Bai et al. 2022).

2.4 LLM Zoo

Transformers allow learning semantic representations of words or phrases and make predictions about what words are likely to come next in a given context. LMs are generally trained under several training paradigms. Initially, they are pre-trained on extensive datasets through self-supervised learning. These PLMs are then fine-tuned for specific tasks using supervised learning. Unlike conventional DL models that usually comprise a few layers and are easily replicable, training LLMs from scratch is a resource-demanding and time-intensive endeavor. Consequently, only a limited number of PLMs are available and widely shared. Figure 2.6 depicts the current LLM Zoo, with model families typically spanning from a few hundred million to a few hundred billion parameters. These models are subsequently benchmarked² on a range of popular challenges covering a wide array of areas such as the Abstract Reasoning Challenge (ARC) (Clark et al. 2018),

2. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

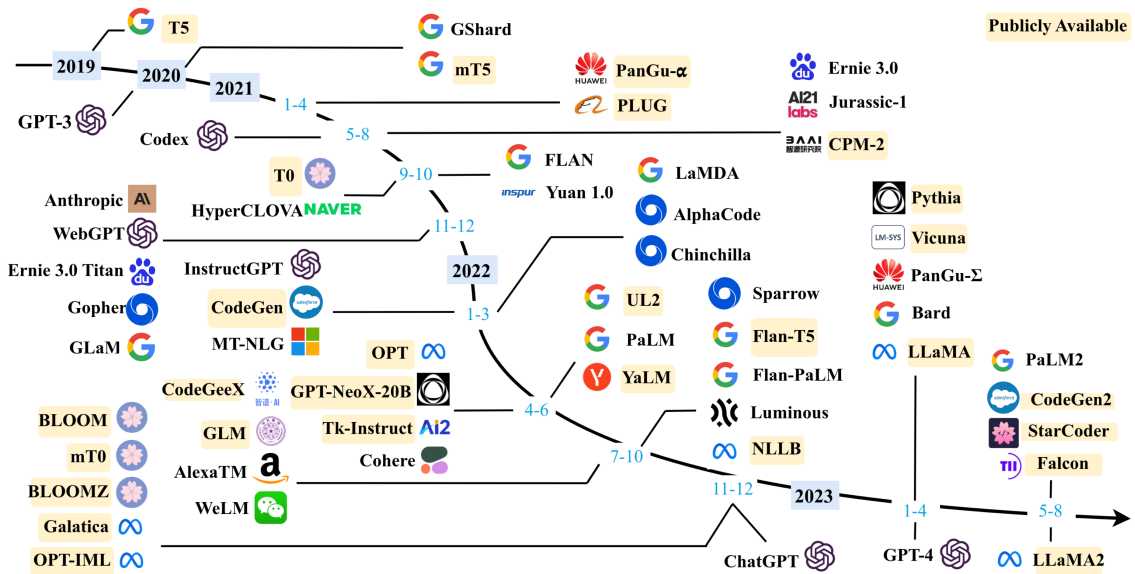


Figure 2.6 – **LLMs Zoo**. Diagram of existing LLMs. Publicly available (high-lighted) means that weights are shared but the code, training details or data are not always available (source: Zhao et al. 2023)

Multitask Language Understanding (MMLU) (Hendrycks et al. 2021), TruthfulQA, and even exams or competitions (OpenAI 2023). Since LLMs are primarily trained by private entities and may not always be publicly accessible, several open-source or academic initiatives have surfaced to facilitate the reproduction of these models (Workshop et al. 2023). Much of the academic research on LLMs hinges on a few foundational models specialized for distinct tasks or applications. For instance, the LLaMA model developed by META (Touvron et al. 2023) serves as a basis for numerous LLMs specialized in diverse domains like Programming, Math, Finance, Medicine, or Law, inheriting from it. The approach to new models has evolved beyond merely scaling models and data size or employing multilingual pre-training. Now, models are fine-tuned to a wide spectrum of tasks, encompassing instruction following, conversation, translation, summarization, and question answering, enhancing their transfer abilities to new tasks and zero-shot capabilities.

2.5 Emergent properties

Scaling up LMs has proven to improve performance on various tasks as well the emergence of unpredictable capabilities (Wei et al. 2022b; Brown et al. 2020b; Chowdhery et al. 2022). These emergent properties in LLMs are abilities that are

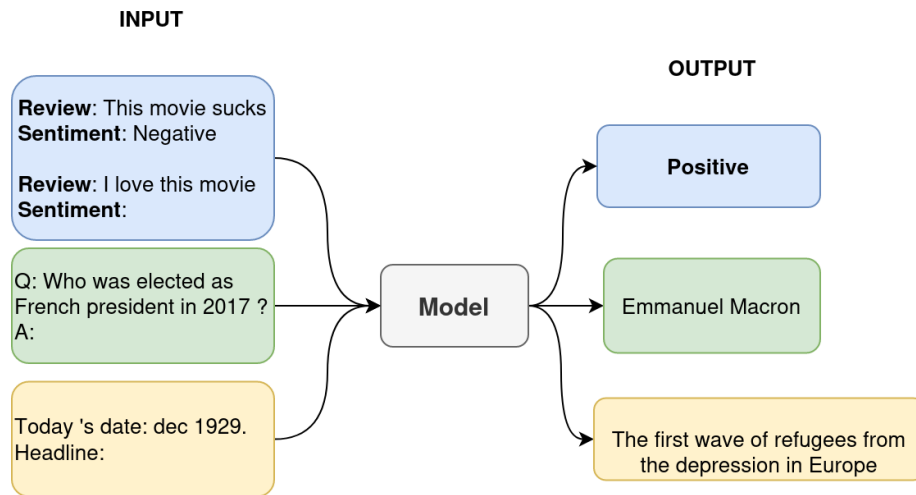


Figure 2.7 – **Emerging capabilities Examples.** Example of 3 emerging properties: Sentiment analysis (blue), Question answering in green. Conditional text generation is in yellow. These examples are generated using llama2-7B (Touvron et al. 2023)

not present in smaller models and arise unpredictably in larger models. These properties are difficult to explain and often lead to abrupt performance gains in various tasks. These emergent abilities include performing arithmetic, answering questions, and summarizing passages without being specifically finetuned for these tasks, as shown in Figure 2.7.

One example of an emergent property in LLMs is the ability to learn new tasks quickly and efficiently by reproducing patterns in the context but without being specifically trained for these tasks. This is known as few-shot learning. For example, an LLM may be able to learn a new task after being given just a few examples such as answering questions, and summarizing texts as depicted in Figure 2.8.

Additionally, the benefit of several finetuning can be observed only at a certain scale, with a clear jump in performances such as instruction following, 8-digit addition... These abilities emerge depending on model size as seen in the Table 2.1

2.6 Limitations of LLMs

Hallucinations. Generative language models are known to produce mistakes, inaccuracies, or incorrect statements called Hallucination (Dziri et al. 2022) referring to a neurological disorder causing the brain to misinterpret incoming sensory

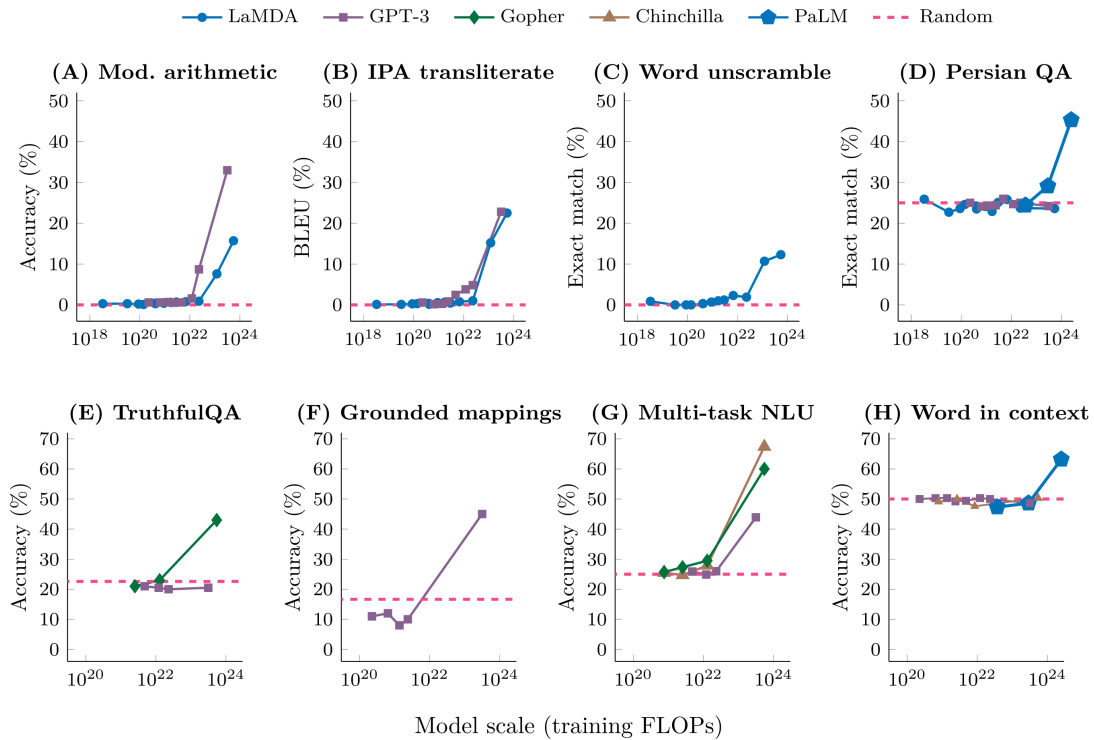


Figure 2.8 – **Performance against LLMs Scale.** This figure shows the performance given the size of the models. We can see an abrupt performance gain with the scale. (Wei et al. 2022b)

stimulation. This is a concern, especially for tasks requiring grounded and factual information generation. Cause and mitigation of hallucinations are studied in various NLG tasks requiring factual and faithful generations such as machine translation, dialogue generation, data-to-text, and summarization. Hallucinations fall into two main categories: 1) Intrinsic hallucinations when the model’s output contradicts the input context, for example for the input: ‘The Apollo Lunar Module Eagle landed on Dec 21, 1969’ contradicts the input text ‘The Apollo Lunar Module Eagle landed on July 20, 1969’ (Ji et al. 2023) 2) Extrinsic hallucinations are information that cannot be verified by the input context. This additional information may be True or False and appear in documents seen during pre-training or not. While LLMs can be used as knowledge sources (Roberts et al. 2020; Petroni et al. 2019; Metzler et al. 2021) for various applications such as conversational and Question Answering (QA) systems, extrinsic hallucinations can be harmful as there is no evidence that generated sequence is factually correct without access to external resources.

Hallucinations are the results of multiple factors: the primary factor behind this issue is the divergence of data between input and output within the supervised

Table 2.1 – **List of emergent abilities of LLMs.** List of emergent abilities of large language models and the scale (both training FLOPs and number of model parameters) at which the abilities emerge. (Wei et al. 2022b)

Ability	Train. FLOPs	Params.	Model
Few-shot prompting abilities			
Addition/subtraction (3 digit)	2.3E+22	13B	GPT-3
Addition/subtraction (4-5 digit)	5.1E+23	175B	GPT-3
MMLU Benchmark (57 topic avg.)	3.1E+23	175B	GPT-3
Toxicity classification (CivilComments)	5.0E+22	7.1B	Gopher
Truthfulness (Truthful QA)	3.0E+23	280B	GPT-3
MMLU Benchmark (26 topics)	5.0E+23	280B	GPT-3
Grounded conceptual mappings	3.1E+23	175B	GPT-3
MMLU Benchmark (30 topics)	5.0E+23	70B	Chinchilla
Word in Context (WiC) benchmark	2.5E+24	540B	PaLM
Many BIG-Bench tasks (see Appendix E)	Many	Many	BigBench
Augmented prompting abilities			
Instruction following (finetuning)	1.3E+23	68B	FLAN
Scratchpad: 8-digit addition (finetuning)	8.9E+19	40M	LaMDA
Using open-book knowledge for fact checking	1.3E+22	7.1B	Gopher
Chain-of-thought: Math word problems	1.3E+23	68B	LaMDA
Chain-of-thought: StrategyQA	2.9E+23	62B	PaLM
Differentiable search index	5.3E+22	11B	T5
Self-consistency decoding	3.1E+23	68B	LaMDA
Leveraging explanations in prompting	5.0E+23	280B	Gopher
Least-to-most prompting	3.1E+23	175B	GPT-3
Zero-shot chain-of-thought reasoning	3.1E+23	175B	GPT-3
Calibration via P(True)	2.6E+23	52B	Anthropic
Multilingual chain-of-thought reasoning	2.9E+23	62B	PaLM
Ask me anything prompting	1.4E+22	6B	EleutherAI

dataset. The annotations within the dataset may include information that is not consistently tied to the input, leading the LLM to produce text that is not always closely grounded to the input context (Dhingra et al. 2019; Parikh et al. 2020; C. Zhou et al. 2021). The number of occurrences in the training data impacts the learned distribution and therefore the prediction quality. Katherine Lee et al. 2022 showed that duplicate data in the pre-training encouraged memorization which may introduce hallucinations. Decoding techniques, such as beam search, aim to reduce the perplexity of the token sequence generated. However, this reduction in perplexity does not consistently align with the accuracy or veracity of the information being conveyed (Ji et al. 2023). Furthermore, the introduction

of randomness into the decoding process, through mechanisms like top-k or top-nuclei sampling, enhances the diversity of generated outputs. Yet, it is important to note that this increased diversity also brings about a higher occurrence of hallucinatory content.

Lastly LLMs memorize knowledge in the parameters during pre-training (Madotto et al. 2020; Petroni et al. 2019; Roberts et al. 2020) and restore this knowledge on various downstream tasks. While this may improve performance and generalization capabilities for some generative tasks (Brown et al. 2020b), LLMs tend to prioritize their knowledge over the input context, therefore generating hallucinations (Longpre et al. 2021).

Attention complexity. Traditional attention mechanisms, particularly those used in models like the Transformer, have a complexity that scales quadratically with the sequence length. This means that the computational resources (both time and memory) required for these models increase significantly as the sentences they process grow longer. This limitation makes it challenging to process long documents or maintain extensive context, especially in real-time applications. To mitigate this, various strategies have been proposed such as Windows Attention or Streaming LMs (Xiao et al. 2023). This approach reduces the computational burden and allows for longer sequences by sacrificing some global understanding.

Autoregressive LM. Autoregressive LM often generates text based on patterns seen during training, without an actual understanding of real-world logic, causality, or common sense (Momennejad et al. 2023; Mahowald et al. 2023; Valmeekam et al. 2023). This gap can lead to nonsensical or contextually disconnected outputs. As these models generate sequences token-by-token, early mistakes can snowball, causing errors to accumulate and potentially leading to nonsensical or off-track generation as the sequence progresses. Additionally, they cannot revise their outputs—they lack mechanisms to revisit and revise text based on new context or corrections, leading to error accumulation. This is the main reason why these types of models cannot plan their answers and need a lot of prompt tricks such as chain of thought (CoT) (Wei et al. 2022c) or tree of thought (ToT) (Yao et al. 2023) to fill the context with guidance and reduce errors. Multiple propositions has been explored such as diffusion language models that denoise sequences in parallels (Strudel et al. 2022).

Latent sequence embedding that guides the generation. Scheduled Sampling (S. Bengio et al. 2015) where models learn to update previously decoded tokens that minimize the overall perplexity. During pre-training predicted tokens are fed

as input to the model with the probability of updating the later to the gold token. Another method would be to introduce a latent variable (a fixed-size vector) that is meant to capture the high-level information or essence of the sequence, where the generation process is conditioned on this latent variable.

2.7 Summary

In this chapter, we saw that **PLMs** can be used to for various **NLP** tasks and can compute accurate and contextual representations. **PLMs** can serve as strong building blocks for conversational search applications.

TOWARD CONVERSATIONAL SEARCH SYSTEMS

Chapter abstract

Conversational Search (CS) systems are systems that aim at finding relevant information through natural language dialogues with users. Information Retrieval (IR) systems are integral to these systems as they allow retrieving information in vast collections of documents. After a quick overview of IR history, we expose sparse and dense retrieval models based on LM. We then present CS systems, evaluations, and associated challenges.

A survey on user simulations, was presented in the Sim4IR workshop at SIGIR 2021:

- Pierre Erbacher, Laure Soulier, and Ludovic Denoyer (2022c). “State of the Art of User Simulation approaches for conversational information retrieval”. In

Contents

3.1	Information Retrieval	36
3.1.1	Historical Overview	36
3.1.2	Neural Ranking Model	38
3.1.3	Evaluation and Performance Measures	39
3.2	Conversational search systems	41
3.2.1	General Overview	41
3.2.2	Asking Clarifying Question	45
3.2.3	Evaluating Conversational Search	46

3.1 Information Retrieval

Information Retrieval (IR) is a field within computer science and information science that focuses on the efficient and effective retrieval of relevant information from large collections of structured or unstructured data given a query. It is primarily concerned with developing techniques and algorithms to enable users to find and access information that matches their information needs or queries. [Figure 3.2](#) illustrates an IR system, a representation of the query is computed and used to find closed document representations in an index. The system may additionally take into account the browsing history or relevance feedback to adjust query representations and retrieve more relevant documents. IR systems are used in a wide variety of domains that involve data collection. This includes multiple modalities of texts, images, videos, music, and database of chemical structures or genomes. In this section, we briefly introduce and summarize the main concepts of IR. After a brief historical overview, we describe various documents and query representations as well as algorithms used for ranking documents.

3.1.1 Historical Overview

The first efforts to preserve and organize document collections date back to 2600 BC in Mesopotamia, with the appearance of the first libraries. These were places where large quantities of information could be stored and accessed. The library in the royal palace of Ashurbanipal, king of the Assyrians, in Nineveh (northern Iraq, near Mosul) registered up to 30,000 clay tablets. The develop-

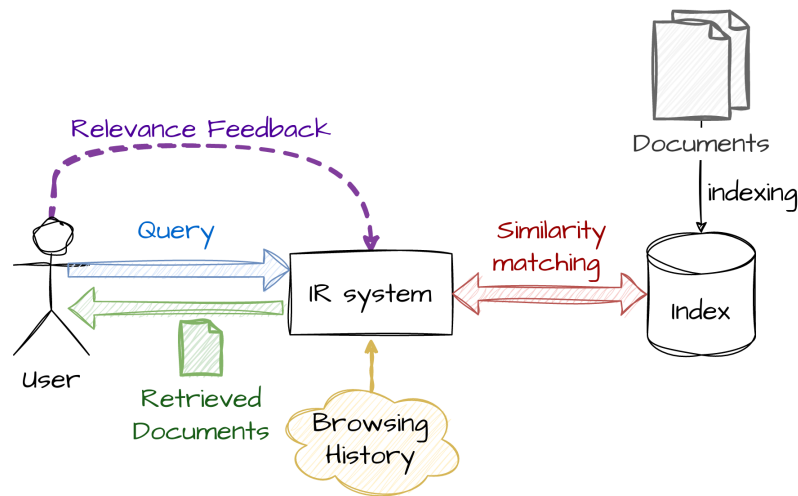


Figure 3.1 – **Diagram of Information Retrieval System.**

ment and evolution of libraries have played an important role throughout history. It allowed to conserve the records of administrative documents (e.g., transactions, governmental orders, and budget allocation) and collection of resources on different domains such as medical science, astronomy, history, geometry, and philosophy. The development of more convenient writing material support, such as papyrus, allows libraries to store increasingly more documents. The Library of Alexandria (fl. 285–145 B.C.) acquired between 40,000 and 400,000 papyrus scrolls, the Library of Pergamum (197–159 B.C.) collected over 200,000 volumes and the Imperial Library of Constantinople (337–361 A.D. – 29 May 473) collection was estimated over 100,000 volumes. Organizing, indexing, and retrieving documents was originally performed by librarians using cataloging schemes usually containing, authors' names, dates, and subjects. While cataloging schemes improved over the years, the process of searching for information remains slow and painful with ever-growing quantities of documents. In the 1920s Emanuel Goldberg conceived the first automatic systems for IR. This electromechanical device was able to search letters and dot patterns through a catalog stored on microfilms (Sanderson and Croft 2012) The search rate is further improved in the 30s with Shaw's rapid selector. This machine was able to perform 78,000 entries per minute in film indexing by 430,000 entries (Sanderson and Croft 2012).

The democratization of computer fast improvement in index and search algorithms. In the early stages of IR systems, Boolean retrieval was the dominant search method, where queries consisted of logical combinations of terms, resulting in exact document matches. However, an alternative approach called ranked retrieval emerged. In this approach, documents were assigned scores based on their relevance to a query, and documents were sorted by these scores, with the

top-ranked ones returned to the user. This approach outperformed Boolean search in experiments. Salton et al. 1975 played a significant role in IR research, and developed ideas such as vector-based document-query comparisons using cosine similarity. Salton's vector space model became a foundational approach for IR systems.

Sparck Jones (1988) introduced the concept of term frequency (tf) and inverse document frequency (idf) weights, which were combined as tf-idf, enhancing document ranking. They proposed to give terms a weights in documents regarding their frequency in the collections (Sparck Jones 1988). From information theory's perspective, the quantity of information provided by an event increases as the probability of this event decreases. In other words, frequent terms are more likely to occur in documents and therefore provide less specific information about the documents as rare terms. One limitation of the tf-idf is that long and more verbose documents contain more terms and are ranked higher than shorter documents. In the 1980s to mid-1990s, variations of tf-idf weighting schemes were developed. The Bm25 introduces and utilizes the document length dl and the average document length $avdl$ to weight terms (Robertson and Zaragoza 2009).

TF-IDF and BM25 are the most popular IR algorithms and are still widely used. Today, IR models are able to search in collections containing several billions of documents thanks to parallel computing, and advances in Machine Learning (ML) and Natural Language Processing (NLP) allow computing better document and query representations. In the following, we expose some state-of-the-art neural retrieval methods that rely on LMs and compute either sparse or dense representations for queries and documents.

3.1.2 Neural Ranking Model

The ability of transformer-based Language Model (LM)s to generate contextual token embeddings is a strong argument in IR. This allows dense contextual representations of documents or queries that capture text semantics. Various query/-document encoding methods exist in the literature as shown in Figure 3.2 ranging from sparse to dense with various granularity in query/documents interactions.

Representation-based similarity models consist of using query and document encoders to compute independent representations (usually vectors) Figure 3.2. These serve as first-stage retrieval systems as document representations can be stored in indexes. These encoders are trained to minimize the similarity of queries

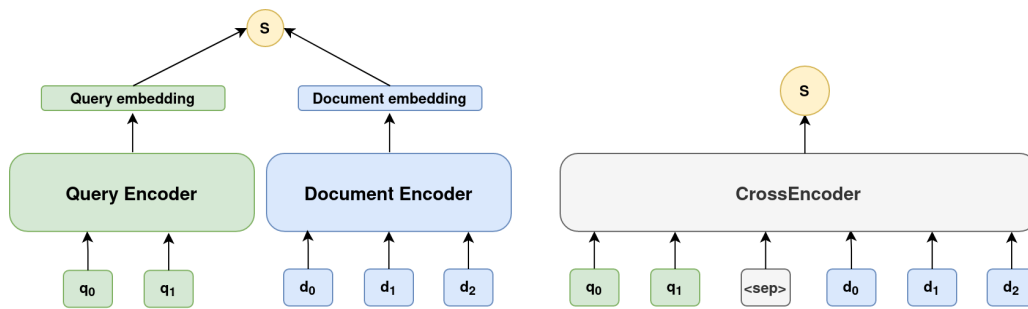


Figure 3.2 – **Dense Retrieval/Ranking Models.** Diagram of Dense models Left: dual encoders that align query and document representations. Right: cross-encoder computing a relevance score between query and document.

with relevant documents and maximize the similarity with non-relevant documents. (Karpukhin et al. 2020). These encoders may share the same set of parameters (siamese network) (Hofstätter et al. 2021) or have disjoint sets of parameters.

The **all-to-all interaction models** also called **cross-encoders** or **second stage re-ranker**, consist in encoding both queries and documents in the sequence (Pradeep et al. 2021). The self-attention is computed using queries and documents leading to fine-grained interactions between tokens embeddings. These models are trained to classify if a document is relevant given the query. The score predicted by the model is used to re-rank documents regarding a query.

Another category of neural retriever models aims at learning sparse representations. For example, Formal et al. 2021 proposed Splade, which learns sparse representations for queries and documents. For each token in the sequence, the LM predicts a score for each token in the vocabulary. Predicted values indicate the importance of a token regarding a document. These values are regularized to make sure that no terms are dominant and by introducing FLOPS, a sparsity regularizer to control the number of terms in document representations.

3.1.3 Evaluation and Performance Measures

Evaluating ranking models is a long-standing challenge that generally follows the Cranfield paradigm (Voorhees 2019). Test sets are commonly annotated by human evaluators. They are presented with a set of queries and tasked with estimating the relevance of documents very sparsely within a collection. This is achieved by assigning a relevance score to each query-document pair. The scores

typically range from -1 or 0 (not relevant) to 1, 2, or 3 (relevant). These sets are benchmarks utilized for the automatic evaluation of the effectiveness of document ranking on collections. Various metrics are used to evaluate IR systems. Because the quality of retrieved documents is evaluated by users based on the initial few search results, the IR typically concentrates on evaluating the top retrieved documents using a ranking cut-off k .

Given query $q \in Q$, retrieved document $d \in D$ the following metrics.

Precision and Recall The precision is defined as the ratio of relevant documents retrieved in retrieved documents:

$$P = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved}} \quad (3.1)$$

The recall is defined number of retrieved relevant documents over the total number of relevant documents :

$$R = \frac{\text{number of relevant documents retrieved}}{\text{total number of relevant documents}} \quad (3.2)$$

Mean Average Precision (MAP) The MAP takes into account both the precision and the recall of the retrieved documents :

$$\text{MAP} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{\text{num}_q} \sum_{k=1}^{\text{num}_q} P(k) \cdot \text{rel}(k) \quad (3.3)$$

Where num_q is the number of relevant documents for query q . $P(k)$ is the precision at cutoff k . $\text{rel}(k)$ is an indicator function that is 1 if the document at rank k is relevant, and 0 otherwise.

Mean Reciprocal Rank (MRR) This metric computes the inverted rank of the first retrieved relevant documents in the ranking as in [Equation 3.4](#).

$$\text{MRR} = \frac{1}{|Q|} \sum \frac{1}{\text{rank}_q} \quad (3.4)$$

with rank_q the position of the first relevant document for the query $q \in Q$

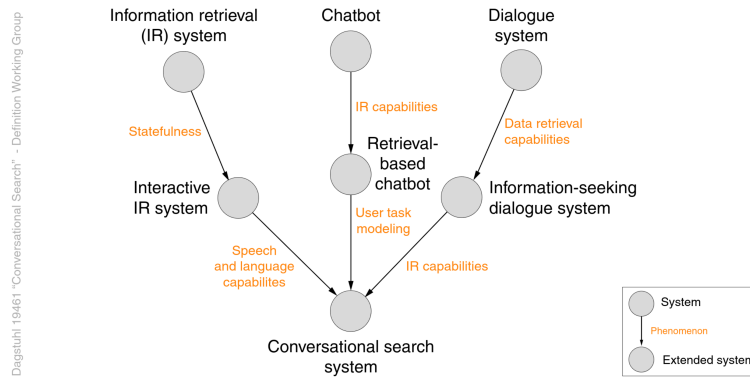


Figure 3.3 – Dagstuhl Conversational Search Typology. (Anand et al. 2020)

Normalized Discounted Cumulative Gain (NDCG) takes into account both the relevance of the items and their position in the list. NDCG is normalized to produce values between 0 and 1, where a higher NDCG indicates a better-ranked list. The formula for NDCG at a specific cutoff k :

$$\text{NDCG}@k = \frac{\text{DCG}@k}{\text{IDCG}@k} \quad (3.5)$$

$$\text{DCG}@k = \sum_{i=1}^k \frac{2^{\text{rel}_i} - 1}{\log_2(i + 1)} \quad (3.6)$$

Where $\text{DCG}@k$ (Discounted Cumulative Gain) measures the quality of the ranked list up to position k as given in Equation 3.6 and $\text{IDCG}@k$ (Ideal Discounted Cumulative Gain) represents the best possible $\text{DCG}@k$ value that could be achieved and rel_i is the relevance score of the item at rank i .

3.2 Conversational search systems

3.2.1 General Overview

Conversational Search (CS) is a well-established research field at the intersection of IR and NLP (Culpepper et al. 2018). CS systems are characterized by their ability to retrieve/extract information from large document collections, modeling users' needs through active dialogue using natural language (Zamani et al. 2023). Figure 3.3 pictures the Dagstuhl typology (Anand et al. 2020) that aims

at defining CS. The CS can be defined through multiple domains and may lead to concurrent research approaches. From IR perspective, CS is an IR system that is given to ability to dialog with users. CS systems may also be seen as chatbots given the ability to search documents and model users' needs. This approach received a lot of attention with Large Language Model (LLM) chatbots endowed with the ability to query search engines before answering. However, these systems still lack the ability to model users' needs or perform exploratory searches. The last definition proposed in Figure 3.3 is a task-oriented dialogue systems given IR capabilities. These task-oriented dialogue systems are often used in specific domains for conversational recommendation, able to interact with users to understand their needs and refine recommendations. In addition, CS systems may have similarities with systems in other application domains (interactive IR, conversational systems, question-answering systems, recommender systems) but differ in many aspects. 1) Unlike interactive IR (Zhai 2020) that is also interested in user feedback, CS's objective is to find the relevant information in a more natural way (i.e., interactions in natural language). 2) It is also different from task-oriented conversation systems (Bordes et al. 2017) which are guided in a particular application area (i.e., travel booking). A CS system is a specific application case of task-oriented conversation systems but is different because the search is performed in an open domain. 3) Finally, very close to question answering systems (Bordes et al. 2014) in which questions are expressed in natural language, CS focuses on exploratory or complex information needs, on open fields, not specifically centered on a specific entity, question or document as in Q&A systems. The objective of a CS system is to help the user in his/her interactions with a search engine through a conversational tool. This whole search engine/conversational system will aim to: 1) respond to a complex information need expressed in natural language and/or through various feedbacks (e.g., sequences of reformulation of requests, clicks on documents, conversations in natural language) and 2) anticipate and guide the user in his/her sequence of search actions using, e.g., query clarification or document ranking (Azzopardi et al. 2018; Radlinski and Craswell 2017). This requires CS systems to interact with users to better understand and/or refine their information needs in a long-term objective. The democratization of voice-only assistants such as Google Assitant, Siri, or Alexa pushes IR search from the standard query-document to dialogue paradigm. CS systems aim at supporting the users in the information-seeking process by minimizing effort, cognitive load, and frustrations and helping them retrieve relevant information through a conversation. This means the interactive nature of web navigation classically performed 'manually' by users through querying, reading, evaluating, clicking, and scrolling is transposed into natural languages.

System: I am here to help you explore open data! This collection includes more than 18,000 open datasets from countries such as Italy, Ireland, Mexico, Austria, etc.

User: What data do you have about Austria?

System: Jobs, healthcare, population statistics, education, environment... Are you interested in anything from the above?

User: What else?

System: Traffic data, sports, agriculture, tourism.

User: Traffic sounds interesting! What kind of data is there?

System: There is data for different cities in Austria: Vienna, Linz, Salzburg, Innsbruck.

User: ok

System: For example, pedestrian zones in Vienna, bridges locations, Electronic timetable information for Linz.

User: Great! Show me something else.

System: There are 96 datasets about Education and Research in Austria. One of them is a List of all driving schools in Burgenland.

User: What is Burgenland?

System: Burgenland is the easternmost and least populous state of Austria [Wikipedia].

Figure 3.4 – **Example of conversational exploratory search.** (source:(Vakulenko et al. 2017)). The user does not have a clear search intention and performs an exploratory search by interacting with the system

Thus CS system must have mixed-initiative capabilities to take full advantage of the conversation as depicted in the example [Figure 3.4](#). This proactivity can serve to engage users by suggesting various topics/sources, summarizing retrieved documents, soliciting feedback, or asking clarifying questions about the search intent. While Question Answering (QA) systems deal with very specific and factual questions (known-item search) such as "What is the capital of France ?" CS systems may additionally deal with more exploratory search associated with vague instructions such as depicted in [Figure 3.4](#). Exploratory Search requires systems to provide a wide spectrum of topics or sub-topics and let the user decide what to explore, trajectories may be difficult to predict and depend on users' expertise or interest.

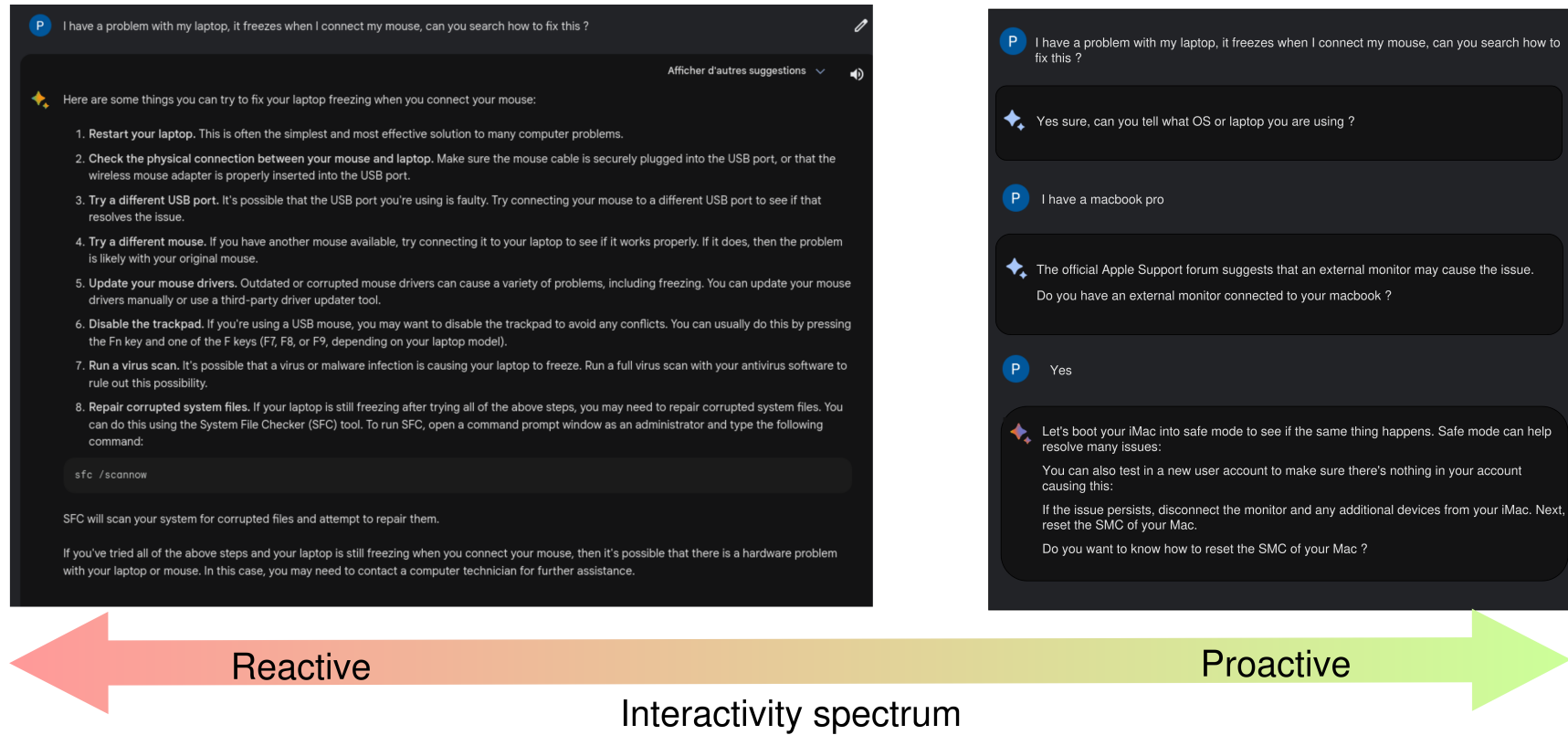


Figure 3.5 – **Interactivity Spectrum.** Diagram showing the interactivity spectrum with 2 examples. The left example shows a reactive system answering the query, the answer is not specific to the user’s need (generated with Google’s Bard). The right example shows a proactive system that refines the user needs with clarifying questions (manually generated example)

In this thesis, we focus on the ability for CS systems to disambiguate or refine user’s query by asking clarifying questions.

3.2.2 Asking Clarifying Question

When the user intent is not clear, asking clarifying questions is an effective method that allows the user to be involved in the query disambiguation process. This helps the system to provide useful responses and narrow down the search space. Clarifying Questions are very diverse and have various goals. Braslavski et al. 2017 provide a taxonomy of clarification types for questions presented in Table 3.1. As described by Zamani et al. (2023), these clarifications may have multi-

Clarification Type	Example
More information	What OS are you using?
Check	Are you on a 64-bit system?
Reason	What is the reason you want a drip pan?
General	Can you add more details to this question?
Selection	Are you using latex or oil based Kilz?
Experience	Have you tried to update video card drivers?

Table 3.1 – **Taxonomy of clarifying questions.** (Braslavski et al. 2017)

ple objectives. When queries are vague or may be open to multiple interpretations, CQ can help disambiguate the need and narrow down the search space. They can also be used for personalization to align searches and answers with users’ expectations or preferences. For example, by asking for personal information like age, expertise, gender, and localization the system may be able to refine the search and provide answers adapted to the user. Kiesel et al. (2018) conducted a user study demonstrating that users tend to favor systems that request confirmation before retrieving search documents rather than providing direct results, especially when the retrieved documents are irrelevant. Moreover, Zou et al. (2023) showed that low-quality CQs tend to disrupt users, resulting in decreased interaction with the Search Engine Result Page (SERP)s. For open-domain conversational IR, a first line of works focus on identifying CQs relies on ranking strategies applied to a pool of predefined, human-generated candidates. In the pioneering work, Aliannejadi et al. (2019) propose a ranker that iteratively selects a clarifying question at each conversation turn. Bi et al. (2021) complete this approach with an intent detection based on negative feedback and a Maximal Marginal Relevance-based BERT. Hashemi et al. (2020) use a transformer architecture to retrieve useful clarifying questions using the query and the retrieved documents. However, leveraging a

fixed question pool will limit the coverage of topics, and therefore hinder the effectiveness of the approach. To overcome this limitation, a second line of work aims at generating clarifying questions. In Salle et al. (2021) use templates and facets collected from the Autosuggest Bing API to generate clarifying questions. At each turn in the conversation, they select a new facet to generate the question until the user's answer is positive. This inferred facet is then used to expand the initial query. Sekulić et al. (2021) propose to further improve the fluency by using a LM to condition the clarifying questions generation on the initial query and a facet. For instance, the query 'Tell me about kiwi', conditioned to facets 'information fruit' or 'biology birds' can generate questions like "Are you interested in kiwi fruit?" or 'Are you interested in the biology of kiwi birds?'. They rely on ClariQ dataset (Aliannejadi et al. 2019) to fine-tune GPT2, and found that generated questions are more natural and useful than template-based methods. They have extended this work by generating questions using facets extracted from retrieved documents (Sekulić et al. 2022b). Zamani et al. (2020a) propose to generate clarifying questions associated with multiple facets (clarifying panels) which are collected using query reformulation data. They investigate three methods to generate clarifying questions: templates, weak supervision, and maximum likelihood, and use reinforcement learning to maximize the diverse set of clarifying panel candidates.

The literature review highlights that there is a lack of adequate large-scale datasets containing mixed-initiative interactions for the IR task.

3.2.3 Evaluating Conversational Search

While QA and recommendation have well-established benchmarks, designing an adapting framework for evaluating CS systems is still challenging in the IR community. Indeed, CS involves both dialogue and IR tasks that should lead to mixed-initiative interactions to help the user navigate in documents (Jeffrey Dalton et al. 2022). A CS system should therefore be able to suggest topics, generate questions to clarify/ elicit users' information needs and retrieve documents providing relevant information. The diversity of tasks underlying search (known-item, exploratory search, task-oriented,..) and all the possible conversational trajectories make it challenging to measure interaction success. There are two main strategies to train and evaluate these systems either, leveraging existing datasets, often at the cost of not having all the dimensions of conversations, or simulating interactions between the user and the system. We briefly review some typical attempts for different types of conversational systems.

Dataset	# Domain	Scale	# Clar. Q	Documents	User Answer	Resource	Clar. Source
Qulac	198	10K	3K	Yes	Yes	TREC WEB	Crowdsourcing
ClariQ	300	10K	4K	Yes	Yes	TREC WEB	Crowdsourcing
MSDialog	12	35K	877	No	Yes	MC	Crowdsourcing
MIMICS	-	462K	586K	No	No	Bing users	Machine Generated
MIMICS-Duo	-	1K	1K	No	No	Bing users	MTurk

Table 3.2 – Overview of Conversational Search Datasets

On question answering (QA), the datasets have been extended from one-shot QA such as Squad (Rajpurkar et al. 2018), Quac (Choi et al. 2018), ELI5 (Fan et al. 2019), or OpenQA (D. Chen et al. 2017) to conversational Q&A such as coQA (Reddy et al. 2019). One can train/evaluate answer generation systems, and possibly IR systems using the collection of passages. Despite this interesting move, the datasets are insufficient for IR because they usually focus on factual questions instead of complex or vague questions that characterize information needs. The TREC CAsT dataset (Jeffrey Dalton et al. 2020) extends the scope of questions and addresses different information facets within the conversation (a facet can be seen as a specific sub-category of the topic). However, the interactions provided are often limited to answer generation without proactive interactions engaging the system in real support of search. Other datasets, such as CANARD (Elgohary et al. 2019), focus on query refinement or reformulation, without proactive interactions. Therefore, most approaches focused on generating reformulated queries as input to ranking systems (Hai et al. 2023). The TREC CAsT track (Jeffrey Dalton et al. 2020) introduced a pool of clarifying questions related to the proposed IR-oriented conversations, without providing the associated users' responses. This dataset constitutes a first step toward exploring mixed-initiative CS but does not have complete and coherent conversations.

Several initiatives have been taken to build query clarification datasets as observed in Table 3.2. For example, the MIMICS dataset (Zamani et al. 2020a) contains large-scale open domain clarifying questions collected from real users on the Bing search engine. The query clarifications are associated with initial users' queries and other information such as clicks. However, this dataset does not provide document relevance judgments or conversational interactions between the user and the system. To the best of our knowledge, only Qulac and ClariQ datasets contain both document relevance judgments and the associated mixed-initiative conversations. These datasets are built from the TREC Web Track 2009-12 collection (Clarke et al. 2009a), which provides annotated topics and facet pairs, associated with relevant documents. Users' responses have been collected through crowd-sourcing platforms, building a complete dataset of mixed-initiative interactions grounded in an ad-hoc IR dataset. However, collecting these interactions

is costly and the datasets remain small with only 237 topics and 762 topic facets. This is too limited for training and evaluating conversational retrieval systems.

Facing the lack of adequate datasets, a growing idea in the community is to rely on user Simulations to evaluate CS systems (Erbacher et al. 2022a; Salle et al. 2021). User simulations mimicking users' queries and feedback are cost-efficient and allow for the evaluation of various strategies without direct data annotations. For example, Salle et al. (2021) evaluate their query clarification systems with a user simulation aiming at generating answers. Their user simulation relies on a BERT model fine-tuned to classify "Yes"/"No" answers to the clarifying questions. With a controllable parameter, the user simulation can also add words from the intent in the answer to simulate more or less cooperative feedback. Sekulić et al. (2022a) confirmed with an additional human judgment that user simulations can generate answers and give feedback with fluent and useful utterances. User simulation is also exploited to design evaluation frameworks for conversational recommender systems (Kang et al. 2019; C. Gao et al. 2022; F. Wu et al. 2020; K. Zhou et al. 2020; Fu et al. 2020), resulting in large synthetic dialogue interactions from ad-hoc recommendation datasets (Kang et al. 2019; C. Gao et al. 2022; F. Wu et al. 2020; K. Zhou et al. 2020; Fu et al. 2020). However, in the recommendation context, we notice that conversations are generated under explicit search constraints over annotated features like price range, color, location, movie genre, or brand, whatever the generation approaches used – either by using LLMs (Asri et al. 2016) or following agenda (sequence of requests)(Schatzmann et al. 2007; Peng et al. 2018; X. Li et al. 2017; Kreyssig et al. 2018). Unfortunately, similar approaches cannot be used for complex and exploratory search tasks (Belkin and Croft 1992). In open-domain IR, the facets underlying information needs are not necessarily discrete or easily identifiable, making it much harder to identify and annotate users' needs.

This literature review highlights that Asking clarifying questions and understanding user information need remains a challenge. This mainly comes from the difficulty to collect large and diverse conversational search datasets. Ideally, such datasets should contain natural search oriented conversation between systems and users and document relevance judgments annotated. There is a need to construct synthetic interactions' data for CS application.

Part II

MULTI-TURN INTERACTIVE INFORMATION
RETRIEVAL

MULTI-TURN INTERACTIVE MODEL WITH USER SIMULATION FOR QUERY REFINEMENT

Chapter abstract

The work in this chapter, done in collaboration with Ludovic Denoyer and Laure Soulier, was published as a short paper at the SIGIR conference 2022:

- Pierre Erbacher, Ludovic Denoyer, and Laure Soulier (2022a). *Interactive Query Clarification and Refinement via User Simulation*. arXiv: 2205.15918 [cs.IR].

Contents

4.1	Introduction	52
4.2	Question Clarification Simulation Framework	53
4.2.1	Overview and Research Hypotheses	53
4.2.2	The IR System Agent	55
4.2.3	The User Agent	56
4.3	Evaluation Protocol	57
4.3.1	Dataset	57
4.3.2	Baselines and Scenarios	57
4.3.3	Model Implementation	58
4.4	Results	59
4.4.1	Preliminary Analysis	59
4.4.2	Effectiveness Results	61
4.5	Conclusion and perspectives	62

4.1 Introduction

In the previous chapter, we have seen that user simulations are used to train and evaluate recommender systems and task-oriented dialogue. However, their application to open domain Information Retrieval (IR) is still limited to a single turn for query clarifications. In this chapter, we address this limitation and propose to explore multi-turn query refinement using a user simulation for open-domain search.

To this end, we propose to build a fully simulated query clarification framework allowing multi-turn interactions between IR and user agents. Following (Aliannejadi et al. 2019), the IR agent identifies candidate queries and ranks them in the context of the user-system interactions to clarify the initial query issued by the user agent. We particularly target simple information needs, multiple information needs are left for future work since they might impact the modeling of the query ranking function. Our framework can be seen as a basis and a proof-of-concept for future work willing to integrate sequential models (namely reinforcement learning models) for question clarification. All agent components in our framework are based on continuous and simple models. To validate our simulation framework, we conducted an experimental analysis on the MS MARCO dataset. We show

the benefit of multi-turn interactivity and evaluate the effectiveness of different question selection strategies.

4.2 Question Clarification Simulation Framework

4.2.1 Overview and Research Hypotheses

Our query clarification simulation framework is inspired by Aliannejadi et al. (2019) but provides the possibility to leverage user-system agents' interactions sequentially. More particularly, our framework is illustrated in Figure 6.1 and relies on the following workflow:

- A) The user issues an initial query q_0 associated to her/his information need i to the IR system.
- B) The IR system generates a set $Q = \{q_1, q_2, \dots, q_m\}$ of candidate queries which might express different query reformulations or diversified queries to better explore the information need i .
- C) The IR system selects N queries to display to the user. To do so, we propose to follow (Aliannejadi et al. 2019) and design a model ranking the candidate query set Q to identify the top N queries.
- D) The user selects one of the N queries, enabling the extraction of positive and negative feedback, resp. noted (q^+, q^-) .
- Steps C) and D) can be repeated several times to model multi-turn interactions. The query set ranking function (step C) integrates the user's sequential feedback (q^+, q^-) to improve the query ranking along with the interaction simulation.
- E) After T turns, the IR system considers the best-ranked query as the optimal query reformulation and runs a ranking model to retrieve documents.

The design of this evaluation framework is guided by some choices/research hypotheses.

- First, following (Aliannejadi et al. 2019), we consider a fixed set of candidate queries $Q = \{q_1, q_2, \dots, q_m\}$ constituting the reformulation of the initial query q_0 . All the interactions are leveraged to improve step by step the ranking of this candi-

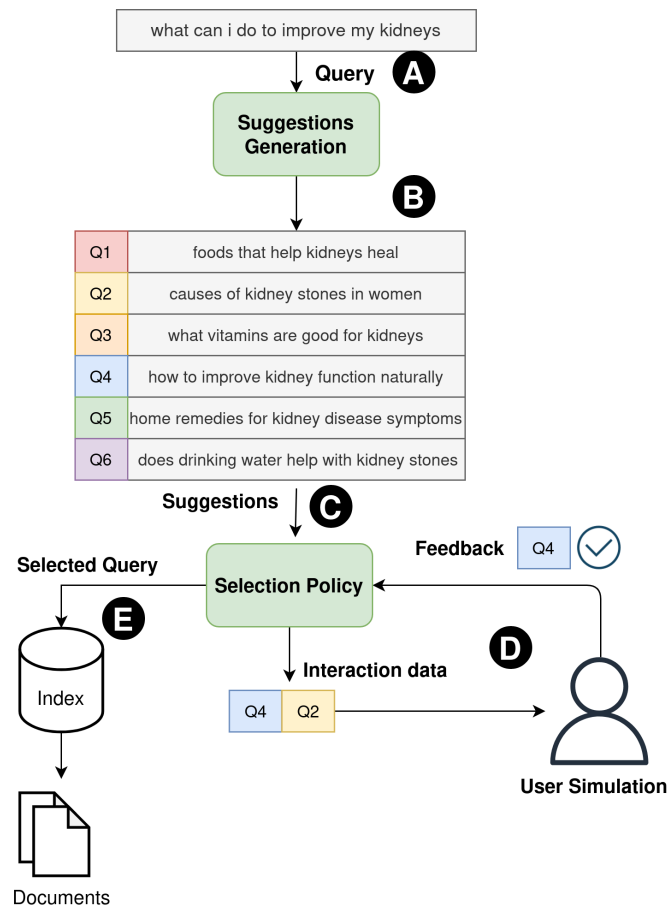


Figure 4.1 – **Query clarification simulation framework.** The input query is used to generate a set of query reformulations that are later used iteratively to generate user feedback.

date query set so that, at the end of the session, the final query used for retrieving documents is a good clarification of its initial one. Obviously, this means that the set of candidate queries includes a large variety of queries which, for some of them, improve the search performance.

- Second, following (Zamani et al. 2020a), we propose to model question clarification as a possible option between two reformulated queries. In other words, expressed in natural language, the IR system agent would ask the user agent the following question: "Which reformulated query do you prefer? A or B". This implies that the user is willing to judge queries A or B regarding its information need.
- Third, guided by the motivation to propose a framework for future work on sequential models, we consider here that each agent component is modeled at the embedding level. Indeed, leveraging Large Language Model (LLM)s for generating/ranking questions is very effective, but integrating them into reinforcement

learning models is still challenging (one main reason being the computational cost). This means that we processed *a priori* all queries and documents to represent them using text embeddings. This processing is done offline, alleviating the sequential modeling of the text encoding.

In what follows, we present the different components behind the IR system and user agents.

4.2.2 The IR System Agent

The IR agent has three objectives in our framework: 1) generating the set of candidate reformulated queries that are suggested to the user, 2) ranking this set to identify the most relevant queries according to the interaction history, 3) ranking documents using the best-ranked query (ending the interactive session).

4.2.2.1 Generation of the candidate reformulated query set.

The objective here is to instantiate various and diverse reformulations covering a wide range of relevant topics for the initial query q_0 . Different techniques might be used, leveraging LLMs (R. Nogueira et al. 2019c; Raffel et al. 2020b; Rao and III 2019), query diversification (Cai et al. 2016; MacAvaney et al. 2021; Ye et al. 2021) or query expansion (Pal et al. 2013). We propose here to use the T5 model (Raffel et al. 2020b) which is designed to translate token sequences into other token sequences. It has already been used for query reformulation tasks, demonstrating its ability for our approach (J. Z. Chen et al. 2020; Raffel et al. 2020b; S.-C. Lin et al. 2020). We finetune the model on the reformulations task. The model takes as input a query q and outputs a query q' . We rely on the parametric knowledge of the models to output new tokens semantically close to the original query, possibly covering multiple topics. On the top of that model, the generation process is driven by beam diversity (Vijayakumar et al. 2016) which aims at generating a set Q of diversified query reformulation, $Q = \{q_1, q_2, \dots, q_m\}$.

4.2.2.2 Ranking of queries based on the interaction history.

The role of the selection policy is to select queries used to interact with the user agent. Following (Aliannejadi et al. 2019) which proposes to rank queries according to both performance criteria and the interaction context, we use a conditional ranker (Burges et al. 2005) which computes a pairwise score between two candidate queries given the context, namely the initial query q_0 and the

additional information provided by interaction with the user. Let q_i and q_j be the candidate queries with their supervised effectiveness scores, resp. y_i, y_j . The ranking model relies on:

$$P(y_i > y_j | q, q_i, q_j, feedback_{t-1}, \dots, feedback_1) \quad (4.1)$$

For the sake of simplicity, we assume that each query (initial or candidate) is represented through text embeddings. In the following, q refers to query embedding, and d to document embedding.

In practice, the ranking model estimates a score for each query q_i and q_j given all the context, $\{q, q_i, q_j, feedback_{t-1}, \dots, feedback_1\}$ and then compare these scores to identify which one is the most relevant. $feedback_t$ corresponds to selected or not selected queries (resp. q^+ and q^-) by the user agent at interaction turn t . These queries are concatenated as follows: $feedback_t = (q^+, q^-)$ and feedback overall interaction turns are aggregated, the whole process using a Hierarchical Recurrent Neural Network (RNN) to encode at the interaction level and also the sequence of interactions. Note that queries q^+ and q^- are encoded differently using resp. a *cosin* and *sin* function. Moreover, we do not encode the position in which each query is presented to the user agent, as the latter does not have position bias on the clarification query selection.

4.2.2.3 Final ranking of documents

Documents are retrieved with the top-ranked query using a Dense Retriever model (Hofstätter et al. 2021).

4.2.3 The User Agent

After issuing the initial query q_0 , the user agent interacts with the IR system agent to refine her/his information need. With this in mind, we hypothesize that the user is greedy toward her/his intent and fully cooperative. Therefore, he always selects the preferred query as the most similar to the intent. Despite being unrealistic, we ignore the click bias problem for the clarification panel presented in (Zamani et al. 2020c; Zamani et al. 2020d) (as mentioned earlier). Other choices for user simulation could be done, following (Câmara et al. 2022), but we let these variations for future work.

In practice, let d be a user intent, q_i , and q_j the clarification queries presented to the user agent. The user agent selects the best query (noted q^+ for highlighting

positive feedback from the user) according to a similarity metric (in our case, the dot product) between the proposed queries q_i and q_j and intent d :

$$q^+ = \operatorname{argmax}_{q_i} (\langle q_i, d \rangle) \quad (4.2)$$

The non-selected query, q^- , expresses negative feedback.

4.3 Evaluation Protocol

Evaluating our simulation framework consists of measuring the effectiveness of the final ranking after T clarification interactions. Since the user behavior is greedy and follows a simple behavior dependent on the query selection process, the effectiveness results mainly denote the quality of this latter component. Other components (candidate set generation and final document ranking) do not depend on the interaction feedback, so we mainly focus on understanding whether the selection policy integrates users' feedback and makes good decisions to select the N clarification questions.

4.3.1 Dataset

We carry out our experiments on MS MARCO 2020 passages which regroups 8.8M passages and more than 500K Query-Passage relevance pairs. Following (R. Nogueira et al. 2019b), we evaluate our model on 2 test sets. The small test set (43 queries) and a subset of the dev set (1000 queries sampled from 59 000). One motivation to consider these two datasets is their difficulty level: in the dev set, only one passage per query is labeled relevant in the ground truth, while several passages are considered relevant in the test set.

4.3.2 Baselines and Scenarios

To evaluate the effectiveness of our selection policy component, we compare with:

(1) Non-interactive settings to show the gain of interacting with users. We measure the ranking effectiveness of the user's initial user query (noted **User Query**), the **Best Reformulation** in the candidate query set - which can be seen as

an oracle, and the **MonoT5** Documents re-ranker which acts as a strong ranking baseline (Pradeep et al. 2021). This model is a pointwise ranking, estimating relevance scores for query/document pairs. This model relies on a large sequence-to-sequence language model pre-trained on various tasks (Raffel et al. 2020b). Please note that using this model for the selection policy, and therefore integrating user’s feedback, is not obvious since this is a seq2seq pointwise model, labels associated with queries are binary (relevant or not) and have to be grounded relative to a value. For that reason, we only consider its non-interactive scenario.

(2) **Naive interactive selection:** At each step, we select the 2 top ranked queries from the current query rank and then remove the query that has not been selected by the user agent. The re-ranking of the candidate query set is only carried out once, at the beginning of the session, and the size of this list decreases with the interaction number.

To instantiate the selection policy after each interaction-driven query ranking step (step C in Figure 6.1), we consider these scenarios:

(1) **Interact. + Random Sampl:** we sample 2 queries from the ranked candidate query set to constitute the interaction pair.

(2) **Interact. + Top 2:** we select the top 2 query reformulations at each turn.

(3) **Interact. + random sampl@5:** we randomly select 2 queries among the top 5 query reformulations at each turn.

(4) **Interact. + Kmeans selection:** At each turn, queries in the candidate set are clustered in 2 groups using Kmeans. Queries from each cluster are ranked by the model. The best-ranked query from each cluster is then selected for interaction with the user. The cluster of the query not selected by the user is removed for the next turn from the set of candidate queries. This strategy corresponds to a refinement strategy, removing a group of semantically similar queries that have not been chosen by the user and going deeper into the other cluster.

4.3.3 Model Implementation

All queries and passages embeddings are pre-computed using the Dense Retriever proposed by (Hofstätter et al. 2021). Embeddings are stored and indexed using faiss HSWN32 index (Johnson et al. 2021). The candidate query set is generated by a diversity beam with a group penalty equal to 0.6. The size of the candidate set is 64. The number of queries displayed to the user agent is set to $N = 2$. For the model hyper-parameters, we use batches of size 128, the optimizer is Adam ($\beta_1 = 0.9, \beta_2 = 0.99$) with weight decay (= 0.01). We use batch

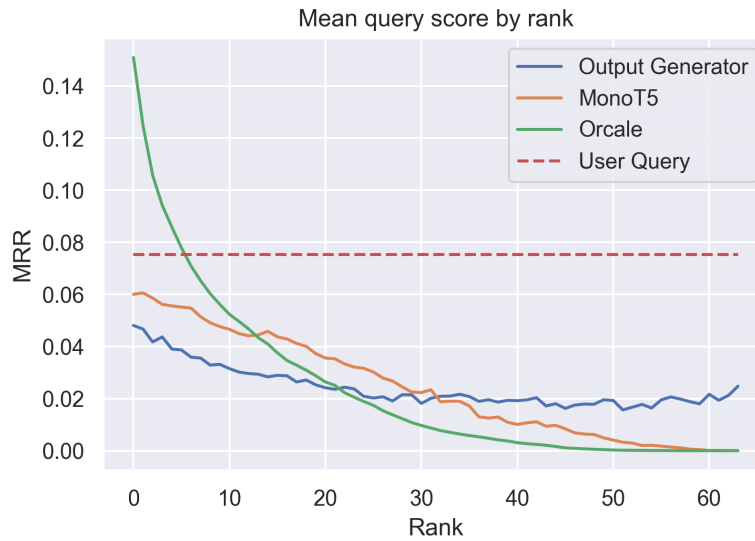


Figure 4.2 – Effectiveness score of query reformulation by rank

normalization and dropout ($p = 0.3$) between each layer. The learning rate is set to 1×10^{-4} .

4.4 Results

4.4.1 Preliminary Analysis

We present here a preliminary analysis to quantify the potential retrieval performance gain of the candidate query set within the question clarification step. To do so, we compare the performance of different query rankings: 1) the candidate queries generated by the T5 model ranked by decreasing order of likelihood resulting from the Diversity Beam search (without application of our ranking function); 2) the Oracle corresponding to the candidate query set ranked in decreasing order according to their performance according to Mean Marginal Rank metric in the ground truth. We emphasized that this Oracle curve shows the performance of our T5 model at generating search-oriented reformulations. 3) The MonoT5 ranking corresponds to candidate queries re-ranked by MonoT5. Figure 4.2 illustrates the performance of queries depending on their rank in the different mentioned lists. From Figure 4.2, we can see that ranking queries with MonoT5 allows us to improve the performance for the top k queries (MonoT5 vs. Output Generator). This has a negative effect on the end of the list, but it is not critical in our case,

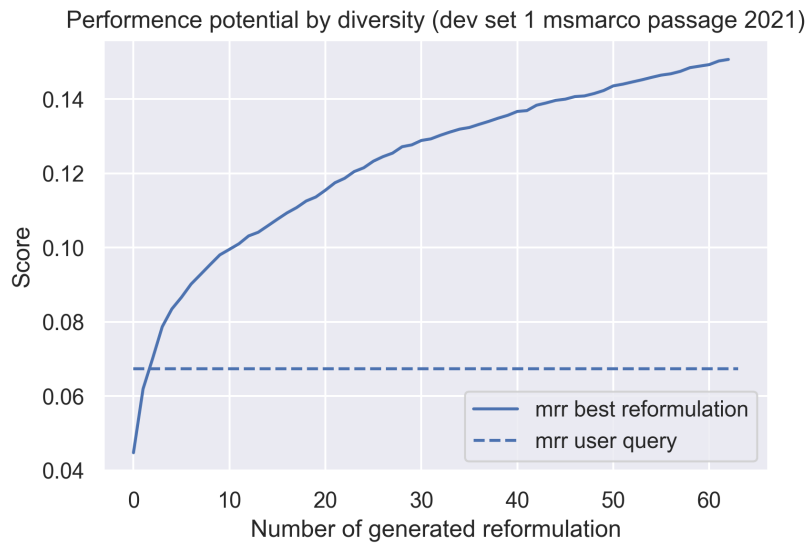


Figure 4.3 – Maximal Score with increasing number of queries in the set

since we consider selection policy regarding the top query list. Moreover, one can notice that, although performance has increased, there is still a gap between the curve of the MonoT5 ranked list and the Oracle curve. Our intuition is that leveraging users’ interactions will lower this gap, which leads to the evaluation we performed in what follows.

We believe that the user query can be optimized with few interactions by simple strategies such as query diversification and query ranking. To validate that, we performed a preliminary analysis by measuring the theoretic performance gain of query diversification. For each query, we generate a set of diversified queries using diversity beam (Vijayakumar et al. 2016). We then measured the maximal retrieval performance considering a variable number of query reformulations generated. We used user queries to generate reformulation and measured their performances. Reformulations are generated using a T5 model fine-tuned on a reformulation task. We perform the analysis on the 3000 queries of the dev set 1 of MS MARCO passage 2021 (Nguyen et al. 2016a). The Figure 4.3 shows the Mean Marginal Rank of the best reformulation. By increasing the number of generated reformulations, the maximal MRR also increases. This indicates that adding diversity is beneficial and has a higher chance to generate better reformulations, leading to better performances.

		No interaction	1	2	3	4	5
User Query	mrr@10	0.4554	-	-	-	-	-
	map@10	0.3382	-	-	-	-	-
Best Reformulation	mrr@10	0.8720	-	-	-	-	-
	map@10	0.5646	-	-	-	-	-
Monot5 (query ranker)	mrr@10	0.4713	-	-	-	-	-
	map@10	0.3209	-	-	-	-	-
Naive selection	mrr@10	0.2135	0.3270	0.3597	0.4036	0.4191	0.4271
	map@10	0.1222	0.1943	0.2205	0.2553	0.2688	0.2766
Interact. + random sampl	mrr@10	0.4031	0.4786	0.4814	0.4903	0.4814	0.5019
	map@10	0.2531	0.3344	0.3413	0.3529	0.3480	0.3685
Interact. + Top 2	mrr@10	0.4031	0.4746	0.4693	0.4903	0.4786	0.5019
	map@10	0.2531	0.3294	0.3436	0.3520	0.3471	0.3428
Interact. + random sampl@5	mrr@10	0.4031	0.4734	0.4670	0.4903	0.4798	0.5019
	map@10	0.2531	0.3287	0.3420	0.3517	0.3469	0.3451
Interact. + Kmean	mrr@10	0.4031	0.5232	0.4658	0.4692	0.4863	0.5515
	map@10	0.2531	0.3706	0.3207	0.3402	0.3181	0.3347

Table 4.1 – **Results Multi-turn Test set MS MARCO.** Results on the Test set of MS MARCO passage 2020 (43 queries - multiple relevant documents per query)

		No interaction	1	2	3	4	5
User Query	mrr@10	0.2090	-	-	-	-	-
Best Reformulation	mrr@10	0.4119	-	-	-	-	-
Monot5 (query ranker)	mrr@10	0.1557	-	-	-	-	-
Naive selection	mrr@10	0.1228	0.1513	0.1659	0.1767	0.1866	0.1911
Interact. + random sampl	mrr@10	0.1719	0.2012	0.1990	0.1954	0.2003	0.2016
Interact. + Top 2	mrr@10	0.1719	0.2020	0.1987	0.1973	0.2017	0.1990
Interact. + random sampl@5	mrr@10	0.1719	0.2020	0.1983	0.1966	0.2007	0.2008
Interact. + Kmean	mrr@10	0.1719	0.1748	0.1984	0.2016	0.2158	0.2224

Table 4.2 – **Results Multi-turn Dev set MS MARCO.** Results on the subset of MS MARCO passage 2020 dev set (1000 queries - 1 relevant document per query)

4.4.2 Effectiveness Results

We analyze here the performance of the query ranked at different interaction turns using mrr@10 and map@10. Table 4.1 and Table 4.2 resp. show the results on the MS MARCO passage 2020 test set and dev set. From a general point of view, we can see that performance metrics are lower for the dev set than for the test set. This can be explained by the task difficulty, which is higher for the dev set in which only one document per query is assessed as relevant. By comparing all baselines and scenarios, we can outline the following trends. 1) The first candidate query ranking within our interactive models (No interaction columns) provides lower performance than non-interactive baselines. For instance, the **Interact. + Top2** scenario yields a decrease of 12% in terms of mrr@10 for

the test set w.r.t. the initial user query. 2) But this trend is reversed with each interaction turn to obtain for some scenarios performance higher than baseline ones (see all interaction models in the test set, and the **Interact + Kmeans** for the dev set). 3) The interaction model with Kmean strategy looks to be the best selection policy for question clarification since it obtains the highest $mrr@10$ for both datasets. This is somehow intuitive because this strategy might correspond to a refinement strategy, going deeper and deeper into clusters. This is also connected with the dataset peculiarity since MS MARCO is mainly composed mono-faceted questions in natural language.

4.5 Conclusion and perspectives

This exploratory work focuses on sequential click-based interaction with a user simulation for clarifying queries. We provide a simple and easily reproducible framework simulating multi-turn interactions between a user and a IR system agent. The advantage of our framework is the simplicity of interaction, as there is no need for a dataset of real and annotated user-system interactions. Experiments highlight performance gain in terms of document retrieval through the multi-turn query clarification process and provide a comparative analysis of selection strategies. The next steps for this work are 1) leveraging reinforcement learning for the selection policy, 2) testing more diverse and more sophisticated user simulations, as done in (Câmara et al. 2022) for multi-faceted information needs.

OPTIMIZING GENERATION FOR MULTI-TURN QUERY CLARIFICATION WITH REINFORCEMENT LEARNING

Chapter abstract

The work in this chapter, done in collaboration with Laure Soulier, is under review

- Pierre Erbacher and Laure Soulier (2023). *CIRCLE: Multi-Turn Query Clarifications with Reinforcement Learning*. arXiv: [2311.02737](https://arxiv.org/abs/2311.02737) [cs.IR].

Contents

5.1	Introduction	64
5.2	A generative model for multi-turn query clarification with reinforcement learning	65
5.2.1	General overview	65
5.2.2	Query clarification Policy	66
5.2.3	Model training and inference	68
5.3	Evaluation protocol	69
5.3.1	Dataset	70
5.3.2	Metrics	70
5.3.3	User Simulation	70
5.3.4	Baselines	71
5.3.5	Implementation details	72
5.4	Results	72
5.4.1	Retrieval Performance	72
5.4.2	Qualitative example	73
5.4.3	Suggestions Coverage	74
5.4.4	Impact of user cooperation	74
5.5	Conclusion and perspectives	75

5.1 Introduction

In the previous work, the main limits of the multi-turn interactive query clarification come from the simplicity of the user simulation and the use of non-learned strategies to identify query suggestions. In this chapter, we address these problems and introduce a multi-turn query clarification model aiming at generating new and diverse query suggestions between each interaction turn. Our model relies on pre-trained Language Model (LM)s and leverages both supervised learning and reinforcement learning to clarify the initial query issued by the user. More particularly, we fine-tune the language model with a policy at the sequence level such as in (Ouyang et al. 2022b). In addition, our policy contributes to the generation of a set of clarification queries that are optimized to balance diversity and effectiveness. Based on our previous work (Erbacher et al. 2022a), we integrate our multi-turn query clarification model into our proposed simulation framework to mimic user-system interactions. It is worth noting that in such a framework we

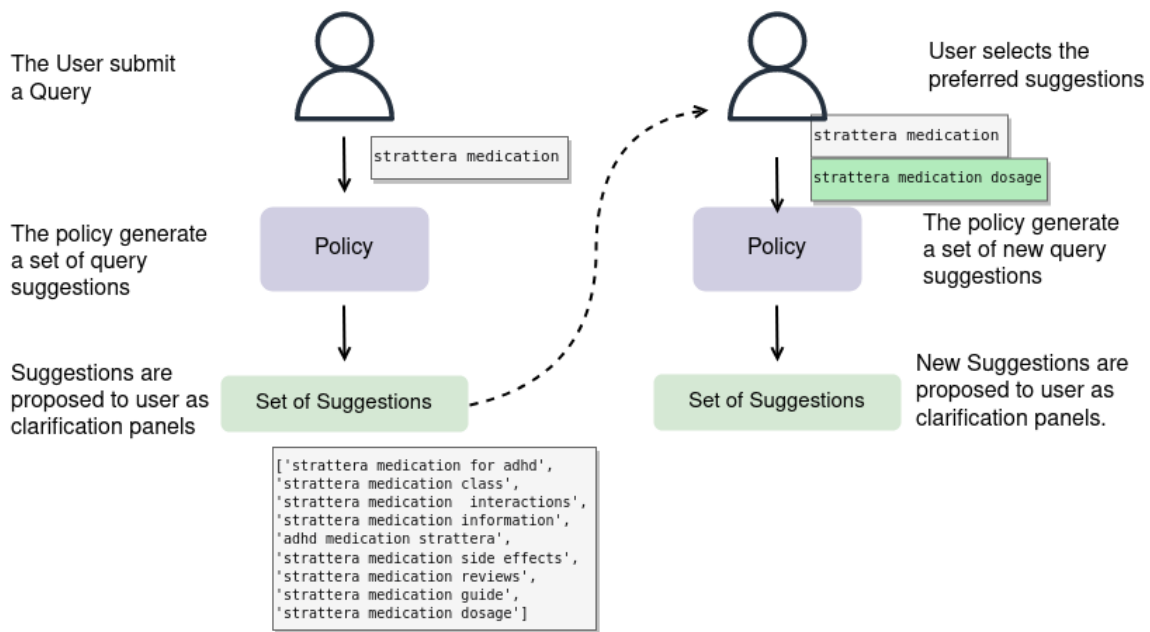


Figure 5.1 – Multi-turn Query clarification framework

expect our model to embed a multi-turn level policy that optimizes the trajectory over all interactions. We let this perspective for future work since it will imply a very sparse reward, and training such a policy is more complex.

To show the benefit of multi-turn interactivity and evaluate the effectiveness of our model, we conduct an experimental evaluation using the MS Marco dataset in a simulated framework. We compare our results against well-established search engines that can rely on logs of billions of users to construct query suggestions.

5.2 A generative model for multi-turn query clarification with reinforcement learning

5.2.1 General overview

Our model, called CIRCLE, aims at generating a set of query clarifications using reinforcement learning and is evaluated within the sequential interaction framework proposed in (Aliannejadi et al. 2019; Erbacher et al. 2022a). This framework, shown in the Figure 5.1 consists of iteratively proposing query suggestions to the user in response to his/her query. The user feedback collected at each interaction step is used to generate the following set of suggestions. A retrieval model can

be launched at each interaction to retrieve documents and evaluate the quality of the selected query. Formally, let's consider the user query as a sequence of tokens $x = w_0, \dots, w_{t-1}$ sampled from a distribution of queries D , the goal is to learn a model using policy $\pi(w_t|w_{t-1}, \dots, w_0)$ that completes the sequence x producing the set of queries $Y = Y = \{y_1, \dots, y_K\}$. Queries are separated with a $\langle sep \rangle$ token in the sequence.

With this in mind, our model is based on the following intuitions:

- **LMs**, well-known in the literature (Radford et al. 2019), have demonstrated their skills in generating sequences. We will therefore rely on decoders to generate query clarifications based on an initial information need.
- To provide a choice to the user, we need to generate a set of query clarifications. We leverage supervised pairs of (initial query-set of query suggestions) to fine-tune the model to generate a set of query clarifications autoregressively. In other words, we utilize a decoder-only architecture to generate a sequence of tokens that expresses the sequence of query clarifications, separated with a special token.
- To force the diversity of the different generated query clarifications within the sequence, we exploit reinforcement learning techniques estimating the similarity between clarifications while maintaining the distribution near the supervised model.

5.2.2 Query clarification Policy

The objective of our CIRCLE policy is to generate a set of query clarifications that are both 1) grounded near the user query and 2) Sufficiently diverse to let the user explore the potential universe of keywords able to specify his/her need. To do so, our model is based on a reinforcement learning approach designed on top of a finetuned **LM**. It is worth noting that the reinforcement learning approach is focused at the iteration level to generate more diverse sets of query suggestions for each turn, but does not embed a multi-turn level policy.

In what follows, we describe these two components of our model.

Supervised learning Numerous work heavily relies on **LMs** and, particularly their decoding mechanism to generate sets of sequences (Vijayakumar et al. 2016; Yuan et al. 2018; Meng et al. 2020; Holtzman et al. 2019; Ye et al. 2021). As **LM**

are initially trained on a large corpus to predict the next word, given previous words in some text (Radford et al. 2019) $P(w_t|w_{t-1}, \dots, w_0)$, they manage to learn the common pattern and word associations in natural languages. In this work, we propose to rely on a pre-trained LM to fine-tune it to complete sequences of queries. Following (Yuan et al. 2018), our objective is to fine-tune the pre-trained LM to generate sequences of query suggestions $Y = \{y_1, \dots, y_K\}$ given an input query x using the cross-entropy loss. Query suggestions are separated with a $\langle sep \rangle$ token in the sequence. In this work we use a decoder-only architecture, which means that the sequence can be written: $x \langle sep \rangle y_1 \langle sep \rangle \dots y_K$.

Despite various decoding tricks already used in the literature (Vijayakumar et al. 2016; Holtzman et al. 2019), generating set of sequences using LM decoders suffer from the lack of diversity within the generated set as new decoded sequences are not conditioned on other previously decoded sequences. We present what follows our strategy to control the diversity of generated query clarification set by using reinforcement learning.

Reinforcement Learning. To improve the diversity of generated suggestions, the language model is also fine-tuned using the proximal policy optimization (Proximal Policy Optimization (PPO)) (Schulman et al. 2017) algorithm to maximize the expected reward R . Given a state $s_t = (w_{t-1}, \dots, w_0)$ the policy predicts the next token w_t maximizing the expected reward:

$$\mathbb{E}_\pi(R) = \mathbb{E}_{x \sim D, Y \sim \pi(\cdot|x)}[R(x, Y)] \quad (5.1)$$

With R the reward function and π the policy, x the user query and $Y = \{y_1, \dots, y_K\}$ the set of query suggestions.

The value function (critic) used in PPO is initialized to the parameters of the reference model. However, the last linear layer is randomly initialized. For each state, this model predicts a value $V(s_t)$ used to compute an estimation of the advantage function $A = R(x, Y) - V(s_t)$. The critic is optimized to minimize the following loss function:

$$L_v = \sum_{t=0}^T |R(x, Y) - V(s_t)|^2 \quad (5.2)$$

Using PPO the policy objective is written as below:

$$L_{\pi} = \mathbb{E}_{\pi} \left[\min \left(A(w_t, s_t) \frac{\pi(w_t | s_t)}{\pi_{\text{old}}(w_t | s_t)}, \right. \right. \quad (5.3)$$

$$\left. \left. A(w_t, s_t) \text{clip} \left(1 - \epsilon, \frac{\pi(w_t | s_t)}{\pi_{\text{old}}(w_t | s_t)}, 1 + \epsilon \right) \right) \right]$$

Note that, PPO smooths the policy π update by constraining it to be close to the previous policy π_{old} by clipping the gradients.

In this work, we want to improve the diversity in the set generated by our supervised model. For this purpose, the reward function used in this work is composed of 2 parts:

- $r(x, Y)$: Following M. Liu et al. (2020), we compute the dissimilarity between generated queries in the set:

$$r = - \sum_{y_i \in \hat{Y}} \sum_{y_j \in \{\hat{Y} \setminus y_i\}} \text{sim}(y_i, y_j) \quad (5.4)$$

where sim is a similarity metric computed between queries described in Equation 6.12 .

- The second part ensures that the policy stays near the supervised model. For this, we follow (Stiennon et al. 2020), and add a KL term that penalizes the divergence between the policy π and the reference model π^{ref} . This KL regularization makes sure the policy stays grounded near a fixed reference model. This guarantees that generated sequences by the policy are not far from generated sequences from the reference model. Otherwise, our model would likely generate diverse but random sequences. The reward function can be written as:

$$R(x, Y) = r(x, Y) - \beta \log \left[\frac{\pi^{\text{RL}}(Y|x)}{\pi^{\text{ref}}(Y|x)} \right] \quad (5.5)$$

5.2.3 Model training and inference

The Figure 5.2 shows the training of CIRCLE, which is composed of 3 main steps:

1. Conditioned on the initial user query, the policy generates a trajectory, namely a sequence containing multiple queries. Queries are chained with $\langle \text{sep} \rangle$ token.

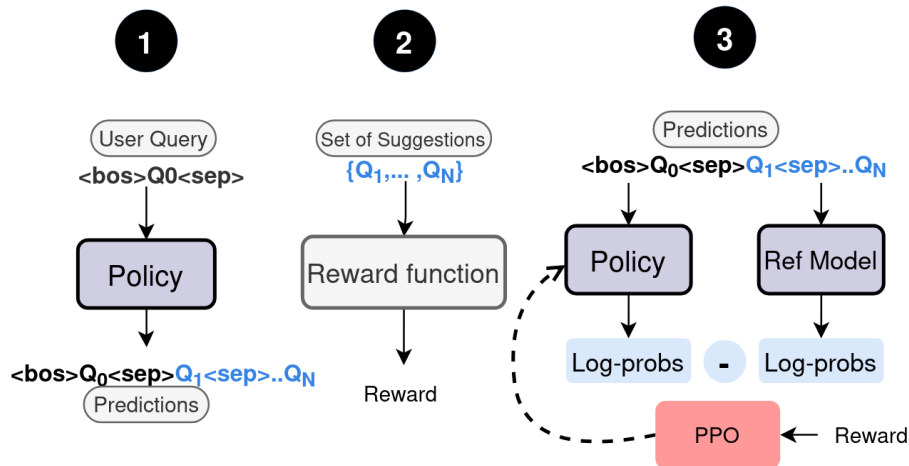


Figure 5.2 – Fine-tuning GPT2 with PPO

We control the number of generated queries using stop criteria counting the number of $\langle sep \rangle$ tokens.

2. The reward is computed using the Equation 5.5.
3. The generated sequence is fed into the reference model and the policy. Resulting log-probabilities and rewards are used to update the policy using PPO using the Equation 5.2 and Equation 5.3

During evaluation, the model is conditioned on the user query but also the suggestions selected by the user at each interaction turn. This means that at each step i the model completes the following sequence:

$$\text{input: } \langle bos \rangle x \langle sep \rangle y_1^+ \langle sep \rangle \dots y_i^+$$

with $q_{t=1}^+$ the selected query at step i and x the initial user query.

5.3 Evaluation protocol

The goal of this paper is to show that improving the diversity in the queries suggested to the user helps cover a wider range of topics and therefore better helps in the clarification process. To do so, the effectiveness of different models is evaluated in terms of how much they improve search performance in a multi-turn clarification framework. Therefore, we rely on classic Information Retrieval (IR) metrics to evaluate these models. The user's actions are simulated based on various hypotheses.

5.3.1 Dataset

The experiment is conducted on MS Marco 2020 passages (Nguyen et al. 2016b) which is an open domain dataset regrouping 8.8M passages and more than 500K Query-Passage relevance pairs. As R. Nogueira et al. (2019b), we trained our model on the train set of Msmacro and evaluated our model on a subset of the dev set (1200 queries sampled from 59 000).

5.3.2 Metrics

In accordance with the evaluation objectives, we rely on IR metrics to assess the effectiveness of the model in the multi-turn clarification framework but also to assess the diversity of proposed suggestions, we define different metrics.

- To measure the quality of the selected query clarification, we evaluate the effectiveness of document ranking issued from this query using the well-known Mean Reciprocal Rank (MRR) (Equation 3.4).

- To measure the similarity between document rankings at different iterations, we rely on the Rank-Biased Overlap (RBO) (Webber et al. 2010). This measures the similarity between incomplete and non-conjoint rankings and also values more heavily top-ranked documents. The more diverse the rankings are, the lower the score is.

$$\text{RBO}(S, T, p) = (1 - p) \sum_{d=1}^{\infty} p^{d-1} \cdot A_d \quad (5.6)$$

S, T are two document rankings, d is the actual depth of the ranking. A_d expresses the agreement (the size of the intersection of both rankings) at depth d : $A_d = \frac{|S_{:d} \cap T_{:d}|}{d}$. p determines the weight given the top-ranked document.

5.3.3 User Simulation

Because real human feedback is time-consuming and costly, the IR community often relies on user simulations to train or evaluate models (Erbacher et al. 2022c; Eckert et al. 1997; Komatani et al. 2005; Pietquin 2004; Schatzmann et al. 2006; Scheffler and Young 2000). We therefore design user simulation based on the hypothesis that users do not always fully cooperate with IR systems. For instance, a user may not know helpful suggestions. The goal is not to have realistic behavior

but to see the robustness and the limits of the proposed model. The proposed user simulation follows an epsilon greedy policy in which the user chooses a random action with a probability epsilon and chooses the best-known action with probability $1 - \epsilon$. ϵ enables a wide range of different possible user behaviors from the most cooperative user to the most random user.

It is worth noting that in our evaluation framework, the user does not have stop criteria to end the search session. This allows us to observe all possible scenarios and record performance at every interaction turn. Moreover, we do not take into account the position bias that would be induced by the position of the suggestions when presented to the user.

5.3.4 Baselines

We evaluated various methods to generate sets of query clarifications. These baselines are tested under different scenarios following the evaluation framework described in previous sections.

Google suggestions: Suggestions generated by Google search engine through their public API. Because Google suggestions only rely on the previously considered query. We mimic multi-turn interaction by iteratively switching the previous query with the selected query by the user. Depending on the queries' specificity, between 1 and 10 Google suggestions are proposed. We report the mean number of suggestions in Table 5.1. The suggestions also depend on the language and the geographic location. In this work, we use English for the United States. This can be considered as a strong baseline, thanks to its billions of user logs and feedback¹. However, we do acknowledge that the performance of these suggestions might be truncated because these suggestions are not specific to the MSMARCO corpus and might be biased by current trends.

Interact+Kmean: This is the multi turn model proposed by (Erbacher et al. 2022a). We evaluated this model using the K-mean selection mechanism as suggested by the authors, selecting the best 2 ranked queries from different clusters in the pool of 64 queries.

Beam search: We fine-tuned a GPT2 model to perform one-to-one reformulations. We use beam search to generate a set of query suggestions.

Supervised: We use a GPT2 fine-tuned using supervised learning to complete sequences of queries. This model was trained on Google Suggestions. This base-

1. How Google autocomplete works in Search: <https://blog.google/products/search/how-google-autocomplete-works-search/>

line is the supervised version of our CIRCLE model without the reinforcement learning fine-tuning.

CIRCLE: The overall version of our model, including the fine-tuning using Reinforcement Learning to leverage more diverse sets of query suggestions.

5.3.5 Implementation details

For the IR model, we opted for a Bert-base Dense retrieval model (Hofstätter et al. 2021). This model was trained on Msmarco-passages to maximize the dot product between queries and their associated relevant passages. We used this model to compute document embeddings. Embeddings are stored and indexed using Faiss HSWN32 index (Johnson et al. 2021). We used the pre-trained GPT2 provided by (Wolf et al. 2019b). The same model is used to compute query embeddings. For supervised learning, we used a learning rate of $lr = 2e^{-5}$ with a batch size of 128. We trained the model on 3 epochs. For finetuning with reinforcement learning we use a learning rate of $lr = 0.8e^{-6}$ with a batch size of 128. The exploration is constrained to $top_p = 0.9$ and $top_k = 20$. We use $\beta = 0.01$ and a clip ratio $\epsilon = 0.1$. The similarity metric used is the RBO Equation 6.12. We use greedy decode the circle model using greedy decoding. We used $p = 0.9$ in the RBO metrics. This means that the first ten documents are weighting for 85% of the overall score.

5.4 Results

In this section, we report results that were assessed by following the evaluation protocol described above.

5.4.1 Retrieval Performance

In the Table 5.1, we report IR scores of different approaches to generate sets of suggestions on a subset of the dev-set of Msmarco passages. We set the user to fully cooperative ($\epsilon = 0$). Models are evaluated using the MRR metric (Equation 3.4).

We can see that Google’s suggestions are a strong baseline for this task. With only an average of 5.64 query suggestions, it manages to reach an MRR of 0.4049.

	i=0 (No interaction)	i=1	i=2	i=3	i=4	i=5
User Query	0.2419	-	-	-	-	-
Google Suggestions $k = \sim(7.5/5.4/5.3/5.3/4.7)$	0.2419	0.3820	0.3997	0.4034	0.4043	0.4049
Beam Search $k = 2$	0.2419	0.2108	0.1872	0.1636	0.1536	0.1467
Beam Search $k = 4$	0.2419	0.3169	0.2686	0.2573	0.2457	0.2434
Beam Search $k = 8$	0.2419	0.3740	0.3469	0.3416	0.3295	0.3259
Interact+Kmean Erbacher et al. 2022a/ $k=2$ (Kmean 64)	0.2419	0.1990	0.2195	0.2370	0.2478	0.2551
Supervised $k = 2$	0.2419	0.2859	0.3017	0.3080	0.3074	0.3102
Supervised $k = 4$	0.2419	0.3030	0.3298	0.3350	0.3357	0.3354
Supervised $k = 8$	0.2419	0.3224	0.3614	0.3654	0.3649	0.3660
CIRCLE $k = 2$	0.2419	0.2824	0.3063	0.3174	0.3277	0.3290
CIRCLE $k = 4$	0.2419	0.3006	0.3250	0.3478	0.3614	0.3734
CIRCLE $k = 8$	0.2419	0.3244	0.3555	0.3737	0.3844	0.3889

Table 5.1 – **Effectiveness of CIRCLE methods.** Effectiveness results on the subset of MS Marco passage 2020 dev set (1200 queries - 1 relevant document per query), k the size the set of suggestions for interaction. Metric used is Mean Marginal Relevance

For $k = 8$, the beam search managed to reach an MRR of 0.3740 in the first interaction. However, the score decreased with each subsequent interaction. This can be explained because the model takes as input only the last selected query and therefore the model generation can diverge if generated queries are less efficient than in the previous turn and because the user always selects one of the generated queries. CIRCLE managed to improve the mean score from 0.3244 in the first interaction to 0.3889 in the last interaction. Against 0.3224 to 0.3660 for the supervised model. This means that the additional Reinforcement Learning (RL) fine-tuning was beneficial to improve the overall performance.

5.4.2 Qualitative example

Table 5.2 show sets of examples generated by Google suggestions, Supervised model, and Circle model.

Turn	Query	Google Suggestions	Supervised 4	Circle 4
0	what is randomness	what is randomness in statistics	what is randomness definition	what is randomness definition
		what is randomness in probability	what is randomness meaning	what is randomness in statistics
		what is randomness in error	what is randomness in statistics	what is randomness error
		what is randomness in chemistry	what is randomness error	what is randomness bias
1	what is randomness error	what is randomness error in measurement	what is randomness error	what is randomness error definition
		what is randomness error in decision making	what is randomness error in statistics	what is randomness error synonyms
		what is randomness error in physics	what is randomness error definition	what is randomness error in statistics
		what is randomness error in chemistry	what is randomness error meaning	what is randomness error bias

Table 5.2 – **Qualitative Examples against Google suggestions.** Table showing sets of suggestions for 2 consecutive queries. For this example, to better compare how the selected query affects the new sets of suggestions across models we forced the user to select the same suggestions for all models.

5.4.3 Suggestions Coverage

In this section, we analyze the coverage of proposed suggestions models. To analyze this, we compute the mean similarity between ranking associated to each suggestion using the RBO metric (Equation 6.12).

	Google Suggestions k = 5.64	Beam search k=8	Supervised k=8	CIRCLE k=8
Mean RBO	0.3243	0.3610	0.3998	0.3604

Table 5.3 – **Ranking Diversity between turns (RBO)**. This table shows the mean ranking similarity between different model suggestions. The metric used is the RBO. The more diverse the rankings are, the lower the score is.

The Table 5.3 shows that Google suggestions have good coverage of possible user needs with a mean RBO of 0.3243 while proposing only an average of 5.64 query suggestions. Surprisingly, the beam search also has good coverage with an RBO of 0.3610. We can see that the RL finetuning improves the mean RBO between the supervised and CIRCLE. CIRCLE and the beam search have sensibly the same RBO.

5.4.4 Impact of user cooperation

In this section, we assess the performance of Google suggestions and CIRCLE interacting with non-perfect users. The objective is to observe the robustness of models with variable feedback quality. The user is following a greedy epsilon policy with epsilon ranging from 0.0 to 0.5.

We can see in Figure 5.3b that none of the proposed models is robust to non-cooperative users. We observe that our model is a bit more resilient with a non-fully cooperative user. This is explained because our model takes as input all the sequence of previously selected queries including the initial user query. Whereas Google suggestions and Beam search only consider the last selected query as input. This means that when the user selects a query not-relevant for his/her information need, the newly generated suggestions diverge from the initial user query.

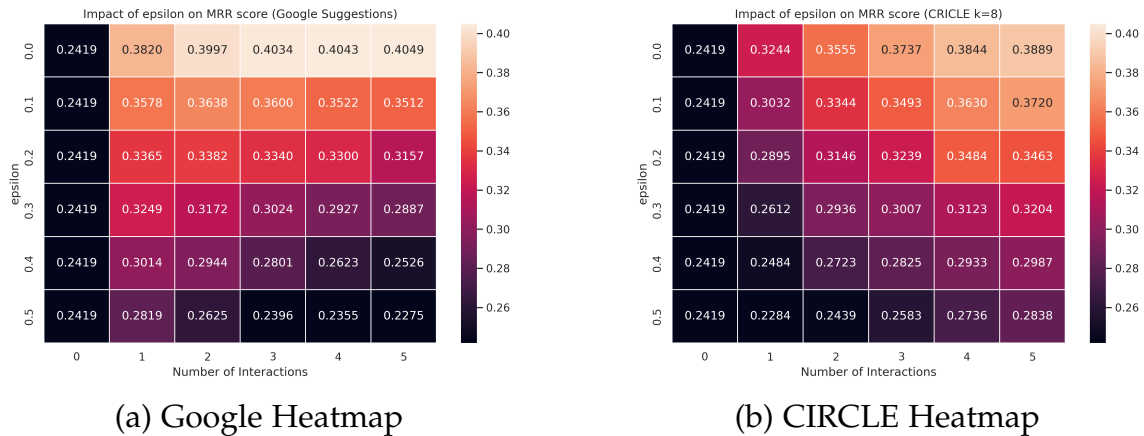


Figure 5.3 – **Google and CIRCLE coop/turns HeatMap**. Impact of the interactions with various user hypotheses on the system efficiency. The observed score is the MRR@1000.

5.5 Conclusion and perspectives

In this work, we consider the multi-turn interactive query clarification problem for IR. We propose CIRCLE, a GPT2 model that generates a sequence of query suggestions conditionally to one or several queries. By using reinforcement learning, we manage to increase the set diversity compared to the supervised model. Additionally, this model completes sequences composed of past feedback (selected queries) to generate new sets of queries iteratively, showing the benefit of multi-turn clarification. Especially against non-cooperative user simulations, relative to Google suggestions. However, there are several downsides: the proposed model is far from Google suggestions performances, and it requires several interactions with the user to perform better than the beam search. Additionally, the proposal model does not embed strategy at the interaction level and was not optimized on any user behavior or IR metrics. There are several ways this work can be extended.

1) The policy can be trained to maximize a utility function with multi-turn interactions with the user. The quality of interactions between user/system could benefit strongly. However, the main difficulties are that the rewards are very sparse, and incorporating interaction feedback in the model is challenging during the training.

2) The performance of the model may be improved by using a loss invariant to query permutation during supervised training. Note that we tried to reproduce the method in (Ye et al. 2021) and apply it with a pre-trained LM on the

query suggestions task. However, we were not able to control the language model generation with the control code without deteriorating the generation quality.

3) The model can be extended to leverage search session or multi-faceted information needs. Generated suggestions based on multiple user information needs and therefore multiple queries require user simulations that can simulate complete search session (Câmara et al. 2022).

Part III

DATASET AUGMENTATION FOR MIXED-INITIATIVE CONVERSATIONAL SEARCH

AUGMENTING AD-HOC IR DATASET FOR MIXED-INITIATIVE CONVERSATIONAL SEARCH

Chapter abstract

The work in this chapter, done in collaboration with Philippe Preux, Jian-Yun Nie and Laure Soulier, was published at CORIA-TALN 2023 and will be sent to a journal.

- Pierre Erbacher, Jian-Yun Nie, Philippe Preux, and Laure Soulier (2023b). *Augmenting Ad-Hoc IR Dataset for Interactive Conversational Search*. arXiv: 2311.06119 [cs.IR].

Contents

6.1	Introduction	80
6.2	Simulated interactions	81
6.2.1	Problem definition	81
6.2.2	Extracting Facets	82
6.2.3	Generating Mixed-Initiative Interactions	84
6.2.4	Leveraging Mixed-Initiative Interactions to Adapt Ad-Hoc IR Datasets	85
6.3	Assessing the Quality of the Dataset Generation Methodology	88
6.3.1	Evaluation protocol	88
6.3.2	Evaluation of the generated interactions	92
6.4	Evaluation on IR Task	93
6.4.1	Neural Ranking Model Leveraging Mixed-Initiative Interactions	94
6.4.2	Training details	94
6.4.3	Metrics and Baselines	95
6.4.4	Effectiveness of mixed-initiative-oriented neural ranking	96
6.5	Complementary Experiments	96
6.5.1	Extension to Multi-Turn Interactions	96
6.5.2	Transferability of the Methodology	99
6.5.3	Additional Analysis: Ranking similarity	100
6.6	Conclusion and discussion	101

6.1 Introduction

In previous chapters interactions with users remained simple and consisted of multiple suggestions of query reformulations. The retrieval performances are improved only by using a more optimal reformulation of the initial query and interaction were not as conversational as expected for Conversational Search (CS). In this chapter we address this limitation, we explore how interactions with users can directly improve IR systems in CS and how mixed-initiative interactions can serve document retrieval. To this end, we aim to leverage simulation techniques to automatically generate mixed-initiative interactions between a user and a system and propose a methodology to augment ad-hoc Information Retrieval (IR) datasets with such interactions. To do so, we design a query clarification generator leveraging the ClariQ dataset as well as a user simulation for the user’s response. We use them to generate mixed-initiative interactions on the MS MARCO ad-hoc IR dataset. Our contribution is threefold:

- We propose a methodology to augment ad-hoc IR datasets to integrate mixed-initiative interactions;
- We evaluate our dataset-building methodology, and particularly the quality of mixed-initiative interactions generated for the ad-hoc IR dataset MS MARCO;
- We demonstrate the utility of simulated interactions for document ranking on augmented MS MARCO. This result can also be seen as an evaluation proxy of the usefulness and the relevance of the simulated mixed-initiative interactions within the ad-hoc IR dataset.

6.2 Simulated interactions

6.2.1 Problem definition

We introduce our methodology to automatically generate large-scale mixed-initiative-driven IR datasets. To do so, we propose to augment ad-hoc IR datasets with simulated user-system interactions, namely clarifying questions (for the system side) and the corresponding answers (for the user side). To provide a dataset useful for training mixed-initiative-oriented neural ranking models and capturing similarity signals in the matching loss, it is important to provide a wide range of interactions, namely clarifying questions that give rise to either positive or negative answers. Having in mind that a topic might be complex or ambiguous, we follow previous works (Sekulić et al. 2021; Zamani et al. 2020a; Salle et al. 2021) leveraging facets to generate those clarifying questions. Extracting positive or negative facets around a topic can be seen as a proxy to constrain the generation of clarifying questions expecting ‘yes’ and ‘no’ answers. Moreover, to ensure the overall quality of the mixed-initiative interactions, we propose to introduce another constraint variable modeling the user’s search intent. The pair of facet and intent variables allows generating positive and negative clarifying questions (thanks to the facet) by always keeping the answer generation coherent with the relevant judgments in the initial dataset (thanks to the intent).

Said otherwise, sampling different facet-intent pairs from passages with known relevance judgment allows building a dataset with positive and negative mixed-initiative interactions that reflect the search intent of the user.

For the sake of simplicity, we only consider single-turn interactions, and discuss the extension to multi-turn interactions in [Section 6.5](#).

Let us consider an ad-hoc IR dataset $\mathcal{D} = \{\mathcal{P}, \mathcal{Q}, \mathcal{R}\}$, in which \mathcal{P} is a collection of passages (or documents), \mathcal{Q} is a set of queries, and \mathcal{R} is a set of relevance judgments. \mathcal{R} includes tuples $(q, \mathcal{P}_q^+, \mathcal{P}_q^-)$ indicating relevant $\mathcal{P}_q^+ \subset \mathcal{P}$ and irrelevant passages $\mathcal{P}_q^- \subset \mathcal{P}$, for a query $q \in \mathcal{Q}$. We assume $\mathcal{P}_q^- \cap \mathcal{P}_q^+ = \emptyset$. Our objective is to augment this dataset \mathcal{D} with a mixed-initiative interaction set $X = \{X_1, \dots, X_i, \dots, X_n\}$. We note a mixed-initiative interaction $X_i = (q, cq, a)$ where q refers to an initial query, cq a clarifying question, and a the associated answer. With this in mind, we design a dataset-building methodology $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{D} \cup \{X_1, \dots, X_i, \dots, X_n\}$ relying on two main steps: 1) extracting the (positive and negative) facets f related to each topic (if not available in the initial ad-hoc IR dataset) which are then used to constrain the clarifying question generation, and 2) generating mixed-initiative interactions given a query q and this facet f . Depending on the dataset, the sets of positive \mathcal{F}^+ and negative \mathcal{F}^- facets associated with query q might be available or need to be built (Section 6.2.2).

We also assume that the search intent int of the user is characterized by the relevant documents available in the initial dataset. Then we propose to generate a mixed-initiative interaction X_i given a query q and the constraint variables f and int .

We rely on 1) a clarifying model $\mathcal{CM} : q, f \rightarrow cq$ aiming at generating a clarifying question cq around facet f given the query q , and 2) a user simulation $\mathcal{US} : (cq, int, f) \rightarrow a$ which infers the answer a in response to the clarifying question cq given the relevance of the facet f and the user intent int .

Below, we introduce the method to extract facets, as well as the components enabling to generate clarifying questions and answers. Then, we present the overall methodology to augment ad-hoc IR datasets with mixed-initiative interactions. Examples of expected mixed-initiative interactions are presented in Table 6.1.

6.2.2 Extracting Facets

Facets might be explicit or implicit depending on the dataset. For example, they are specified in TREC Web 2009-12 (Clarke et al. 2009b), and accordingly, Qulac and ClariQ (Over 2001). If not explicitly specified, we propose to extract them from documents. Previous works have shown that query facets can be extracted from top-retrieved documents (Dou et al. 2016; Kong and Allan 2013). Inspired by the analysis provided by Sekulić et al. (2022b), we extract top contextual keywords to represent facets, as suggested in (Sharma and Y. Li 2019). The goal of the facet extraction is to provide additional keywords that can be used to later generate

Simulated mixed-initiative interactions in MiMarco

Query: why did the us voluntarily enter ww1

System: do you want to know the effect of unrestricted submarine warfare on the us ?

Answer: yes

Intent: The effects of unrestricted submarine (U-Boat) warfare, economic ties to the Allies, the public outrage to the sinking of the Lusitania, the Sussex torpedoing, and the Zimmerman Telegram caused this change, and the United States entered an Allied Alliance and a World War for the first time in its history. **Facet:** effects unrestricted submarine boat warfare

Query: find phone number for irs

System: are you looking for phone numbers for the irs internal revenue service?

Answer: yes

Intent: Internal Revenue Service Phone Numbers. 800-829-1040 For individual and joint filers who need procedural or tax law information and/ or help to file their 1040-type IRS Tax Help Line for individual returns (including Individuals | Schedules C and E); and, general account information for Form 1040 Filers. **Facet:** internal revenue service phone numbers

Query: lps laws definition

System: are you looking for a definition of a limited partnership?

Answer: no

Intent: The Court will not let you establish an LPS conservatorship unless it finds beyond a reasonable doubt, that the mentally ill person, is gravely disabled. Gravely disabled means that, because of a mental disorder, the person cannot take care of his/her basic, personal needs for food, clothing, or shelter. **Facet:** limited partnership business

Table 6.1 – **Example of simulated interactions of MiMarco.** In the first and second examples, both the intent and the facet are sampled from a relevant passage. In the third example, the intent is sampled from a relevant passage but the clarifying question refers to a negative topic facet.

a clarifying question about various topics or subtopics. In this work, facets are a set of keywords providing additional context to the query. We formulate it as a bijective function $\psi(P) \rightarrow \mathcal{F}$ that maps a set P of passages to a set of facets. Given a query q , we construct the sets \mathcal{F}^+ and \mathcal{F}^- of positive and negative facets from respectively relevant and irrelevant passage sets, resp. \mathcal{P}_q^+ and \mathcal{P}_q^- . This allows us to keep the relevance of facets. To do so, for a passage $p \in (\mathcal{P}_q^+ \cup \mathcal{P}_q^-)$, we extract as a facet $f \in \mathcal{F}$ the set of K words in the passage that are the most similar to the passage embedding (i.e., the embedding of the [CLS] token). To compute the similarity, we use a pre-trained Sentence-Bert (i.e., MiniLM-L6-v2

model) (Reimers and Gurevych 2019) between each token embedding and the passage one.

6.2.3 Generating Mixed-Initiative Interactions

6.2.3.1 Generating clarifying questions

The goal of the clarifying model \mathcal{CM} is to ask relevant clarifying questions relating to an ambiguity in the meaning or the object of the query. In most of the proposed models (Zamani et al. 2020a; Sekulić et al. 2022b; Sekulić et al. 2021; Salle et al. 2021; Aliannejadi et al. 2019), this ambiguity is addressed by using the concept of facet. Therefore, the generation of clarifying questions cq is conditioned on the initial query q and a facet f :

$$p(cq|q, f) = \prod_i p(cq_i|cq_{<i}, q, f) \quad (6.1)$$

where q_i is the i^{th} token in the sequence and $q_{<i}$ the previously decoded tokens. Our clarifying question generation model is based on a pre-trained sequence-to-sequence model which is fine-tuned to generate a clarifying question cq given the following input sequence:

$$\text{Query: } q \text{ Facet: } f \quad (6.2)$$

where $Query$: and $Facet$: are sequences of tokens used to distinguish query and facet in the attention.

In this paper, we limit the clarifying questions to those that expect yes/no answers.

6.2.3.2 User Simulation

The goal of the user simulation US is to mimic the user’s answer in response to a clarifying question given his/her intent. In the user simulation, we expect accurate answers to clarifying questions, giving useful feedback to help the system understand his/her intent. The intent is a representation of the information needs or the goal behind the initial query. It is used to constrain the user simulation’s answer towards this goal (Kang et al. 2019; C. Gao et al. 2022; F. Wu et al. 2020; K. Zhou et al. 2020; Fu et al. 2020; Erbacher et al. 2022a). While sophisticated user simulations have been proposed to exhibit various types of behaviors like

cooperativeness or patience (Salle et al. 2021), we limit the clarifying question to ask if the intent is about a facet and the answer of the user simulation to ‘yes’ or ‘no’ answer.

This limited form of answer is motivated by two reasons: (1) despite the simplicity, a correct answer of this form corresponds to basic realistic interactions with users and is highly useful for the system to better identify the intent behind the query. (2) This simple form of question and answer is easier to generate and evaluate. As an initial attempt, we prefer to start with this simple setting.

More formally, the user simulation aims at estimating the probability of an answer $a \in \{yes, no\}$ given a query q , a search intent int , and a clarifying question:

$$p(a|q, int, cq) \tag{6.3}$$

This is implemented as a sequence-to-sequence model that encodes the following input:

$$\text{Query: } q \text{ Intent: } int \text{ Question: } cq \tag{6.4}$$

and generates a ‘yes’/‘no’ answer.

Intent modeling The user’s intent corresponds to the user’s information need and is only known by the user. While multiple intent representations can be adopted (such as a detailed description of the information need (Aliannejadi et al. 2019; Aliannejadi et al. 2021), a vector representation (Erbacher et al. 2022a) or constraints (Kang et al. 2019; C. Gao et al. 2022; F. Wu et al. 2020; K. Zhou et al. 2020; Fu et al. 2020)), IR datasets usually do not have annotated intent associated with the query. However, relevant passages are known in an IR dataset. In this paper, we use a sampled relevant passage $p \in \mathcal{P}_q^+$ and assimilate its content to the underlying intent int . Formally: $int \leftarrow p$. We acknowledge that this choice relies on a strong hypothesis and we discuss it in Section 6.6.

6.2.4 Leveraging Mixed-Initiative Interactions to Adapt Ad-Hoc IR Datasets

Given an ad-hoc IR dataset \mathcal{D} , our objective is to augment \mathcal{D} with mixed-initiative conversations X . It is worth distinguishing the creation of training and testing datasets since they have different purposes. The training set requires including positive and negative interactions to allow the community to train properly mixed-initiative IR-oriented neural models. As a reminder, those posi-

Algorithm 6.1 Offline methodology. Methodology for building Mixed-Initiative IR dataset

```

Require:  $\mathcal{D} = \{\mathcal{P}, \mathcal{Q}, \mathcal{R}\}$ 
 $X \leftarrow \{\}$  ▷ Set of mixed-initiative IR-oriented interactions
for  $q \in \mathcal{Q}$  do
   $\mathcal{F}^+ \leftarrow \psi(\mathcal{P}_q^+)$  ▷ Extract the positive facets
   $\mathcal{F}^- \leftarrow \psi(\mathcal{P}_q^-)$  ▷ Extract the negative facets
  for  $f \in (\mathcal{F}^+ \cup \mathcal{F}^-)$  do
     $cq \leftarrow \mathcal{CM}(q, f)$  ▷ Generate the clarifying question
    if  $f \in \mathcal{F}_q^+$  then ▷ Building the answer
       $a \leftarrow \text{'yes'}$ 
    else
       $a \leftarrow \text{'no'}$ 
    end if
     $X_i = (q, cq, a)$ 
     $X \leftarrow X \uplus X_i$  ▷ Increment the interaction set
  end for
end for
return  $\mathcal{D} \cup X$ 

```

tive/negative interactions are built based on relevant and irrelevant documents determining positive and negative facets. Using the same heuristics to generate a testing dataset is not suitable since it would imply including relevance judgments as evidence sources of the clarifying question generation at the inference step. Therefore, we propose to design an online evaluation methodology, leveraging the clarifying model \mathcal{CM} and the user simulation \mathcal{US} to generate mixed-initiative interactions without introducing a bias related to relevance judgments. We present these two methodologies aiming at generating offline and online datasets in what follows.

6.2.4.1 Building an offline training dataset with relevance judgments

Our offline methodology aims at generating a wide range of positive and negative mixed-initiative interactions based on an ad-hoc IR dataset. To do so, we use relevant/irrelevant documents to build positive/negative facets constraining the clarifying question generation. As a supplementary quality constraint in the dataset supervision, we would like to ensure that answers fit with the relevance of the used documents. Said otherwise, the user simulation presented

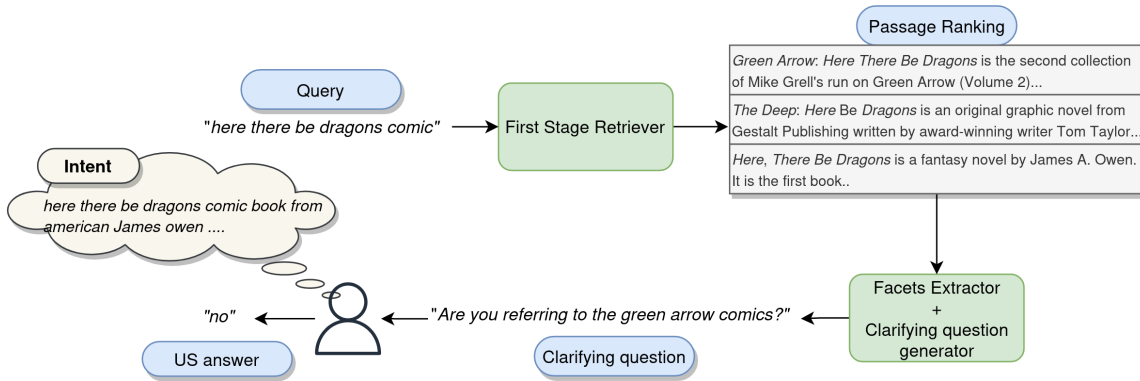


Figure 6.1 – **Online evaluation pipeline.** Evaluation pipeline to create mixed-initiative interactions on a test ad-hoc IR set.

in Section 6.2.3.2 is replaced by a simple heuristic matching answers a with the relevance of facets f :

$$a = \begin{cases} 'yes' & \text{if } f \in \mathcal{F}^+ \\ 'no' & \text{otherwise} \end{cases} \quad (6.5)$$

We propose the 3-step pipeline presented in Algorithm 6.1. Given a query q : 1) positive and negative facets, resp. \mathcal{F}^+ and \mathcal{F}^- , are extracted from relevant and non-relevant passage sets, resp. \mathcal{P}_q^+ and \mathcal{P}_q^- ; 2) a mixed-initiative interaction X_i is issued for a facet f , generating the associated clarifying question cq (with \mathcal{CM}) and associating answer a with the facet relevance (Equation 6.5); 3) the interaction set X is incremented with this new interaction X_i , allowing to build a mixed-initiative IR dataset by associating the interaction set X built over all queries with the initial ad-hoc IR dataset \mathcal{D} .

6.2.4.2 Building a testing dataset for online evaluation without relevance judgments

Our online methodology aims at generating mixed-initiative interactions without relying on relevant/irrelevant documents. Instead, we leverage pseudo-relevance feedback by using SERPs of a first-stage ranking model as a proxy to extract query facets. Each facet conditions the generation of the clarifying question and the answer. More particularly, the proposed pipeline to generate online mixed-initiative interactions for a query q is presented in Figure 6.1. It is built on the following steps: 1) ranking documents using a first-stage ranker (in our case BM25), 2) extracting the set of facets based on pseudo-relevant/pseudo-irrelevant documents, and 3) generating the mixed-interactive interaction.

Depending on the evaluation needs, different choices can be made regarding facet extraction. One can extract a single facet from the top-retrieved document to perform a single retrieval step for a query (the strategy used in our experiments). Other tasks or evaluation objectives would require generating multiple facets, and accordingly, multiple mixed-initiative interactions. This can be done by identifying top/flop documents obtained with the first-stage ranking as pseudo-relevant/irrelevant documents; each document conditioning the facet extraction as described in Section 6.2.2.

6.3 Assessing the Quality of the Dataset Generation Methodology

In this section, we evaluate our methodology, and particularly, the quality of simulated interactions. Please note that we focus on augmenting the MS MARCO dataset but our methodology is generalizable to any ad-hoc IR datasets.

6.3.1 Evaluation protocol

6.3.1.1 Datasets

We focus here on the MS MARCO 2021 passages dataset (Nguyen et al. 2016b) which is an open-domain IR dataset containing 8.8M passages and more than 500K Query-Passage relevance pairs with approximately 1.1 relevant passages per query on average. MS MARCO is commonly used to train and evaluate first-stage retriever and cross-encoder architectures (Thakur et al. 2021). We leverage the MS MARCO passage dataset with mined Hard Negatives released by sentence-transformers (Reimers and Gurevych 2019)¹ to build our tuples $(q, \mathcal{P}^+, \mathcal{P}^-)$. Hard Negatives are passages retrieved using a state-of-the-art retrieval method, which is more closely related to the query. They allow us to generate more relevant questions and answers.

To train the clarifying model \mathcal{CM} , we use the filtered version of the ClariQ dataset proposed in (Sekulić et al. 2021) that maps clarifying questions with facets. All clarifying questions in this dataset are built to expect 'yes'/'no' answers. This

1. <https://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives>

	Train set	Test set
Number of documents	8M	8M
Number of queries	500K	6980
Avg number of interactions per query	38.5	-
Avg length of clarifying questions	11.0	-
Percentage of positive answers	26.7%	-
Percentage of negative answers	73.3 %	-

Table 6.2 – **Statistics of MiMarco.** Statistics of the generated mixed-initiative IR dataset MIMarco.

dataset provides 1756 supervised tuples of (query-facet-clarifying questions) for 187 queries.

To train the user simulation \mathcal{US} , we do not use the answers included in the ClariQ dataset for supervision since answers are verbose (sentences with detailed information). Therefore, we leverage half of the train set of the MsMarco dataset (250000 queries) to extract positive and negative facets as detailed in Section 6.2.2 and generate clarifying questions using the \mathcal{CM} model. The supervision label related to answers is inferred as proposed in the offline evaluation (see Equation 6.5).

To build a final dataset including training and testing sets, we respectively apply the offline evaluation methodology (Algorithm 6.1) on the other half of the training set (not used to train the user simulation) and the online evaluation methodology (Figure 6.1) on the test set of the MS MARCO dataset. For the offline evaluation, because the original dataset includes sparse annotations, i.e. some passages are actually relevant but not annotated as such, it might be possible that relevant documents are considered irrelevant. This trend is however exhibited in the MS MARCO train set which only includes one relevant document by query. Therefore, to ensure labeling consistency, we follow (Y. Qu et al. 2021) and denoise hard-negative in the training set using a well-trained cross-encoder model² that captures similarities between passages.

For the online evaluation, we choose to generate a single mixed-initiative interaction based on the top-retrieved document to fit with our extrinsic evaluation task based on IR.

Statistics of the obtained mixed-initiative IR dataset, called MIMarco, are presented in Table 6.2. Table 6.1 depicts some examples of simulated conversations generated from MS MARCO queries.

2. <https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2>

6.3.1.2 Baselines and metrics

Evaluating clarifying questions with automatic metrics. We follow Sekulić et al. (2021) and compare our clarifying model, denoted \mathcal{CM} , with 1) a template-based approach (*Template*). The template follows a predefined sequence concatenating facets: 'Are you looking for + Facet'. 2) $\mathcal{CMw/oFacet}$: the version of our \mathcal{CM} model only conditioned on the query. This is based on a T5 model trained as a machine translation model, which generates a clarifying question from the query only.

We evaluate the ability of \mathcal{CM} to generate clarifying questions using references provided in the ClariQ test set. We consider the METEOR metric (Banerjee and Lavie 2005) and the average cosine similarity between sentence embeddings (COSIM). METEOR is commonly used to evaluate machine translation output considering unigram recall and precision. At the sentence level, this has a good correlation with human judgments (Banerjee and Lavie 2005). To compute the similarity score, we encode the questions using a well-trained MiniLM-L6-v2 (Reimers and Gurevych 2019). We use t-test to assess the significance of metric differences (***: p-value<0.005).

To evaluate if the generated questions on MS MARCO are similar to their relative passage, we also compute the mean cosine similarity between clarifying questions and their retrieved relevant and non-relevant passages. We encode the questions using MiniLM-L6-v2 (Reimers and Gurevych 2019).

Human evaluations on clarifying questions. To compare and better assess the quality of a generated clarifying question on MS MARCO, we performed a human evaluation. Given the initial user query and the passage used to generate the question, we ask human annotators to evaluate the quality of 200 sampled clarifying questions among the three models (*Template*, $\mathcal{CMw/oFacets}$, and our \mathcal{CM} model). To do so, annotators are asked to select a preferred clarifying question from the three suggestions displayed in a shuffled order for the following criteria:

- **Usefulness:** Assess if a question can help to better understand or refine the query by providing additional information or suggestions.
- **Naturalness:** Assess the question fluency and readability.
- **Relevance:** Assess whether a question is specific or related to the information contained in a passage.

Each annotator evaluated 20 cases and, for each metric identified the best model output. We recruited 10 evaluators. Each instance is evaluated by 2 annotators and we obtain a Kappa metric equal to 0.324 showing a fair agreement between evaluators. We also distinguished results for both positive and negative user answers by sampling relevant and irrelevant facets.

Human evaluations on answers. A strong hypothesis in our method is that clarifying questions generated with facets extracted from relevant passages leads to positive answers while using irrelevant passages to generate negative facets intrinsically leads to negative answers. To validate this strong hypothesis, we have shown human evaluators different instances including a query q , a clarifying question cq , and the relevant passage p used to build the facet f . For each instance, we asked human evaluators to answer with 'yes' or 'no' to clarifying questions. This human evaluation involves 10 human annotators for a total of 200 questions, with balanced relevant and non-relevant facets used to generate the clarifying question. Each instance is annotated by 2 humans. We obtain a Kappa metric equal to 0.472 showing a moderate agreement between evaluators. To validate our hypothesis, we set human answers as the reference, and we compare them with our auto-labeling method (namely, the user simulation US) to calculate the accuracy metric.

6.3.1.3 Implementation details

For both CM and US , we used the pre-trained T5 checkpoint available on the Huggingface hub (Raffel et al. 2020b; Wolf et al. 2019a). To finetune these two models, we used teacher forcing (Williams and Zipser 1989b) and a cross-entropy loss. For optimization, we use AdaFactor (Shazeer and Stern 2018), weight decay, and a learning rate of $5 \cdot 10^{-5}$ with a batch size of 64. Keyword embeddings are computed using an off-the-shelf pre-trained MiniLM-L6-v2 model (Reimers and Gurevych 2019). The number of extracted words is fixed to $k = 5$ for the overall experiments. For inference, we use nucleus sampling ($p=0.95$) for the CM and US models.

	METEOR	COSIM
<i>Template</i>	0.338***	0.643***
<i>CMw/oFacet</i>	0.326***	0.608***
<i>CM</i>	0.557	0.812

Table 6.3 – **Evaluation of different clarifying models.** Evaluation of different clarifying models on the test set of ClariQ dataset. Significance two-sided t-test: *** indicates statistically significant difference between baselines and our *CM* model ($p < 0.0015$)

	q	p+	p-
cq+	0.675	0.721	0.450
cq-	0.521	0.450	0.685

Table 6.4 – **Clarifying Question/Passages Cosine Similarity.** Mean cosine similarity between generated clarifying questions and their related passages on the train set. The *cq+*, *cq-* denote respectively clarifying questions generated using positive and negative facets.

6.3.2 Evaluation of the generated interactions

6.3.2.1 Automatic evaluation

Table 6.3 reports the effectiveness of the clarifying model on the ClariQ test set. Results show that our model *CM* significantly outperforms all baselines. The lower results obtained by the *CMw/oFacet* baseline highlight that a simple machine translation model is less effective than templates using facet terms. Facets are useful to constrain the clarifying model, and seq-to-seq models based on Large Language Model (LLM)s are more natural than template-based methods. Facets are extracted from a relevant or irrelevant passage and used to generate clarifying questions. Table 6.4 reports the cosine similarity between embeddings of questions and respective passages. We observe that the similarity between clarifying questions and their related passages (in bold) is higher than that between the clarifying questions and the queries. This shows that the generated questions are not generic to the query but oriented toward the provided passages.

6.3.2.2 Human Evaluation

We report human evaluation of clarifying questions in Table Table 6.5. The *CMw/oFacet* fine-tuned without facet generates more natural questions than other baselines (preferred for 46.3% of the sample). The *CM* model fine-tuned

	Answer	Naturalness	Usefulness	Relevance
Template	positive	0.044	0.086	0.120
	negative	0.073	0.095	0.146
	total	0.119	0.181	0.267
$\mathcal{CMw/oFacet}$	positive	0.243	0.195	0.077
	negative	0.220	0.140	0.056
	total	0.463	0.336	0.133
\mathcal{CM}	positive	0.206	0.213	0.297
	negative	0.211	0.268	0.301
	total	0.417	0.481	0.599

Table 6.5 – **Results of the human evaluation.** The table shows the results of the human evaluation on MS MARCO-passage. The \mathcal{CM} without facet produces more natural questions, however not as relevant as \mathcal{CM} with facet.

with facet generates more useful and relevant questions, this model is considered as the more relevant by evaluators in 59.9% of the test sample. This shows that the retrieved facet in the generation helps generate more useful and relevant questions.

In the human evaluation of answers, we obtain an accuracy of 0.685 between human answers and automatic labeling of clarifying questions. There are multiple causes explaining the difference between human answers and auto-labeling. 1) Facet may not always capture correctly the information provided in a passage, leading to poor clarifying questions. 2) The \mathcal{CM} model does not always generate a question oriented toward the provided facet and produce a reformulation of the initial query, therefore asking a question not related to a facet.

6.4 Evaluation on IR Task

In this section, we propose to assess indirectly the quality of the generated dataset through an IR task. Indeed, previous works (C. Qu et al. 2020; K. Zhou et al. 2020; R. Li et al. 2018; Fu et al. 2020; Jia et al. 2022) have already used extrinsic tasks to validate a dataset. Therefore, we introduce a neural ranking model that estimates passage relevance scores based on the query and a mixed-initiative interaction. Our objective is twofold: 1) Applying this model to our generated dataset provides some insights on whether the clarifying question and the associated answer actually give useful feedback to better understand the underlying information need. The evaluation is based on the following assumption: if a rank-

ing model using the generated interactions outperforms the one without them, the interactions are deemed relevant and useful. 2) We provide a first baseline for mixed-initiative IR tasks.

6.4.1 Neural Ranking Model Leveraging Mixed-Initiative Interactions

We propose a simple model based on a cross-encoder architecture that is effective for IR task, especially when using LLMs (Pradeep et al. 2021).

Typical cross-encoder aims at predicting the relevance of a passage p given a query q $P(\text{relevant} = 1|q, p)$. Our model estimates a score for passages based on the query, a clarifying question, and a user answer (q, cq, a) , i.e.

$$p(\text{relevant} = 1|p, q, cq, a) \quad (6.6)$$

Following (Pradeep et al. 2021), the above score is transformed to the log-probability of predicting (decoding) the true/false tokens, i.e.

$$s_p = \log p(\text{true}|q, p, cq, a) \quad (6.7)$$

Following (Pradeep et al. 2021), we use the MonoT5 model and integrate mixed-initiative interactions to estimate document scores. The input sequence is a concatenation of query, document, question, and answer separated by special tokens:

$$\text{Query: } q \text{ Document: } d \text{ Question: } cq \text{ Answer: } a \quad (6.8)$$

6.4.2 Training details

We used a pre-trained MonoT5 checkpoint available on the Huggingface hub (Raffel et al. 2020b; Wolf et al. 2019a). We fine-tune this model on our train set in 1 epoch, using our methodology with teacher forcing and a cross-entropy loss. We consider a maximum sequence length of 512 and a batch size of 128 sequences. To properly learn to contrast between relevant and non-relevant passages given a question, we use in-batch negative answers.

For optimization, we use AdaFactor (Shazeer and Stern 2018), weight decay, and a learning rate of 10^{-4} . The model fine-tuning takes approximately 4 hours on 4 RTX 3080 (24 Go).

	MRR@10	NDCG@1	NDCG@3	NDCG@10
BM25	0.1840***	0.105***	0.1690***	0.228***
BM25 + RM3	0.1566***	0.0807***	0.1386***	0.2021***
BM25 + MonoT5	0.3522***	0.2398***	0.3457***	0.4034***
BM25 + CLART5	0.3863	0.2788	0.3817	0.4327

Table 6.6 – **IR effectiveness on the MiMarco.** The table shows the effectiveness on the test set. ***: two-sided t-test w.r.t. BM25+CLART5. with p-value<0.005

At test time, we perform a first-stage retrieval on the initial query using the pyserini (J. Lin et al. 2021) implementation of BM25. We then apply our model as a second-stage ranker with additional information. We set the number of retrieved documents to 100.

6.4.3 Metrics and Baselines

We use classical metrics to evaluate the document ranking quality, namely the normalized discounted cumulative gain (NDCG) at rank 1, 3, and 10; and the Mean Reciprocal Rank (MRR) at rank 10.

To evaluate the potential of our mixed-initiative dataset, we compare the performance of our model, noted **BM25+CLART5**, against the following approaches:

- **BM25.** BM25 is a well-known sparse first-stage retriever commonly used as a baseline (Thakur et al. 2021).
- **BM25 + RM3.** RM3 is a pseudo-relevance feedback method for query expansion. The query is expanded using expansion terms extracted from the top 10 retrieved documents. RM3 is a competitive baseline and is still used for benchmarking IR models (Thakur et al. 2021; Adolphs et al. 2022).
- **BM25 + MonoT5.** MonoT5 is a second-stage ranker pre-trained on the original training set of MS MARCO, i.e. only queries and relevance judgments. This model achieves state-of-the-art performance on the beir leaderboard (Thakur et al. 2021) and is a natural baseline as BM25+CLART5 uses the same second-stage pre-trained model before fine-tuning it on mixed-initiative interactions.

6.4.4 Effectiveness of mixed-initiative-oriented neural ranking

We present the results of our mixed-initiative-driven neural ranking model obtained on the online evaluation pipeline presented in [Section 6.2.4](#) applied on the MS MARCO test set ([Table 6.6](#)).

[Table 6.6](#) highlights the fact that additional information helps BM25+CLART5 improve significantly all retrieval metrics on the mixed-initiative-augmented MS MARCO dataset. For example, BM25+CLART5 increases the MRR@10 score by 0.034 point compared to BM25+MonoT5. Further analysis of the results on MS MARCO shows that for 33.0% of the queries, the relevant passage is not retrieved in the top-100 by BM25, leading MRR@100 to 0.0. For 25.6% of the queries MonoT5 and ClarT5 obtain the same MRR@10. For 30.3% of the queries, BM25+CLART5 obtains a better MRR@10 while 11.1% obtain a lower MRR@10. Overall these results show that the feedback provided by the user simulation to the clarifying question is relevant and useful. It helps increase the ranking of relevant passages. This result indirectly confirms that the simulated interactions indeed encode relevant information to the underlying search intents, which is what real users would provide in conversations. Therefore, the proposed simulations are reasonable.

6.5 Complementary Experiments

6.5.1 Extension to Multi-Turn Interactions

In the previous section, we simulated one interaction given a single query $X = (q, cq, a)$ for the online inference. However, multiple facets can be extracted from retrieved passages. This means that sequences of interactions X_0, \dots, X_t can be inferred by sequentially selecting different facets. While a new pool of passages could be retrieved using the last interaction, we only consider here facets from passages retrieved with the initial query. Each t^{th} turn exploits the t^{th} document in the document list by the first-stage ranking to build a facet and generate a clarifying question. Multi-turn interactions are therefore generated in a non-arbitrary order.

Impact on the design of the neural ranking model. We propose to extend the model to multi-turn re-ranking using multiple clarification turns around the same query. We evaluate passages using multiple interactions around the same

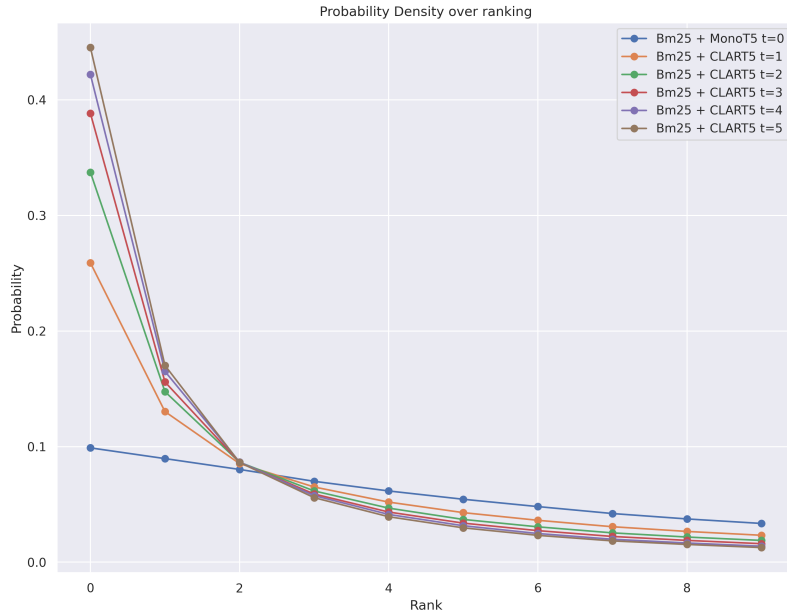


Figure 6.2 – **Score Distribution over ranking.** This figure shows the score distribution (probability) by rank. The score distribution is linear with MonoT5, however, with additional feedback provided by the US, the probability of top-ranked documents increases. Showing that the model is more confident in the new ranking. This is coherent with the decreasing entropy as the number of clarifying turns increases.

search intent. At each time step t a new score s_d^t is computed for the passages in the same ranking using a single interaction. This score is computed using Equation 6.9 which predicts cumulative relevance scores at all interactions, i.e. the sum of relevance scores till time T . This score is used as the ranking score of a document following a sequence of interactions $X_t = \{q, cq^1, a^1, \dots, cq^t, a^t\}$. cq^t and a^t are the clarifying questions and the answer is generated at timestamp t .

$$s_d^T = \sum_{t=0}^T \log p(\text{relevant} = 1 | q, p, cq^t, a^t) \quad (6.9)$$

where s_d^T is the score of document p at time T . As the ranking is updated between turns, we select facets from the top retrieved passage at each time step. We evaluate the retrieval performance at different lengths of interactions, from $T = 1$ to $T = 5$. We also report ranking entropy (Shannon 1948) as a measure of the system’s confidence by measuring how the scores are distributed in the

Multi-turn Conversations	mrr@100	Entropy
User: consolidate function	0.125	
System: do you want to know what human resources functions the county provides?		
User: no	0.14286	
System: are you looking for media files referenced to itunes?		
User: no	0.16667	
System: do you want to know how to consolidate data source?		
User: no	0.2	
System: do you want to know how to use excel to consolidate data?		
User: yes	0.5	
<hr/>		
User: where was the bauhaus built	0.04167	3.854
System: do you want to know where the bauhaus was founded?		
User: no	0.25	3.529
System: do you want to know where the gropius sommerfeld house was built?		
User: yes	0.5	0.693
System: do you want to know where the bauhaus art school was built?		
User: no	1.0	0.000218
<hr/>		
User: here there be dragons comic	0.16667	4.27
System: are you looking for a graphic novel called the deep ?		
User: no	0.2	3.12
System: are you referring to the green arrow comics?		
User: no	0.25	2.93
System: do you want to know what lenox globe dragons means is?		
User: no	0.33333	1.81
System: are you looking for a james owen american comic book		
User: yes	1.0	1.15

Table 6.7 – **Examples of successful cases.** Examples of successful cases. In this example, asking clarifying questions helps to refine the user’s intent and better identify relevant passages. These examples show typical ambiguity in the retrieved document. For example, the Bauhaus can refer to the rock band, the art school, the prize *here there be dragons comic* can also refer to multiple comics with the same name.

ranking as shown in the Equation 6.11. This entropy is maximized when the score distribution is uniform over the ranking.

$$\text{Entropy} = - \sum_{i=1}^{|D|} p(d_i) \log(p(d_i)) \quad (6.10)$$

$$\text{with } p(d_i) = \frac{\exp(s_{d_i})}{\sum_{j=1}^{|D|} \exp(s_{d_j})} \quad (6.11)$$

Results Table 6.9 shows the ranking performance while increasing the number of clarifying turns for the MS MARCO test set. We observe that increasing the number of turns does improve MRR, NDCG scores and while minimizing the Entropy, suggesting that additional feedback helps to refine the user’s information need further.

We depict qualitative examples of successful and unsuccessful clarification sequences in Table 6.7. We can see in the first example that additional interaction helps better refine passage scores leading to better MRR@100, while the entropy decreases. In the last interaction, the entropy is very low, meaning that the score

Failure cases	mrr@100	Entropy
User: what are tor	0.0	3.776
System: do you want to know if tor is free?	0.0	3.612
User: no		
System: do you want to know more about tors transoral robotic surgery?	0.0	3.586
User: no		
System: do you want to know how to connect to a computer network	0.0	3.567
User: no		
System: are you looking for a web browser for tor onion?	0.0	3.541
User: no		
System: do you want to know about tor anonymity?	0.0	3.329
User: no		
User: what is the fastest production sedan	1.0	3.484
System: do you want to know what the fastest production sedan is?	0.1	3.463
User: yes		
System: do you want to know what the fastest production sedan is?	0.1	3.416
User: yes		
System: do you want to know what the fastest production sedan is?	0.090	3.346
User: yes		
System: do you want to know what the fastest production sedan is?	0.083	3.251
User: yes		

Table 6.8 – **Examples of failure cases.** Examples of failure cases. In the first example, none of the relevant passages are retrieved, asking clarifying questions and re-ranking passages do not improve scores. In the second example, the CM fails to capture the passage’s facet and generates the same questions. Additionally, this tends to decrease the scores.

	MRR@10	NDCG@1	NDCG@3	NDCG@10	Entropy
BM25 + CLART ₅ T=1	0.3863	0.2788	0.3817	0.4327	2.951
BM25 + CLART ₅ T=2	0.44467	0.35186	0.43734	0.48038	2.303
BM25 + CLART ₅ T=3	0.48176	0.39828	0.47483	0.51089	2.163
BM25 + CLART ₅ T=4	0.50861	0.43266	0.50321	0.53183	2.06
BM25 + CLART ₅ T=5	0.52949	0.45946	0.52311	0.54814	1.99

Table 6.9 – **Effectiveness results on T interaction.** The table shows the effectiveness results on T interaction turns issued on the MS MARCO test sets.

distribution is dense over a few passages. On the other hand, the Table 6.8 shows examples of failure cases where relevant passages are not even retrieved or where the interaction decreases the MRR. The failure cases we observe, where the interaction turns to deteriorate the ranking, shows that generated interactions are still not perfect. There is room for improvement in the future.

6.5.2 Transferability of the Methodology

To test the potential to transfer the generators trained on a dataset to other datasets, we apply the same clarifying model \mathcal{CM} and user simulation \mathcal{US} trained on MS MARCO (as described in Figure 6.1) to generate simulated interactions on a new Natural Questions dataset (Kwiatkowski et al. 2019b). Results are pre-

	MRR@10	NDCG@1	NDCG@3	NDCG@10
BM25	0.2634	0.1648	0.2354	0.3055
BM25 + RM3	0.2483	0.1457	0.2177	0.2941
BM25 + MonoT5	0.4422***	0.3314***	0.4190***	0.4764***
BM25 +CLART5 T=1	0.4749	0.3725	0.4471	0.5012
BM25 +CLART5 T=2	0.54674	0.46002	0.51072	0.552
BM25 +CLART5 T=3	0.58267	0.50956	0.54204	0.57787
BM25 +CLART5 T=4	0.6047	0.53911	0.563	0.59291
BM25 +CLART5 T=5	0.62115	0.56286	0.57666	0.60469

Table 6.10 – **IR effectiveness on NQ**. IR effectiveness on the augmented version of the Natural Question test set (3452 queries). ***: two-sided t-test w.r.t. BM25+CLART5 T=1. with p-value<0.005

sented in Table 6.10. The higher results obtained by our method BM25+CLART5 w.r.t. other baselines suggest that the generated mixed-initiative interactions can benefit the neural ranking model. In other words, the generators trained on a dataset can be transferred to another dataset to create reasonable simulations. The experimental results on Natural Questions are consistent with those on MS MARCO. This result is particularly interesting, showing that our methodology can be used in inference of out-of-domain datasets. This opens the potential perspective of constructing generic simulators of mixed-initiative interactions for any ad-hoc IR dataset.

6.5.3 Additional Analysis: Ranking similarity

To validate our results, we compute similarity metrics between ranking at each turn in the conversation. To measure the similarity between document rankings at different iterations, we rely on the Rank-Biased Overlap (RBO) (Webber et al. 2010):

$$\text{RBO}(S, T, p) = (1 - p) \sum_{d=1}^{\infty} p^{d-1} \cdot A_d \quad (6.12)$$

where S and T are two document rankings, d is the actual depth of the ranking. A_d expresses the agreement (the size of the intersection of the two rankings) at depth d : $A_d = \frac{|S_{:d} \cap T_{:d}|}{d}$. p determines the weight given to the top ranked document. We set $p = 0.9$, which means that the 10 top documents weigh 85% of the score.

RBO measures the similarity between incomplete and non-conjoint rankings and also values more heavily top-ranked documents. The more diverse the rankings, the lower the score. Figure 6.3 shows the ranking similarity for each addi-



Figure 6.3 – **Passages ranking similarity.** Passages ranking similarity between interaction turn. Ranked Biased Overlap (RBO) metric ($p=0.9$). Interaction 0 corresponds to the document ranking using Bm25 + MonoT5

tional user feedback. We can observe that the ranking seems to stabilize with the number of interactions: the similarity is higher between 4 and 5 interactions than between 0 and 1 interactions.

6.6 Conclusion and discussion

There is a critical need for adequate datasets with mixed-initiative interactions for conversational IR, but creating such a dataset is very costly. In this chapter, we proposed a method to augment ad-hoc IR datasets by simulating a simple form of mixed-initiative interactions between a user and a conversational IR system. This method generates automatically clarifying questions and answers from a large open-domain IR dataset, making it possible to experiment conversational IR approaches at a large-scale. The proposed approach is generic and can be applied to any existing ad-hoc IR dataset. In the experiments, we augmented the MS MARCO dataset and evaluated the quality of the interactions with intrinsic and extrinsic tasks, relying on automatic metrics and human evaluations. The results show that, despite the simple form, the generated interactions are relevant to the search intents and useful for better document ranking. This is a first investigation

on large-scale dataset augmentation for conversational IR. It demonstrates the feasibility of the automatic construction of datasets. As a first investigation, this methodology has several limitations that can be improved in the future.

- First, our investigation is limited to clarifying questions based on a single facet, often assimilated to questions of the type: "Are you referring to 'facet'?". However, real clarifying questions might also question about multiple topics/facets in a single turn (ex: Are you interested to know about *topic1*, *topic2* or *topic3*) or also be formulated as open-ended questions (e.g., "What would you like to know about *topic*?"). These more complex questions are more difficult to generate and answer in simulations, but can potentially bring more information and be more natural in the conversation.
- Second, the facet extraction relied on a few keywords and this can be improved. We observe that when passages are long and address multiple topics, the generated question may not represent the topic addressed in the passage.
- Third, In a realistic CS, the user might be more or less cooperative and provide more or less information about its intent, however controlling information leaks and verbosity in answers is not easy. To better evaluate how simple answers can be used by CS system, the user simulation has been limited to the least cooperative answers 'yes'/'no'.
- Finally, we also generated multi-turn interactions but did not consider the dependency between turns. In real CS, later turns may depend on previous ones. More reasonable simulations of multi-turn interactions should take the dependency into account.

Despite the limitations, the demonstration of feasibility made in this chapter to create large-scale conversational IR datasets opens the door for more investigations at large scale on the topic.

PAQA: TOWARD PROACTIVE OPEN-RETRIEVAL QUESTION ANSWERING

Chapter abstract

The work in this chapter, done in collaboration with Philippe Preux, Jian-Yun Nie and Laure Soulier, is currently under review

- Pierre Erbacher, Jian-Yun Nie, Philippe Preux, and Laure Soulier (2023c).
PAQA: Toward ProActive Open-Retrieval Question Answering

Contents

7.1	Introduction	104
7.2	Task Definition	105
7.3	PAQA dataset	105
7.3.1	Extending AmbigNQ with clarifying questions	106
7.3.2	Evaluating generated questions	108
7.3.3	Proactive question answering with generative model	109
7.3.4	Automatic Evaluation Metrics	110
7.4	Results	111
7.4.1	Human evaluation results	111
7.4.2	Effectiveness of baselines	112
7.5	Conclusion and Discussion	114

7.1 Introduction

In the previous chapter, we proposed a method to augment IR datasets. However, this method did not take into account the query’s ambiguity, resulting in clarifying questions that remain straightforward and single-topic clarifying questions of the type ‘Are you interested in *[topic]*?’’. Additionally, clarifying questions were considered generated from retrieved passages regardless of possible ambiguities. In this chapter we address this limitation, we propose to explore how to train and evaluate systems to detect and resolve ambiguous questions by interacting with users.

Identifying possible underlying ambiguities requires models to discern probable answers across multiple documents. This chapter addresses these gaps by proposing the PAQA dataset containing annotated ambiguous questions with clarifying questions resolving underlying ambiguities and annotated documents, aiming for improved training and evaluation of systems in recognizing and resolving ambiguities. To this end, we build upon the AmbiNQ dataset (Min et al. 2020) dataset by providing clarifying questions generated with GPT3. The dataset is associated with reference models aiming at jointly detecting ambiguity and generating clarifying questions whenever necessary. We carried out human and automatic experiments to assess the relevance of the resulting dataset.

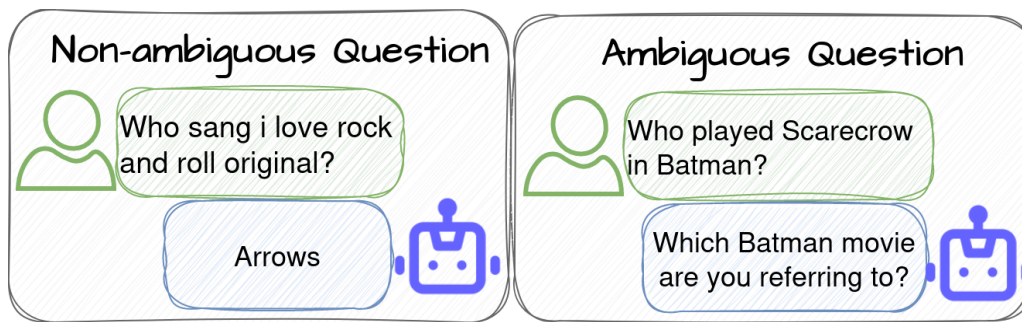


Figure 7.1 – **Proactive question-answering task.** The figure depicts the proactive question-answering task. The system answers directly for non-ambiguous questions (left) and asks clarifying questions for ambiguous questions (right)

7.2 Task Definition

A typical open-retrieval question-answering task involves extracting information from large collections of documents to answer questions accurately. The proactive question-answering task, as illustrated in [Figure 7.1](#), additionally focuses on detecting and resolving ambiguities in questions posed to a system. The system is faced with two types of questions: non-ambiguous ones for which the system can directly answer and ambiguous ones for which the system must recognize that there are multiple possible interpretations. In this latter case, instead of providing a direct answer that may not be relevant, the system aims to clarify the user’s need by asking a clarifying question. These interactions assist users in specifying their needs, allowing systems to deliver more precise answers in subsequent conversation turns. Formally, questions q are associated with a set of equally plausible answers a_1, \dots, a_n grounded to a collection of passages $p \in P$, and considered ambiguous if $n > 1$. The system is expected to output the answer directly when the question is not ambiguous ($n = 1$) and to ask a clarifying question cq that resolves the underlying ambiguity otherwise. Systems should be evaluated on their ability to detect ambiguities and generate relevant clarifying questions.

7.3 PAQA dataset

Training and evaluating models to perform the task described in [Section 7.2](#) requires having a large dataset with labeled ambiguous questions, annotated

documents within a collection, and clarifying questions and answers. Because there are currently no known datasets providing this supervision, we propose the PAQA dataset, an extension of the existing AmbigNQ datasets that additionally contains clarifying questions (Section 7.3.1). We also provide reference models and evaluation metrics (Section 7.3.4).

7.3.1 Extending AmbigNQ with clarifying questions

Source Collection The AmbigNQ (Min et al. 2020) dataset is a question-answering dataset specifically designed to tackle ambiguous questions in an open-domain setting. It was introduced with the AmbigQA task which consists of predicting a complete set of plausible answers given ambiguous questions. This dataset is constructed on NQ-open (Kwiatkowski et al. 2019a) and contains about 14000 manually annotated examples across various topics. Questions q are associated with multiple pairs of plausible query interpretations and corresponding answers $(q_1, a_1), \dots, (q_n, a_n)$ with $n \geq 1$ found on Wikipedia. Questions are considered non-ambiguous if $n = 1$. An example is provided at the top of Figure 7.2 (the AmbigNQ grey box). The dataset is balanced between ambiguous and non-ambiguous questions. Additionally, to the Wikipedia dump containing (21 million passages), AmbigNQ also provides semi-oracle (evidence) articles, which are 3 annotated Wikipedia pages either containing answers or not (see (Min et al. 2020)). AmbigQA is associated with 2 tasks: 1) multiple answers prediction: Given q , output the set of plausible answers $(a_1, ..a_n)$ with n being unknown. And 2) Question Disambiguation that given q and the set of answers (a_1, \dots, a_n) it must output the (q_1, \dots, q_n) with minimal edit. In the context of Conversational Search (CS) between a system and a user, both AmbigQA’s tasks are not suitable, as the goal is to cover all answers without considering the user’s information needs. This provide a motivation to augment the dataset with clarifying questions to allow proactive clarification.

Methodology As illustrated in Figure 7.2, given the set of disambiguated questions (q_1, \dots, q_n) , we generate a clarifying question cq asking how the question q should be interpreted. We used openai’s GPT3-davinci¹ with few-shot examples. The prompt is structured as follows:

```
"Generate a clarifying question given
multiple queries \n\n"
```

1. <https://openai.com/>

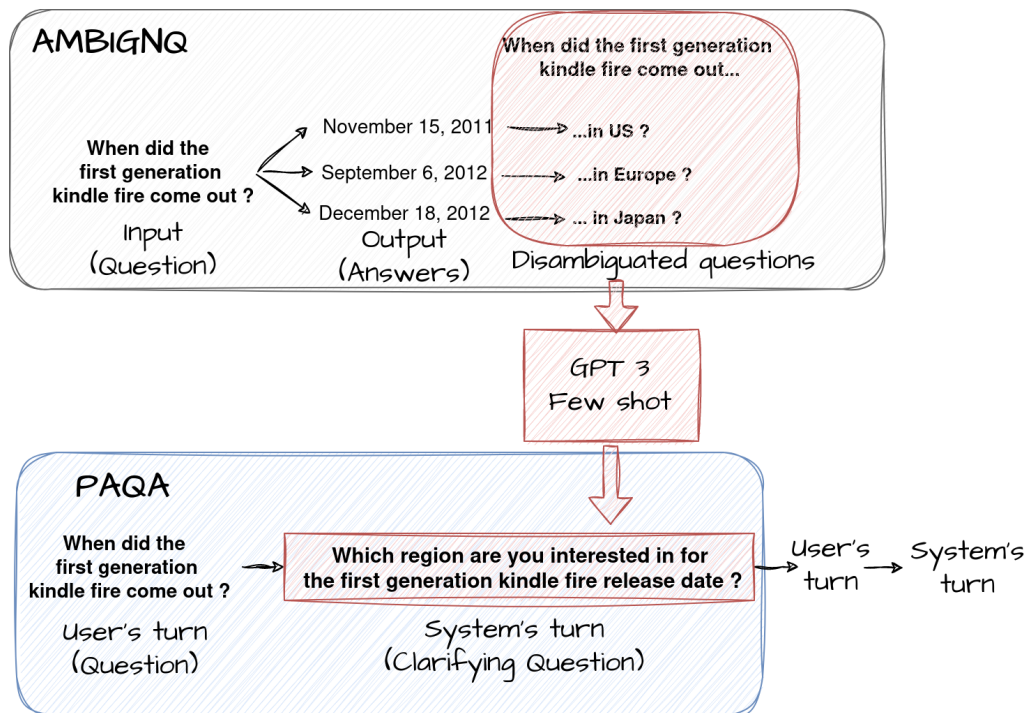


Figure 7.2 – **PAQA dataset**. The figure illustrates the PAQA dataset construction: enriching the AmbigNQ dataset with clarifying questions.

```

"Examples:\n"
"-----\n"
"Query1:Who is the 4th chairman of
african union commission?\n"
"Query2:Who is the 3rd chairman of
african union commission?\n"
"Query3:Who is the 2nd chairman of
african union commission?\n"
"Question:Are you interested in 2nd,
the 3rd or the 4th chairman of
african union commission \n"
"-----\n"
...[other examples]
"-----\n"
"Query1:How many teams are there in
the afl in 1960-1965?\n"
"Query2:How many teams are there in
the afl in 1966-1967?\n"
"Query3:How many teams are there in
the afl in 1968-1970?\n"

```

	Size	Ambig	nb_{ans}	length cq
train	10000	47.1%	2.9	18.5
dev	1001	54.2%	3.3	18.6
test	1000	52.5%	3.2	18.5

Table 7.1 – **PAQA Statistics.** Table describing the PAQA dataset. Ambig refers to the percentage of ambiguous questions. nb_{ans} is the average number of answers for ambiguous questions. "length cq" is the mean length of generated clarifying questions.

Question	When did england last make the quarter final of the world cup?
Label	Are you looking for the men’s or women’s FIFA World Cup?
Q	Are you looking for the last time England made the quarter final of the world cup or the last time they made the quarter final of the world cup?
$Q + P_{evidence}$	Are you looking for the men’s or women’s world cup quarter finals?
$Q + P_{evidence} + A_{gold}$	Are you looking for the men’s or women’s world cup quarter finals?
$Q + P_{dpr}$	Are you looking for the quarter final of the FIFA World Cup or the Rugby World Cup?
$Q + P_{dpr} + A_{extracted}$	Do you want to know the year or the date of England’s quarter final appearance in the FIFA World Cup?

Table 7.2 – **Examples clarifying questions.**

"Question:"

The resulting PAQA dataset contains questions, associated sets of possible answers, annotated documents containing answers (evidence), clarifying questions, and the Wikipedia collection, (see Table 7.7). The Table 7.1 describes the dataset’s statistics.

7.3.2 Evaluating generated questions

We evaluate the quality of the clarifying questions in PAQA using human evaluators. 100 clarifying questions and their associated set of disambiguated questions are sampled from the dataset, and 6 participants are asked to assess both Naturalness and Relevance. Naturalness is defined as being written in a fluid manner, in everyday language, coherent, and could have been generated by a human. Annotators can choose between Unnatural, Fair, Good, and Natural. Relevance measures how the clarifying questions cover/represent the set of provided disambiguated questions. Annotators can choose between irrelevant, partially relevant, and fully relevant. Each question is annotated by 2 humans. It is important to note that our annotators, despite not being specialists in the field or native speakers of English, are well-equipped for their task. The assign-

ment is straightforward and evaluators are guided by comprehensive guidelines and illustrative examples, that ensures clarity. Evaluators are presented with the ambiguous queries, corresponding answers, disambiguated questions from the AmbigNQ dataset, and the automatically generated clarifying questions. They assess the relevance by comparing the scope of answers covered in the clarifying questions compared with the ‘disambiguated queries’ annotated in the AmbigNQ dataset.

7.3.3 Proactive question answering with generative model

7.3.3.1 Model architecture

While this task can be decomposed into a classification task and a generation task, our approach involves using a single generative model to tackle the whole task by generating either the answer or a clarifying question when needed. We consider a sequence-to-sequence architecture trained to generate a clarifying question cq for input question q associated with multiple plausible answers and directly provide answers if the question q is non-ambiguous. We explore different versions of the model:

Question-only Q : The model is only conditioned on the input question q during training and inference. This means that the LM must rely entirely on its parametric knowledge for this task.

Retrieval Augmented $Q + P_{evidence}$ or $Q + P_{dpr}$: The clarifying model takes as input a set of n passages $p_1 \dots p_n$ in addition to the query q . Passages noted P_{dpr} are retrieved by a dense passage retriever (DPR) model (Karpukhin et al. 2020) and re-ranked with a cross-encoder. Passages noted $P_{evidence}$ are evidence passages included in the AmbigNQ dataset (Section 7.3.1).

Retrieval Augmented with Pre-extracted Answers: This setting refers to the retrieval augmented model by integrating answers extracted from the retrieved passages. To do so, given top- n passages $p_1 \dots p_n$, we use a QA model to predict answers $a_1 \dots a_n$ associated with scores $s_1 \dots s_n$. Answers are filtered to maintain a set of unique answers, and we consider a threshold to filter predicted answers based on their scores. Answers are denoted A in the results. Answers noted A_{gold} and $A_{extracted}$ are gold and automatically extracted answers, respectively.

For all model variants, inputs (if considered) are concatenated as follows:

$$Question : q Docs : p_1 \dots p_n Answers : a_1, \dots, a_n \quad (7.1)$$

Input	ROUGE-L	BLEU-1	METEOR
Q	0.512	0.180	0.480
Q + $P_{evidence}$	0.554	0.213	0.518
Q + $P_{evidence}$ + A_{gold}	0.560	0.225	0.527
Q + P_{dpr}	0.546	0.204	0.509
Q + P_{dpr} + $A_{extracted}$	0.519	0.189	0.484

Table 7.3 – **Clarifying questions generation performances.** Metrics are computed only on generated cq.

7.3.3.2 Implementation details

We implemented our code base using the Transformers library (Wolf et al. 2020). For the generative model, we used the Flan-Large (H. W. Chung et al. 2022) available on the Huggingface hub². This model has a context size of 512 tokens and can take 4 passages as input. We used the pre-trained dense retrieval model DPR (Karpukhin et al. 2020) to encode questions and retrieve Wikipedia passages. To train the seq-to-seq model, we follow Min et al. (2020) split evidence into 100-word passages, and re-rank them using a cross-encoder given the question q . For re-ranking passages, we rely on the pre-trained MonoT5³ (Pradeep et al. 2021) trained on MSMARCO (Nguyen et al. 2016a). For Automatic answers extraction, we rely on the deberta-large model (P. He et al. 2020) trained on squad (Rajpurkar et al. 2016). We used a learning rate of $2e - 5$ with a batch size of 64 for 15 epochs.

7.3.4 Automatic Evaluation Metrics

Models evaluated on the PAQA task should be evaluated on their ability to detect underlying ambiguities and to generate relevant clarifying questions. The F1 score, recall, and precision are known metrics to evaluate classification performances. The quality of generated clarifying questions can be evaluated using ROUGE (C.-Y. Lin 2004), METEOR (Banerjee and Lavie 2005), BLEU (Papineni et al. 2002b) commonly used for generative tasks with available gold references.

Input	Accuracy	Precision	Recall
Q	0.527	0.535	0.920
Q + $P_{evidence}$	0.644	0.699	0.569
Q + $P_{evidence}$ + A_{gold}	0.873	0.952	0.798
Q + P_{dpr}	0.572	0.619	0.487
Q + P_{dpr} + $A_{extracted}$	0.565	0.621	0.447

Table 7.4 – **Ambiguous questions classification performances.**

7.4 Results

7.4.1 Human evaluation results

Naturalness	
Unnatural	Sounds awkward or incorrect, either in terms of grammar, vocabulary, or syntax
Fair	Understandable, but still has some issues with flow or clarity
Good	well-written, clear, and flows nicely, but may not necessarily sound completely natural or effortless
Natural	well-written and sounds like something a native speaker would say, with appropriate vocabulary and syntax for the context.

Table 7.5 – **Naturalness metric description.** Metrics used to annotate naturalness of clarifying questions

Relevance	
Irrelevant	Not related to the question or contains hallucinations
Partially-Relevant	Relevant but does not cover all the possible interpretations Q_1, \dots, Q_n
Fully-Relevant	Relevant and covers all annotated intents Q_1, \dots, Q_n

Table 7.6 – **Relevance metric description.** Metrics used to annotate relevance of clarifying questions

Metrics are described in Table 7.5 and Table 7.6. Our sample of evaluated questions is considered mostly natural with 0 Unnatural, 1 Fair, 5 Good, and 94 Natural. For relevance: 1 is unrelated, 18 are partially relevant meaning that they are not covering all the intents, and 81 are fully relevant covering all the intents. The consensus among annotators, as reflected by Cohen’s kappa scores, stands at

2. <https://huggingface.co/google/flan-t5-large>
3. <https://huggingface.co/castorini/monot5-base-msmarco-10k>

0.68 for the naturalness and 0.79 for the relevance of the questions. This indicates a high level of agreement on both counts. Concerning the 18 Partially Relevant, we observed that 17 of them have 6 or more intents. When there are numerous intents, the language model tends to group or leave suggestions to generate more natural questions, which is more desirable than list-like generation, especially for natural conversational applications where multiple rounds of clarification are required. The unrelated sample is indicating a likely false negative, given the intents: *Japanese hotel run by the same family for 1300 years in (Komatsu?/Hayakawa?)* the generated query, *Are you looking for a hotel in Komatsu or Hayakawa?*, seems unrelated. Its phrasing implies a search for booking instead of specifics about the centuries-old family-run Japanese hotel. Overall, the human evaluation shows that clarifying questions are very natural and relevant regarding annotated ambiguities.

7.4.2 Effectiveness of baselines

Table 7.3 and Table 7.4 showcase generation and classification results across various configurations. Models using evidence ($P_{evidence}$) outperform retrieval pipelines in accuracy and generation with an accuracy of 0.64 with evidence and 0.57 with passages retrieved from DPR. This suggests that the retrieval quality highly affects performance. The model relying on parametric knowledge ("Question only") scores lower in BLEU, ROUGE, and METEOR due to the lack of context and produces poor clarifying questions as seen in Table 7.2 and Table 7.8. Surprisingly, pre-extracting answers does not yield better results and marginally lower the generation and accuracy metrics, from a BLEU of 0.20 for the DPR setting to 0.18 with the model relying on DPR and answers.

Figure 7.3 shows performances with multiple threshold values used to extract answers. This suggest that the quality of generation is sensitive to answers, and retaining only answers with a high level of confidence provides better results. However, the extractive QA model may still retrieve non-relevant answers with high probability, penalizing the entire pipeline. Setting a high threshold also excludes all extracted answers, further penalizing the model due to out-of-domain answers/passages. An example is shown in Table 7.2, where both models with evidence manage to match the labeled CQ, the model with DPR only manages to generate another clarifying question, and the DPR + answer is influenced by retrieved answers with different date formats. The results show that PAQA enables models to generate accurate clarifying questions and detect ambiguities, however,

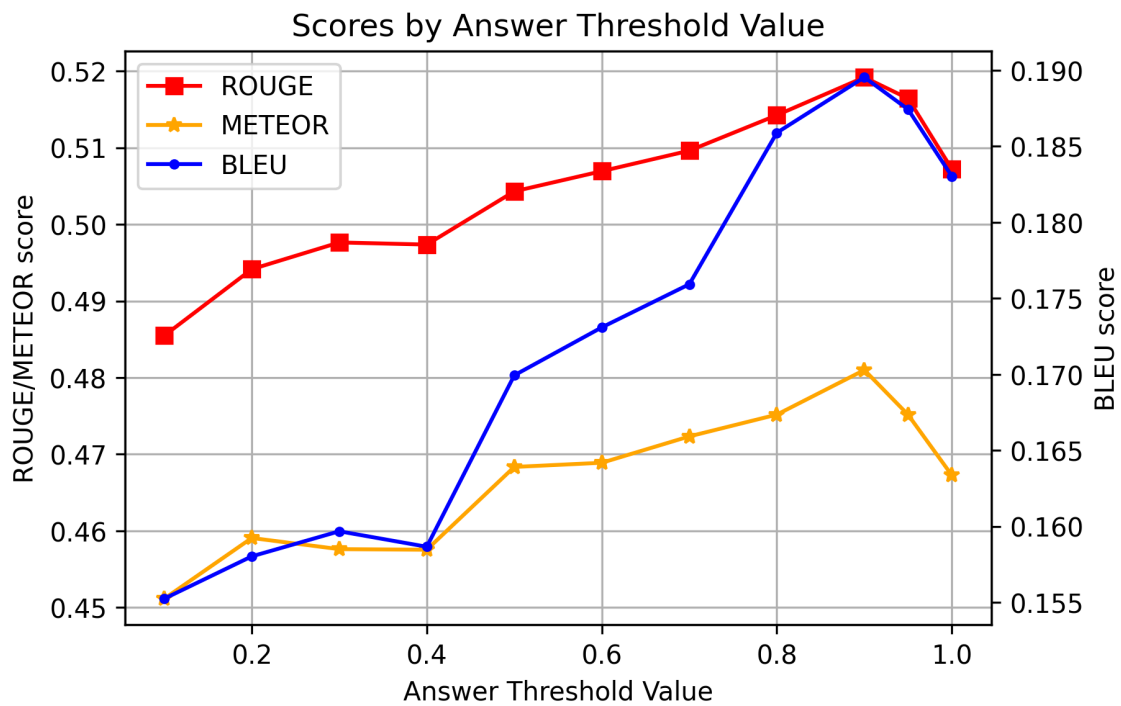


Figure 7.3 – **Evolution of reconstruction metrics against threshold.** Graph displaying the evolution of various of ROUGE, METEOR, BLEU metrics regarding different threshold values.

Q:When did the first generation kindle fire come out?

Intents, Answers:

1. When did the first generation kindle fire come out in US?, A:November 15, 2011
2. When did the first generation kindle fire come out in Europe?, A:September 6, 2012
3. When did the first generation kindle fire come out in Japan?, A:December 18, 2012

CQ:Which region are you interested in for the first generation kindle fire release date?

Q:How many times has green bay beat the bears?

Intents, Answers:

1. How many times has green bay beat the bears in the 2010s?, A: 17
2. How many times has green bay beat the bears in the 2000s?, A:12
3. How many times has green bay beat the bears in the 1990s?, A:13
4. How many times has green bay beat the bears all time?, A:99
5. How many times has green bay beat the bears consecutively?, A:10

CQ:Do you want to know about the 2010s, 2000s, 1990s or all time, and are you interested in c

Table 7.7 – **Examples from PAQA dataset.** Table showing samples from PAQA. Q is the original question, plausible intents and answers are labelled by annotators in the original AmbigQA dataset. CQ are clarifying questions generated using GPT3.

the main bottleneck remains in the retrieval capabilities with a performance gap between models with evidence and retrieved passages.

7.5 Conclusion and Discussion

While conversational systems have made significant progress in generating natural language responses, they do not ask relevant clarifying questions when faced with documents supporting multiple plausible answers but rather generate a comprehensive answer. We propose a new dataset with aligned documents, questions, and clarifying questions to better train and evaluate such systems in their ability to detect ambiguity and ask clarifying questions. PAQA is the first dataset providing questions, answers, supporting documents, and clarifying questions covering multiple ambiguity types (entity references, event references, properties, time-dependent. . .) with enough examples for fine-tuning models. We provide various baselines with generative models. Experiments suggest that retrieval quality highly affects clarification quality and that pre-extracting answers do not lead to better ambiguity detection. We did not investigate how scaling model parameters, nor how increasing the number of retrieved passages influences performances. Flan-T5 has a context size of 512 tokens, corresponding to 4 passages. While this corresponds to the average number of interpretations for am-

Question	Who wrote the music for how to train your dragon?
Label	Are you looking for the composer or the lyricist of the song "Sticks and Stones" for the film How to Train Your Dragon?
Q only	Are you looking for the composer or the composers of the music for how to train your dragon?
Q+E	Are you looking for the composer of the music for How to Train Your Dragon, How to Train Your Dragon 2 or How to Train Your Dragon: The Hidden World?
Q+E+A	Do you want to know about the music for the film or the song "Sticks & Stones"?
Q+P	Do you want to know about the music for the film, the soundtrack album, or the song "How to Train Your Dragon"?
Q+P+A	Are you looking for the composer of the music for the film or the soundtrack album for How to Train Your Dragon?
Question	Who wrote the books of luke and acts?
Label	Do you want to know about the traditional view or the critical views?
Q only	Are you looking for the author of Luke or Acts?
Q+E	Do you want to know who wrote the books of luke and acts in general or who specifically wrote the books of luke and acts?
Q+E+A	Are you looking for the traditional or critical view of who wrote the books of luke and acts?
Q+P	Do you want to know who is traditionally believed to have written the books of luke and acts or who modern scholars believe wrote the books of luke and acts?
Q+P+A	Do you want to know who wrote Luke or Acts?
Question	When did the movie inside out come out?
Label	Are you looking for the release date of the 2015 movie inside out at Cannes, throughout the United States, at the El Capitan Theatre, in limited New York City theatres, or for a different year?
Q only	Are you looking for the release date of Inside Out in the United States or Canada?
Q+E	Do you want to know about the 1986, 2011 or 2015 movie Inside Out?
Q+E+A	Do you want to know about the 2015, 1986 or 1975 movie Inside Out?
Q+P	Are you looking for the release date of the 2015 or the 1975 movie Inside Out?
Q+P+A	Are you looking for the release date of the movie inside out in competition at the Heartland Film Festival, limited theatrical release, or on DVD and VOD?

Table 7.8 – **Qualitative examples of generated cq.** Table showing qualitative examples of generated cq

biguous, Gao et al (Y. Gao et al. 2021) showed that scaling the number of passages

improves performances for answers predictions. Additionally, we relied on off-the-shelf extractive Question Answering (QA) model, which considers only one answer per passage and therefore may miss some answers. The resulting dataset only contains clarifying questions but no user feedback, additional work could be done to simulate the user's response. Lastly, we did not perform any analysis of model performance on various ambiguity types. As described in the ambigQA paper (Min et al. 2020), we observed that some ambiguities are very different. For instance: "When did inside out came out?" requests a date as an answer but may refer to various entities of the same name, while 'What is the flower of the dead in spanish?' may have different answer types: A direct translation in Spanish or more information about the flower used for spanish celebrations of the dead.

Part IV

PROACTIVE INTERACTIONS WITH SEARCH
ENGINE

NAVIGATING UNCERTAINTY: OPTIMIZING API DEPENDENCY FOR HALLUCINATION REDUCTION IN CLOSED-BOOK QA

Chapter abstract

The work in this chapter, done in collaboration with Louis Falissard, Vincent Guigue and Laure Soulier

- Pierre Erbacher, Louis Falissard, Vincent Guigue, and Laure Soulier (2023a). *Navigating Uncertainty: Optimizing API Dependency for Hallucination Reduction in Closed-Book QA*

Contents

8.1	Introduction	120
8.2	Learning when to search with LLMs	122
8.3	Evaluation protocol	123
8.3.1	Datasets	123
8.3.2	Metrics	124
8.3.3	Model architectures and fine-tuning	124
8.3.4	Baselines and model variants	125
8.3.5	Implementation Details	126
8.4	Results	127
8.5	Conclusion	128

8.1 Introduction

In previous chapters, we focus on proactive models that interact with users to better reduce uncertainties around their information needs. However, models may need to also interact proactively with other machines or systems to reduce uncertainties and solve tasks. Even the larger models (Brown et al. 2020b) are prone to producing inaccurate or false responses, commonly known as hallucinations (Ji et al. 2023). In question answering, retrieval augmented methods such as REALM (Guu et al. 2020), RAG (Lewis et al. 2020b), or RETRO (Borgeaud et al. 2022b; Kenton Lee et al. 2019a), were proposed to reduce Large Language Model (LLM)’s hallucinations. These approaches consist of grounding LLM with a retriever model to add context from a large corpus of documents and generate answers. These architectures are effective as they both improve factualness and reduce hallucinations for specific knowledge-intensive tasks such as Open-domain Question Answering (Kenton Lee et al. 2019a). However, retrieved documents are always considered without consideration of their helpfulness in solving the task. This makes the conversational system passive toward external search engines.

In a second line of work, models, such as LaMDA, BlenderBot, WebGPT, Toolformer(Thoppilan et al. 2022b; Nakano et al. 2022; Shuster et al. 2022b; Schick et al. 2023) are specifically trained to generate a query and rely on a search engine when confronted with questions. These models are actively using external resources to resolve tasks. While these LLMs accumulated a lot of knowledge during pre-training, they are fine-tuned to rely on external databases for each

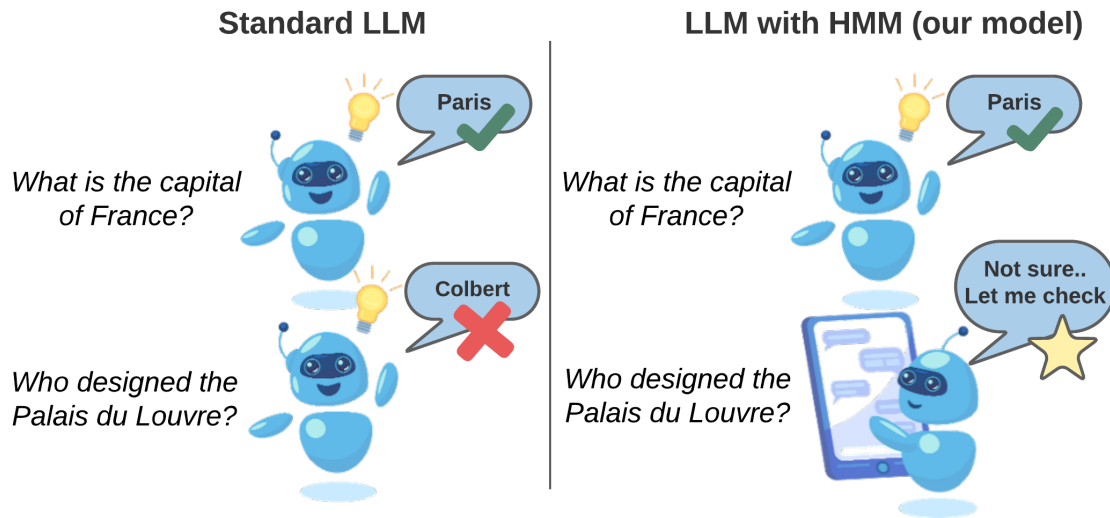


Figure 8.1 – **Task Description.** Our objective is to teach an LLM to trade off between accessing an external tool and answering directly. The LLM should access an external knowledge base only when it is not self-confident about the answer, otherwise, it should generate the answer directly, minimizing the cost of accessing an external search engine.

question, without considering the model’s inherent ability to answer the question. Toolformer calls the web API for almost all the questions, 99.3% with no real discernment between directly answerable questions and the real need for external knowledge. LLMs have accumulated a lot of information and may be able to answer directly when confronted with widely known facts (Brown et al. 2020b; Bubeck et al. 2023; Wei et al. 2022a).

In this chapter, we envision a more nuanced approach that leverages external knowledge while also incorporating LLMs’ intrinsic knowledge. We, therefore, propose a model that either generates a natural language answer or an API call (e.g. $\langle search \rangle$) only when the model is not self-confident about the answer, minimizing the dependency on external resources helps to save inference time and computational costs.

We focus on closed-book question-answering (CBQA) tasks and carried out on two datasets (Natural Questions (NQ) (Kwiatkowski et al. 2019b) and TriviaQA (TQA) (Joshi et al. 2017a)). We study how LLMs perform at self-estimating their ability to correctly answer factual questions.

8.2 Learning when to search with LLMs

Problem Formalization. We consider the set of factual questions Q and a set of all possible answers Ω . Each question $q \in Q$ is associated with a set A of correct answers with $A \subset \Omega$. In CBQA, models are able to answer factual questions without supporting documents, this means relying only on the knowledge stored in parameters. Formally, we define an LLM that maps answers with questions: $LM_\theta : Q \mapsto \Omega$. Let's consider an LLM that generates an answer \hat{a} given a question q , with \hat{a} that might be in the set A of ground truth or not (in this last case the answer is hallucinated). Our objective is to train an LLM to query external resources instead of generating hallucinations or to generate directly the answers, otherwise:

$$LM_{\theta'} : Q \mapsto \Omega \cup \{\langle search \rangle\} \quad (8.1)$$

where $\langle search \rangle$ is a call to an external tool such as a search engine. Training this LLM of parameter θ' can be seen as a budgeted open QA task in which we minimize the probability of generating a wrong answer $P(\hat{a} \notin A | \theta', q)$ and the probability to call an external tool $P(\hat{a} = \langle search \rangle | \theta', q)$ with a budget λ :

$$\operatorname{argmin}_{\theta} \prod_{q \in Q} [P(\hat{a} = \langle search \rangle | \theta', q) + \lambda P(\hat{a} \notin A | \theta', q)] \quad (8.2)$$

where $\lambda \geq 1$ is a hyper-parameter controlling the relative importance between accessing the external resource and a hallucination behavior. The objective of this formulation is to encourage the model to provide direct answers whenever possible, therefore minimizing the cost of searching in an external resource. In this chapter, we limit this analysis to $\lambda = 1.0$ (Equation 8.2). A natural solution to label the model's hallucinations is to verify whether the model outputs are correct and factual. However, the ability to answer correctly to a question correctly is inherent to the model's size and training. There is therefore no fixed dataset with supervision labels identifying when to call an API. We, therefore, propose to leverage a language model fine-tuned on a QA dataset to infer pseudo labels from language model performance during the training. More particularly, we aim to teach LLMs to generate a special sequence of tokens ($\langle search \rangle$), instead of answering incorrectly, without deteriorating the ability of the model to answer questions thanks to a "Hallucination Masking Mechanism".

Hallucination Masking Mechanism Our objective here is to update the model LM_θ to display similar performances on CBQA tasks while also detecting halluci-

nations. Given LM_θ a model able to perform a CQBA task, we learn parameters θ' such that $LM_{\theta'} : Q \mapsto \Omega \cup \{\langle search \rangle\}$ where the LLM can still perform CBQA but predict the $\langle search \rangle$ token instead of hallucinating an answer. We introduce ψ a Hallucination Masking Mechanism (HalM) allowing to mask wrong answers with $\langle search \rangle$ tokens. Formally, $\psi \circ LM_\theta : Q \mapsto \Omega \cup \{\langle search \rangle\}$:

$$\psi(LM_\theta(q)) = \begin{cases} \mathbb{1}(\hat{a}), & \text{if } \hat{a} \in A \\ \langle search \rangle, & \text{otherwise} \end{cases} \quad (8.3)$$

With $\mathbb{1}$ the identity function and $\langle search \rangle$ the sequence of tokens used to query an external knowledge base. ψ enables the generation of labels for data where the identity function is applied for questions answered correctly and hallucinations are masked using $\langle search \rangle$ tokens. This mechanism is composed on top of the original LM_θ to conserve the ability of the model to answer directly when the answer is correct. To avoid additional biases in the experiment, we limit ourselves to in-domain hallucination detection, where questions used for QA fine-tuning and hallucination detection come from the same distribution. This means that we use a single dataset for both steps, avoiding distribution shifts and shared example problems.

8.3 Evaluation protocol

For the experiment, we naturally consider closed-book Question Answering (QA) over factual questions which is an easy way to assess if a model is hallucinating or not.

8.3.1 Datasets

We consider two open-domain CBQA datasets to perform our experiments. **Natural Question Open (NQ-open)** (Kenton Lee et al. 2019b): an open domain question answering benchmark derived from the Natural Question dataset (Kwiatkowski et al. 2019b) which consists of questions from web queries accompanied by a list of appropriate answers, but without the original context provided in Natural Question.

TriviaQA (TQA) (Joshi et al. 2017b): a dataset including questions gathered from quiz league websites and also accompanied by a list of appropriate answers.

		$LM_{\theta'}$		
		C	H	S
LM_{θ}	C	TP	FP	FN
	H	TP	FP	TN

Table 8.1 – **Metric description.** Table showing how predictions are considered. The LM_{θ} is the language model after the fine-tuning and $LM_{\theta'}$ after the second one (Equation 8.2).

Models are finetuned on the train of each dataset to perform traditional CBQA. The dev set is then used to perform the second Halm-based finetuning step. This ensures high CBQA performances. Specifically, we focus on detecting hallucinations within the domain of interest, using questions from the same distribution as those used for QA fine-tuning. By utilizing a single dataset for both steps, we mean to avoid issues related to distribution shifts and shared example problems.

8.3.2 Metrics

The standard approach for assessing generative CQBA model performances is based on the consideration that a generated answer is correct if and only if it constitutes an exact match or correct answer (noted **C**) with at least one element in a list of admissible answers. This metric alone, however, is insufficient to paint a comprehensive picture of the model’s behavior, and we propose to extend it based on a comparison between the ground truth and both LM_{θ} and $LM_{\theta'}$ ’s predictions. Model output can be associated with three distinct events. A model prediction is either Correct (noted **C**), incorrect (noted **H** for Hallucinated), or a query to an external tool, namely $\langle search \rangle$, (noted **S**). As aforementioned, LM_{θ} predictions can only correspond to **C** or **H** events, while $LM_{\theta'}$ predictions can also correspond to **S** type events.

Following these considerations, we define true positive (TP), false positive (FP), true negative (TN), and false negative (FN) events as shown in Table 8.1, and use them to report F1-scores in the results.

8.3.3 Model architectures and fine-tuning

We consider sequence-to-sequence (encoder/decoder) models (Sutskever et al. 2014a) with different sizes to assess how scale might affect the generated data, and

therefore performances. All experiments utilize both the large and XXL T5-SSM models (Raffel et al. 2020a) (770M and 11B parameters, respectively) specifically trained for CBQA using Salient Span Masking (SSM) (Guu et al. 2020). In addition, these models have official checkpoints that were fine-tuned on NQ and TQA, saving us the computational cost of training them ourselves.

Large models are used FP32, however, 11B parameters models are quantized into int8 to fit on GPUs.

Models are finetuned on the train of each dataset to perform traditional CBQA. The dev set is then used to perform the second HalM-based finetuning step. This ensures high CBQA performances. Specifically, we focus on detecting hallucinations within the domain of interest, using questions from the same distribution as those used for QA fine-tuning. By utilizing a single dataset for both steps, we mean to avoid issues related to distribution shifts and shared example problems.

8.3.4 Baselines and model variants

For our models based on T5-Large and T5-XXL models, we consider two strategies to fine-tune with HalM: 1) the standard fine-tuning (FT) and 2) using Low-Rank Adaptation (LoRA) (Hu et al. 2022). Due to computational constraints, the XXL (11B) T5 model is only fine-tuned with LoRA. For training LoRA, we used PEFT (Mangrulkar et al. 2022) and Adapter-transformers (Pfeiffer et al. 2020) libraries to plug the parameters efficient method to LLMs and consider a warmup with a ratio of 0.1, $r = 16$, $alpha = 32$, and a learning rate of $1e - 4$ and $7e - 5$ for large and XXL models.

We compare our model variants to different baselines:

T5-Large and T5-XXL: the models fine-tuned on the train set of the CBQA task. Note that these models have not been trained to call external API, solely to generate answers.

T5-Large+PPL-t and T5-XXL+PPL-t: which is the strongest exogenous hallucination detection method known in the literature (N. Lee et al. 2021; Guerreiro et al. 2023), based on a perplexity threshold. This heuristic assesses the model’s output’s perplexity score and classifies it as a hallucination if it exceeds a predefined, data-derived threshold.

Mistral-7B¹: an in-context learning strong model with 16 examples randomly extracted from the train set. Wrong answers are masked with the ‘search’ sequence.

1. <https://huggingface.co/mistralai/Mistral-7B-v0.1>

These are examples of how the model should behave and be used for in-context learning. We observed that if the number of masked hallucinations and direct answers is unbalanced in the prompt, this also leads to very unbalanced prediction. Thus, we used a balanced set of examples.

Mistral-7B-instruct²: a strong instruction-based LLM prompted to follow this instruction: *Answer to the question only if you know the answer, otherwise answer "I don't know"* followed by the question.

8.3.5 Implementation Details

We used existing checkpoints trained for closed-book QA (Roberts et al. 2020). These checkpoints are available on hugging face hub³. To infer our label, we follow (Roberts et al. 2020) and use greedy decoding. To classify if a prediction is correct in generative QA, predictions are compared using Exact Match against a list of ground truths (GT). Because the list of GT is not exhaustive, a relative amount of predictions are False Negative, introducing noise in the training data for the second fine-tuning. For example, if the model generates "Napoleon I" but the GT only contains "Napoleon", the answer is considered False. To mitigate this, the model predictions are compared with the list of normalized ground truth: all values are lowercase and stopwords and punctuations are removed. We used PEFT (Mangrulkar et al. 2022) and Adapter-transformers (Pfeiffer et al. 2020) libraries to plug the parameters efficient method to LLMs. Large models are used FP32, however, 11B parameters models are quantized into int8 to fit on GPUs. For training LoRA, we used a warmup with a ratio of 0.1, $r = 16$, $alpha = 32$, and a learning rate of $1e - 4$ and $7e - 5$ for large and XXL models.

Regarding TQA, a checkpoint is available for the XXL model, and the T5 Large SSM was fine-tuned by our care using the following hyperparameters used in (Roberts et al. 2020): constant learning rate of $1e - 3$ for 10000 steps, dropout of 0.1 and batch size of 128 and gradient accumulation of 8. However, contrary to what is reported by authors in (Roberts et al. 2020), we encounter overfitting after a few thousand steps.

2. <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1>

3. <https://huggingface.co/>

	NQ				TQA			
	C (\uparrow)	H (\downarrow)	Search	F1	C (\uparrow)	H (\downarrow)	Search	F1
T5-Large	27.3	72.7	0.0	-	19.4	80.6	0.0	-
T5-Large + PPL-t	18.6 (68.1%)	8.9 (12.2%)	72.5	67.7	12.7 (65.4%)	12.7 (15.7%)	74.6	56.4
T5-Large-HalM (FT)	15.7 (57.5%)	7.1(9.7%)	77.2	62.4	14.4 (74.2%)	29.9 (37.0%)	55.8	45.0
T5-Large-HalM (LoRA)	21.3 (78%)	16.6(22.8%)	62.0	65.0	13.6 (70.1%)	27.2 (34.8%)	59.2	45.1
T5-XXL	35.2	64.8	0.0	-	51.9	48.1	0.0	-
T5-XXL + PPL-t	21.7 (61.6%)	12.1 (18.6%)	66.3	65.4	27.7 (53.3%)	24.6 (51.1%)	47.7	63.1
T5-XXL-HalM (LoRA)	23.4 (66.5%)	15.9 (24.5%)	60.7	66.1	28.0 (53.9%)	24.4 (50.7%)	47.6	63.5
Mistral-7B (16 shots)	28.8 (91.7%)	49.9 (72.5%)	21.4	50.6	34.1 (51.8%)	14.7 (42.9%)	51.2	60.0
Mistral-7B-instruct	2.40	2.95	94.65	-	25.9	13.7	60.3	-

Table 8.2 – **Distribution of exact match and hallucinations.** Table showing the distribution of Exact match, Hallucination, and search sequence for NQ and TQA dataset. (%) are showing the remaining fraction of the same behavior (C , H , S) regarding predictions of the base model.

Ratio of correct S	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
C	21.3	27.5	33.7	39.9	46.1	52.3	58.5	64.7	70.9	77.1	83.3
H	78.6	72.4	66.2	60.0	53.8	47.6	41.4	35.2	29.0	22.8	16.6

Table 8.3 – **Fluctuation of correct answers.** A table illustrating the fluctuation in hallucination and correct answer rates as the accuracy of Search S varies

8.4 Results

Table 8.2 shows the results of all model variants and baselines for Natural Question (NQ) and TriviaQA (TQA) datasets, according to the F1-score, and proportions of correct answers C , Hallucination H and search S as defined in 8.3.2. Every rate on adapted models is accompanied by the remaining fractions for each behavior (C , H , S) regarding predictions of the first model.

From a general point of view, all hallucination reduction strategies (PPL-t and HalM) are able to reduce hallucinations regarding the models fine-tuned without consideration of searching on an external API. We notice that for the T5-Large variants the PPL-t (Perplexity-threshold) strategy outperforms the HalM with a high search rate; that might be costly. Our variant T5-Large-HalM fine-tuned with LoRA seems to have a better balance between accurate answer generation and the search rate. By focusing on the T5-XXL architecture, we show that LoRA consistently outperforms PPL-T on both NQ and TQA datasets for most of the metrics. Indeed, the LoRA strategy retains a higher fraction of correct answers on the NQ dataset. For the TQA dataset, both approaches exhibit similar behaviors, with a slight advantage for LoRA, and manage to filter out around half the Hallucinations while retaining a similar amount of Correct answers. Altogether, these results support the claim that our proposed approach enables LLMs to en-

dogenously identify their potential for hallucination better than perplexity-based methods.

To understand the benefit of this mechanism, we consider different correct answer ratios for the search. If the ratio is set to 0.0, all API calls return only incorrect answers, while a ratio of 1.0 returns only correct answers. Table 8.3 shows the variation of this rate for the T5-Large-HalM (LoRA) on NQ. We can see that the user experience heavily relies on the boost of correct answers provided by the search. With a ratio of 0.1, this has similar performances as the T5-Large on NQ. While a ratio of 1.0 provides 83.3% of correct answer the model only searches for 62% of the questions.

Focusing on in-context learning models, one can see that the Mistral-7b does not perform very well with an F1-score of 50.6% on NQ. Additionally, the model is very sensitive to the balance of examples in the prompt. Regarding the instructed capabilities with Mistral-7b-instruct, we observed that the model catastrophically failed to perform the given task as the model outputs 94.65% 'I don't know' while only having 2.4% correct answers on NQ. This suggests that abilities outlined in LLM, namely instructions and in-context learning, are not consistent with the specific behavior to identify uncertainties without devoted fine-tuning.

8.5 Conclusion

We introduced a new model to teach an LLM to internally assess its ability to answer properly a given query, without using anything more than data used for its training. The resulting model can directly identify its ability to answer a given question, with performances comparable -if not superior- to widely accepted hallucination detection baselines such as perplexity-based approaches which are strong exogenous baselines. In addition, this approach enables large language models to condition their generation on their ability to answer appropriately on a given query, a crucially important feature in the Toolformer approach that can learn to search only when needed. In future work, we plan to assess the impact of the λ hyperparameter in the Hallucinations risk/Search trade-off.

Part V

CONCLUSION AND PERSPECTIVE

CONCLUSION

Chapter abstract

This final chapter contains a brief overview and summarization of the contributions. Then, we draw multiple perspectives that offer a comprehensive view of future research directions and exciting prospects that lie ahead. We provide a placement regarding the field of Conversational Search (CS) by describing current limitations, challenges, and possible future research directions.

Contents

9.1	Summary of Contributions	132
9.2	Promising Perspectives and Future work	133
9.3	Open discussion about LLMs and IR	136

We now summarize the contributions of this thesis and offer some future directions of Conversational Search (CS).

9.1 Summary of Contributions

During this thesis, we aim to improve CS by enabling more complex and useful interactions with users. We propose multiple methods and approaches to achieve this goal:

Multi-turn Query Suggestion. First, in [Chapter 4](#) and [Chapter 5](#), we investigate how user simulations can be used to train and evaluate systems that perform query refinement through sequential interactions with the user. In [Chapter 4](#) we focus on sequential click-based interaction with a user simulation for clarifying queries. We provide a simple and easily reproducible framework simulating multi-turn interactions between a user and a Information Retrieval (IR) system agent. The advantage of our framework is the simplicity of interaction, as there is no need for a dataset of real and annotated user-system interactions. Experiments highlight performance gain in terms of document retrieval. This method is later improved in [Chapter 5](#) with a single model generating the suggestions between each interaction with the user simulation.

Dataset for CS. Then, in [Chapter 6](#) and [Chapter 7](#), we explore how existing IR datasets can be enhanced with simulated interactions to improve IR capabilities in CS and how mixed-initiative interactions can serve document retrieval. In [Chapter 6](#) we propose a method to augment ad-hoc IR datasets by simulating a simple form of mixed-initiative interactions between a user and a conversational IR system. This method generates automatically clarifying questions and answers from a large open-domain IR dataset, making it possible to experiment with conversational IR approaches at a large scale. The proposed approach is generic and can be applied to any existing ad-hoc IR dataset. In the experiments, we augment the MsMarco dataset and evaluate the quality of the interactions with

intrinsic and extrinsic tasks, relying on automatic metrics and human evaluations. It demonstrates the feasibility of the automatic construction of datasets. In [Chapter 7](#) we propose to augment the AmbigNQ dataset with clarifying questions to better train and evaluate systems to perform pro-active question-answering tasks, where systems are expected to disambiguate the initial user questions before answering. To our knowledge, PAQA is the first dataset providing both questions, answers, supporting documents, and clarifying questions covering multiple types of ambiguity (entity references, event references, properties, time-dependent. . .) with enough examples for fine-tuning models.

Navigating Uncertainty. Finally, in [Chapter 8](#), we focused on the interaction between systems and an external search engine. We introduced a new approach method to teach a Large Language Model (LLM) to internally assess its ability to answer properly a given query, without using anything more than data comprised used for its training. The resulting model can directly identify its ability to answer a given question, with performances comparable -if not superior- to widely accepted hallucination detection baselines such as perplexity-based approaches which are strong exogenous baselines. It allows models to proactively query search API depending on its ability to answer the question.

9.2 Promising Perspectives and Future work

Generating Datasets using In Context Learning. The current trend in Natural Language Processing (NLP) as depicted in [Figure 9.1](#) is to rely on existing LLMs to generate large and high-quality datasets using few-shot examples to train better models. Y. Wang et al. (2023) show that instruction-following capabilities of pre-trained language models can be improved by bootstrapping from their own generations. This method helps generate diverse instruction/answer pairs that improve the performance of models. Similarly, the Ultrachat dataset is an instruction-following dataset (Ding et al. 2023) that was generated using 2 chat-GPT playing respectively the role of the user and system. They managed to obtain a very large and high-quality dataset for instruction following a wide range of topics. Because these models can be given diverse examples of conversations, one can generate high-quality mixed-initiative conversations from LLMs. This would enable models to actively query search engines and clarify users at each turn in the conversation.



Figure 9.1 – Meme of the current NLP research

Personalization The TREC Interactive Knowledge Assistance Track (iKAT)¹ underscores the importance of personalization in conversational interactions. It advocates for systems capable of multi-turn, multi-perspective conversations that consider user-specific factors like preferences, occupation, and location. These conversations are mixed-initiative, relying heavily on users’ personas and promoting a more engaging, personalized search experience. We believe that this is a promising path toward personalized CS. To improve further personalization one can envision a task where conversational models learn to infer and update the user’s persona from the conversations.

Conversational search systems as agents The potential of Language Model Agents (LLMs) acting as dynamic entities, capable of interacting with search engines and performing various actions to collect a wide array of documents for users, holds great promise. With the growing popularity of LLM-powered tools such as search engines, calculators, calendars, and code executors, LLMs are increasingly perceived as decision-making agents that proactively engage with external tools to efficiently tackle tasks (Z. Wang et al. 2023). These models are expected not only to initiate conversations with users autonomously but also to seamlessly interact with a diverse range of environments, including various tools, the internet, and various APIs. For instance, addressing complex user queries that

1. <https://www.trecikat.com/>

necessitate information from multiple sources might require a well-structured sequence of actions, involving interactions with multiple APIs, with adaptations based on the obtained results. Currently, there are two primary approaches for LLMs to interface with external tools. The first approach involves retrieval-based APIs, which enable LLMs to access relevant APIs for tasks from extensive API aggregators (Qin et al. 2023). The second approach consists of specialized tools designed explicitly for LLMs (Schick et al. 2023; Shuster et al. 2022b). LLMs can gain proficiency in interacting with these tools through in-context learning, utilizing specific prompts and examples, or through additional fine-tuning. For CS, there are multiple challenges, 1) Teaching the LLMs to generate sequences of queries that are effective and not biased to retrieve documents. 2) Teach LLMs to generate answers factually grounded to retrieved information and sources. To evaluate the effectiveness of LLMs in utilizing tools within web or tool environments, various benchmark evaluations have been developed. Notable examples include "Agent-Bench" by Liu et al. and "Mind2Web" by Deng et al. (X. Liu et al. 2023; Deng et al. 2023). However, several studies show that Autoregressive LLMs have no common sense and cannot plan their answer or sequence of actions (Momennejad et al. 2023; Mahowald et al. 2023; Valmeekam et al. 2023). Ar-LLM may not be the ideal architecture for designing CS Model that must plan and anticipate user, results and estimate user's need.

Pro active interaction to align language model with human values The exploration of clarifying questions has primarily focused on their use in search applications, but there is potential for a broader application to enhance prompt clarification and align LLM with human preferences. Current state-of-the-art methods often employ Reinforcement Learning with Human Feedback (RLHF) to fine-tune Language Model (LM)s based on human preferences. In Reinforcement Learning With Human Feedback (RLHF), humans are tasked with making judgments between different LLM-generated sequences, and these judgments are used to adjust the language model. This iterative process aims to generate sequences that maximize expected rewards. However, it's important to note that human preferences and values can vary significantly among individuals. This variability means that different users may have distinct expectations from an LLM when presented with a similar prompt. The reward models used in RLHF typically represent the average judgment across users, potentially overlooking the diversity of preferences. Rame et al. (2023) (Rame et al. 2023) propose an innovative approach known as "Rewarded Soup." This approach involves training various models, each adjusted according to different reward models, and then employing weighted averaging

to select the most suitable model interpolations based on user preferences. One practical way to implement this idea is by training an LLM to proactively ask users questions to extract more information regarding instructions. For example, the instruction 'Reformulate the following text:' can have various underlying parameters such as tone, style, structure... A possible clarifying question could be "Do you want to conserve the text structure or do you want information to be re-arranged?" This interaction can help clarify and refine instructions to better align with individual user preferences.

9.3 Open discussion about LLMs and IR

ChatGPT still doesn't care about... Recent advancements in LMs have significantly propelled the capabilities of conversational systems, particularly in tasks like QA, summarization, and instruction-following applications. Models such as Bard, GPT-4, and Llama have become prominent in this sphere. However, these systems often fall short in CS contexts, particularly due to their limitations in conducting interactive, nuanced dialogues and handling exploratory search queries. These models are predominantly trained for reactive tasks. For example, Llama2-chat was trained by filling the context with concatenations of instructions and answers (Touvron et al. 2023). Therefore, retrieval augmented models generate responses immediately after retrieving documents, without engaging in further interactions with users (Shuster et al. 2022b). Also including those fine-tuned with the RLHF method, tend to provide broad, well-structured, but often lengthy and non-specific responses. Such answers may lack natural conversational flow and may not align with users' specific intentions or needs. However, a dialogue is not a sequence of unilateral instructions. Figure 3.5 illustrates a comparison between reactive and interactive systems. The reactive system offers a comprehensive response, while the interactive system engages in multiple interactions with the user in order to gain a better understanding of their context. This ability to interact with users is essential to refine the need by resolving ambiguous instructions or questions and for exploratory searches. More generally, these systems should be trained to probe users to better understand user's motivations, contexts, and needs. This would improve alignment between humans and machines and provide more natural interactions. Planning may be necessary in order to optimize the sequence of interactions with users and better model the user's needs.

Exploratory Search The IR community highlights the distinction between known-item searches (where users have a specific or factual query, e.g. "where was Nicolas Tesla born?") and exploratory searches (where users' goals are more nebulous). The latter scenario often involves users browsing through various topics without a precise objective, necessitating a system capable of guiding the exploration efficiently and intuitively (eg: 'What is the latest news from around the world'). Unfortunately, existing LLMs are not adequately equipped to handle such exploratory queries due to their lack of training in interactive IR and user intent modeling (Zhai 2020). Enhancing LLMs requires training them to reason over documents, reformulate queries, provide a synopsis of different results or sources, recommend topics, ask for feedback, and predict future users' inquiries.. This improvement is especially crucial for handling non-factual or exploratory searches.

BIBLIOGRAPHY

- Adolphs, Leonard, Michelle Chen Huebscher, Christian Buck, Sertan Girgin, Olivier Bachem, Massimiliano Ciaramita, and Thomas Hofmann (2022). *Decoding a Neural Retriever's Latent Space for Query Suggestion* (cit. on p. 95).
- Aliannejadi, Mohammad, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev (2021). "Building and Evaluating Open-Domain Dialogue Corpora with Clarifying Questions". In: (cit. on p. 85).
- Aliannejadi, Mohammad, Hamed Zamani, Fabio Crestani, and W. Bruce Croft (2019). "Asking Clarifying Questions in Open-Domain Information-Seeking Conversations". In: Paris, France (cit. on pp. 5, 7, 8, 45, 46, 52, 53, 55, 65, 84, 85).
- Anand, Avishek, Lawrence Cavedon, Matthias Hagen, Hideo Joho, Mark Sanderson, and Benno Stein (2020). "Conversational Search - A Report from Dagstuhl Seminar 19461". In: (cit. on pp. 5, 41).
- Arguello, Jaime, Adam Ferguson, Emery Fine, Bhaskar Mitra, Hamed Zamani, and Fernando Diaz (2021). "Tip of the Tongue Known-Item Retrieval: A Case Study in Movie Identification". In: Canberra ACT, Australia (cit. on p. 4).
- Asri, Layla El, Jing He, and Kaheer Suleman (2016). *A Sequence-to-Sequence Model for User Simulation in Spoken Dialogue Systems* (cit. on p. 48).
- Azzopardi, Leif, Mateusz Dubiel, Martin Halvey, and Jeffery Dalton (2018). "Conceptualizing agent-human interactions during the conversational search process". In: (cit. on p. 42).
- Bahdanau, Dzmitry, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio (2016). "An Actor-Critic Algorithm for Sequence Prediction". In: (cit. on p. 27).
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". English (US). In: (cit. on p. 18).
- Bai, Yuntao, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer,

- Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan (2022). *Constitutional AI: Harmlessness from AI Feedback* (cit. on p. 27).
- Banerjee, Satantjeet and Alon Lavie (2005). "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: (cit. on pp. 90, 110).
- Barifah, Maram and Monica Landoni (2020). "Emotions associated with failed searches in a digital library". In: (cit. on p. 4).
- Belkin, Nicholas J. and W. Bruce Croft (1992). "Information Filtering and Information Retrieval: Two Sides of the Same Coin?" In: (cit. on p. 48).
- Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell (2021). "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: Virtual Event, Canada (cit. on p. 5).
- Bengio, Samy, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer (2015). "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks". In: (cit. on p. 32).
- Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent (2000). "A Neural Probabilistic Language Model". In: (cit. on pp. 14, 15).
- Bhavnani, Suresh K. (2001). "Important Cognitive Components of Domain-Specific Search Knowledge". In: (cit. on p. 4).
- Bi, Keping, Qingyao Ai, and W. Bruce Croft (2021). "Asking Clarifying Questions Based on Negative Feedback in Conversational Search". In: Virtual Event, Canada (cit. on p. 45).
- Bordes, Antoine, Y-Lan Boureau, and Jason Weston (2017). "Learning End-to-End Goal-Oriented Dialog". In: (cit. on p. 42).
- Bordes, Antoine, Sumit Chopra, and Jason Weston (2014). "Question Answering with Subgraph Embeddings". In: (cit. on p. 42).
- Borgeaud, Sebastian, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre (2022a). "Improving Language Models by Retrieving from Trillions of Tokens". In: (cit. on p. 9).
- Borgeaud, Sebastian, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau,

- Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre (2022b). “Improving Language Models by Retrieving from Trillions of Tokens”. In: (cit. on p. 120).
- Braslavski, Pavel, Denis Savenkov, Eugene Agichtein, and Alina Dubatovka (2017). “What Do You Mean Exactly? Analyzing Clarification Questions in CQA”. In: Oslo, Norway (cit. on p. 45).
- Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich (2023). *RT-1: Robotics Transformer for Real-World Control at Scale* (cit. on p. 16).
- Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020a). “Language Models are Few-Shot Learners”. In: (cit. on pp. 5, 16).
- Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020b). “Language Models are Few-Shot Learners”. In: (cit. on pp. 16, 26, 28, 32, 120, 121).
- Bubeck, Sebastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Har-

- sha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang (2023). *Sparks of Artificial General Intelligence: Early experiments with GPT-4* (cit. on pp. 16, 121).
- Burges, Chris, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender (2005). "Learning to Rank Using Gradient Descent". In: Bonn, Germany (cit. on p. 55).
- Cai, Fei, Ridho Reinanda, and Maarten De Rijke (2016). "Diversifying Query Auto-Completion". In: (cit. on p. 55).
- Câmara, Arthur, David Maxwell, and Claudia Hauff (2022). "Searching, Learning, and Subtopic Ordering: A Simulation-based Analysis". In: (cit. on pp. 56, 62, 76).
- Chen, Danqi, Adam Fisch, Jason Weston, and Antoine Bordes (2017). "Reading Wikipedia to Answer Open-Domain Questions". In: (cit. on p. 47).
- Chen, Jerry Zikun, Shih Yuan Yu, and Haoran Wang (2020). "Exploring Fluent Query Reformulations with Text-to-Text Transformers and Reinforcement Learning". In: (cit. on p. 55).
- Chen, Lili, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch (2021). *Decision Transformer: Reinforcement Learning via Sequence Modeling*. cite arxiv:2106.01345Comment: First two authors contributed equally. Last two authors advised equally (cit. on p. 16).
- Chen, Wanyu, Fei Cai, Honghui Chen, and Maarten de Rijke (2017). "Personalized Query Suggestion Diversification". In: Shinjuku, Tokyo, Japan (cit. on p. 7).
- Choi, Eunsol, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer (2018). "QuAC: Question Answering in Context". In: (cit. on p. 47).
- Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and

- Noah Fiedel (2022). *PaLM: Scaling Language Modeling with Pathways* (cit. on p. 28).
- Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei (2022). *Scaling Instruction-Finetuned Language Models* (cit. on p. 110).
- Chung, Junyoung, Çağlar Gülçehre, KyungHyun Cho, and Yoshua Bengio (2014). “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”. In: (cit. on p. 15).
- Clark, Peter, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord (2018). *Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge* (cit. on p. 27).
- Clarke, Charles L. A., Nick Craswell, and Ian Soboroff (2009a). “Overview of the TREC 2009 Web Track”. In: *Text Retrieval Conference* (cit. on p. 47).
- Clarke, Charles L. A., Nick Craswell, and Ian Soboroff (2009b). “Overview of the TREC 2009 Web Track”. In: (cit. on p. 82).
- Cronen-Townsend, Steve and W. Bruce Croft (2002). “Quantifying Query Ambiguity”. In: San Diego, California (cit. on p. 4).
- Culpepper, J Shane, Fernando Diaz, and Mark D Smucker (2018). “Research Frontiers in Information Retrieval: Report from the Third Strategic Workshop on Information Retrieval in Lorne (SWIRL 2018)”. In: (cit. on pp. 4, 41).
- Dalton, Jeffrey, Sophie Fischer, Paul Owoicho, Filip Radlinski, Federico Rossetto, Johanne R. Trippas, and Hamed Zamani (2022). “Conversational Information Seeking: Theory and Application”. In: Madrid, Spain (cit. on pp. 5, 46).
- Dalton, Jeffrey, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan (2020). “CAsT-19: A Dataset for Conversational Information Seeking”. In: (cit. on p. 47).
- Deng, Xiang, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su (2023). *Mind2Web: Towards a Generalist Agent for the Web* (cit. on p. 135).
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: (cit. on pp. 16, 21, 23, 26).
- Dhingra, Bhuwan, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen (2019). “Handling Divergent Reference Texts when Evaluating Table-to-Text Generation”. In: (cit. on p. 31).

- Ding, Ning, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou (2023). “Enhancing Chat Language Models by Scaling High-quality Instructional Conversations”. In: (cit. on p. 133).
- Donati, Alice Martin, Guillaume Quispe, Charles Ollion, Sylvain Le Corff, Florian Strub, and Olivier Pietquin (2021). “Learning Natural Language Generation from Scratch”. In: (cit. on p. 27).
- Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: (cit. on p. 16).
- Dou, Zhicheng, Zhengbao Jiang, Sha Hu, Ji-Rong Wen, and Ruihua Song (2016). “Automatically Mining Facets for Queries from Their Search Results”. In: (cit. on p. 82).
- Dziri, Nouha, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy (2022). “On the Origin of Hallucinations in Conversational Models: Is it the Datasets or the Models?” In: (cit. on p. 29).
- Eckert, Wieland, Esther Levin, and Roberto Pieraccini (1997). “User modeling for spoken dialogue system evaluation”. In: (cit. on p. 70).
- Elgohary, Ahmed, Denis Peskov, and Jordan Boyd-Graber (2019). “Can You Unpack That? Learning to Rewrite Questions-in-Context”. In: (cit. on p. 47).
- Erbacher, Pierre, Ludovic Denoyer, and Laure Soulier (2022a). *Interactive Query Clarification and Refinement via User Simulation*. arXiv: 2205.15918 [cs.IR] (cit. on pp. 10, 48, 51, 64, 65, 71, 73, 84, 85).
- Erbacher, Pierre, Louis Falissard, Vincent Guigue, and Laure Soulier (2023a). *Navigating Uncertainty: Optimizing API Dependency for Hallucination Reduction in Closed-Book QA* (cit. on pp. 11, 119).
- Erbacher, Pierre, Jian-Yun Nie, Philippe Preux, and Laure Soulier (2023b). *Augmenting Ad-Hoc IR Dataset for Interactive Conversational Search*. arXiv: 2311.06119 [cs.IR] (cit. on pp. 10, 79).
- Erbacher, Pierre, Jian-Yun Nie, Philippe Preux, and Laure Soulier (2023c). *PAQA: Toward ProActive Open-Retrieval Question Answering* (cit. on pp. 11, 103).
- Erbacher, Pierre and Laure Soulier (2023). *CIRCLE: Multi-Turn Query Clarifications with Reinforcement Learning*. arXiv: 2311.02737 [cs.IR] (cit. on pp. 10, 63).
- Erbacher, Pierre, Laure Soulier, and Ludovic Denoyer (2022b). *State of the Art of User Simulation approaches for conversational information retrieval*. arXiv: 2201.03435 [cs.IR] (cit. on p. 10).

- Erbacher, Pierre, Laure Soulier, and Ludovic Denoyer (2022c). "State of the Art of User Simulation approaches for conversational information retrieval". In: (cit. on pp. 35, 70).
- Fan, Angela, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli (2019). "ELI5: Long Form Question Answering". In: (cit. on p. 47).
- Formal, Thibault, Benjamin Piwowarski, and Stéphane Clinchant (2021). "SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking". In: (cit. on p. 39).
- Fu, Zuohui, Yikun Xian, Yaxin Zhu, Yongfeng Zhang, and Gerard de Melo (2020). *COOKIE: A Dataset for Conversational Recommendation over Knowledge Graphs in E-commerce* (cit. on pp. 48, 84, 85, 93).
- Gao, Chongming, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua (2022). "KuaiRec: A Fully-Observed Dataset and Insights for Evaluating Recommender Systems". In: Atlanta, GA, USA (cit. on pp. 48, 84, 85).
- Gao, Yifan, Henghui Zhu, Patrick Ng, Cicero Nogueira dos Santos, Zhiguo Wang, Feng Nan, Dejiao Zhang, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang (2021). "Answering Ambiguous Questions through Generative Evidence Fusion and Round-Trip Prediction". In: (cit. on p. 115).
- Garigliotti, Dario and Krisztian Balog (2017). "Generating Query Suggestions to Support Task-Based Search". In: Shinjuku, Tokyo, Japan (cit. on p. 7).
- Goodman, Joshua (2001). "A Bit of Progress in Language Modeling". In: *CoRR* cs.CL/0108005. URL: <https://arxiv.org/abs/cs/0108005> (cit. on p. 14).
- Guerreiro, Nuno M., Elena Voita, and André Martins (2023). "Looking for a Needle in a Haystack: A Comprehensive Study of Hallucinations in Neural Machine Translation". In: (cit. on p. 125).
- Guo, Huadong, Lizhe Wang, Fang Chen, and Dong Liang (2014). "Scientific big data and Digital Earth". In: (cit. on p. 4).
- Guo, J., Xueqi Cheng, Gu Xu, and Xiaofei Zhu (2011). "Intent-aware query similarity". In: (cit. on p. 7).
- Guo, Jiafeng, Xueqi Cheng, Gu Xu, and Xiaofei Zhu (2011). "Intent-Aware Query Similarity". In: Glasgow, Scotland, UK (cit. on p. 7).
- Guu, Kelvin, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang (2020). "REALM: Retrieval-Augmented Language Model Pre-Training". In: (cit. on pp. 120, 125).
- Hai, Nam Le, Thomas Gerald, Thibault Formal, Jian-Yun Nie, Benjamin Piwowarski, and Laure Soulier (2023). *CoSPLADE: Contextualizing SPLADE for Conversational Information Retrieval* (cit. on p. 47).

- Hashemi, Helia, Hamed Zamani, and W. Bruce Croft (2020). "Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search". In: Virtual Event, China (cit. on p. 45).
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition". In: Las Vegas, NV, USA (cit. on p. 18).
- He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen (2020). "DeBERTa: Decoding-enhanced bert with disentangled attention". In: (cit. on p. 110).
- Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt (2021). *Measuring Massive Multitask Language Understanding* (cit. on p. 28).
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: (cit. on p. 15).
- Hofstätter, Sebastian, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury (2021). "Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling". In: (cit. on pp. 39, 56, 58, 72).
- Holtzman, Ari, Jan Buys, Maxwell Forbes, and Yejin Choi (2019). "The Curious Case of Neural Text Degeneration". In: (cit. on pp. 66, 67).
- Hu, Edward J, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen (2022). "LoRA: Low-Rank Adaptation of Large Language Models". In: (cit. on p. 125).
- Jansen, Bernard J., Amanda Spink, and Tefko Saracevic (2000). "Real life, real users, and real needs: A study and analysis of user queries on the Web". English (US). In: (cit. on p. 4).
- Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung (2023). "Survey of Hallucination in Natural Language Generation". In: (cit. on pp. 30, 31, 120).
- Jia, Meihuizi, Ruixue Liu, Peiyang Wang, Yang Song, Zexi Xi, Haobin Li, Xin Shen, Meng Chen, Jinhui Pang, and Xiaodong He (2022). "E-ConvRec: A Large-Scale Conversational Recommendation Dataset for E-Commerce Customer Service". In: (cit. on p. 93).
- Johnson, Jeff, Matthijs Douze, and Herve Jegou (2021). "Billion-Scale Similarity Search with GPUs". In: (cit. on pp. 58, 72).
- Joshi, Mandar, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer (2017a). "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension". In: (cit. on pp. 10, 121).
- Joshi, Mandar, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer (2017b). "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension". In: (cit. on p. 123).

- Kamvar, Maryam and Shumeet Baluja (2006). "A Large Scale Study of Wireless Search Behavior: Google Mobile Search". In: Montréal, Québec, Canada (cit. on p. 4).
- Kang, Dongyeop, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston (2019). "Recommendation as a Communication Game: Self-Supervised Bot-Play for Goal-oriented Dialogue". In: (cit. on pp. 48, 84, 85).
- Karpukhin, Vladimir, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih (2020). "Dense Passage Retrieval for Open-Domain Question Answering". In: (cit. on pp. 9, 26, 39, 109, 110).
- Keyvan, Kimiya and Jimmy Xiangji Huang (2022). "How to Approach Ambiguous Queries in Conversational Search: A Survey of Techniques, Approaches, Tools, and Challenges". In: (cit. on p. 5).
- Kiesel, Johannes, Arefeh Bahrami, Benno Stein, Avishek Anand, and Matthias Hagen (2018). "Toward Voice Query Clarification". In: Ann Arbor, MI, USA (cit. on p. 45).
- Komatani, Kazunori, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno (2005). "User modeling in spoken dialogue systems to generate flexible guidance". In: (cit. on p. 70).
- Kong, Weize and James Allan (2013). "Extracting Query Facets from Search Results". In: Dublin, Ireland (cit. on p. 82).
- Kreyssig, Florian, Iñigo Casanueva, Paweł Budzianowski, and Milica Gašić (2018). "Neural User Simulation for Corpus-based Policy Optimisation of Spoken Dialogue Systems". In: (cit. on p. 48).
- Kwiatkowski, Tom, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. (2019a). "Natural questions: a benchmark for question answering research". In: (cit. on p. 106).
- Kwiatkowski, Tom, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov (2019b). "Natural Questions: A Benchmark for Question Answering Research". In: (cit. on pp. 10, 99, 121, 123).
- Lee, Katherine, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini (2022). "Deduplicating Training Data Makes Language Models Better". In: (cit. on p. 31).
- Lee, Kenton, Ming-Wei Chang, and Kristina Toutanova (2019a). "Latent Retrieval for Weakly Supervised Open Domain Question Answering". In: (cit. on p. 120).

- Lee, Kenton, Ming-Wei Chang, and Kristina Toutanova (2019b). “Latent Retrieval for Weakly Supervised Open Domain Question Answering”. In: (cit. on p. 123).
- Lee, Nayeon, Yejin Bang, Andrea Madotto, and Pascale Fung (2021). “Towards Few-shot Fact-Checking via Perplexity”. In: (cit. on p. 125).
- Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela (2020a). “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. In: Vancouver, BC, Canada (cit. on p. 9).
- Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela (2020b). “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. In: Vancouver, BC, Canada (cit. on p. 120).
- Li, Raymond, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal (2018). “Towards Deep Conversational Recommendations”. In: Montréal, Canada (cit. on p. 93).
- Li, Xiujun, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz (2017). “End-to-End Task-Completion Neural Dialogue Systems”. In: (cit. on p. 48).
- Lin, Chin-Yew (2004). “ROUGE: A Package for Automatic Evaluation of Summaries”. In: (cit. on p. 110).
- Lin, Jimmy, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira (2021). “Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations”. In: (cit. on p. 95).
- Lin, Sheng-Chieh, Jheng-Hong Yang, and Jimmy J. Lin (2020). “TREC 2020 Notebook: CAsT Track”. In: (cit. on p. 55).
- Liu, Mingtong, Erguang Yang, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen (2020). “A Learning-Exploring Method to Generate Diverse Paraphrases with Multi-Objective Deep Reinforcement Learning”. In: (cit. on pp. 27, 68).
- Liu, Xiao, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang (2023). *AgentBench: Evaluating LLMs as Agents* (cit. on p. 135).
- Longpre, Shayne, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh (2021). “Entity-Based Knowledge Conflicts in Question Answering”. In: (cit. on p. 32).

- MacAvaney, Sean, Craig Macdonald, Roderick Murray-Smith, and Iadh Ounis (2021). "Intent5: Search Result Diversification using Causal Language Models". In: (cit. on p. 55).
- Madotto, Andrea, Zihan Liu, Zhaojiang Lin, and Pascale Fung (2020). *Language Models as Few-Shot Learner for Task-Oriented Dialogue Systems* (cit. on p. 32).
- Mahowald, Kyle, Anna A. Ivanova, Idan Asher Blank, Nancy G. Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko (2023). "Dissociating language and thought in large language models: a cognitive perspective". In: (cit. on pp. 32, 135).
- Mangrulkar, Sourab, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul (2022). *PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods*. <https://github.com/huggingface/peft> (cit. on pp. 125, 126).
- Meng, Rui, Xingdi Yuan, Tong Wang, Sanqiang Zhao, Adam Trischler, and Daqing He (2020). "An Empirical Study on Neural Keyphrase Generation". In: (cit. on p. 66).
- Metzler, Donald, Yi Tay, Dara Bahri, and Marc Najork (2021). "Rethinking Search: Making Domain Experts out of Dilettantes". In: (cit. on p. 30).
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: (cit. on p. 15).
- Min, Sewon, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer (2020). "AmbigQA: Answering Ambiguous Open-domain Questions". In: (cit. on pp. 104, 106, 110, 116).
- Momennejad, Ida, Hosein Hasanbeig, Felipe Vieira Frujeri, Hiteshi Sharma, Microsoft Research Redmond, sharma Robert Osazuwa, Ness, Nebojsa Jojic, Hamid Palangi, and Jonathan Larson (2023). "Evaluating Cognitive Maps in Large Language Models with CogEval: No Emergent Planning". In: (cit. on pp. 32, 135).
- Nakano, Reiichiro, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman (2021). "WebGPT: Browser-assisted question-answering with human feedback". In: (cit. on p. 9).
- Nakano, Reiichiro, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman (2022). *WebGPT: Browser-assisted question-answering with human feedback* (cit. on p. 120).

- Nguyen, Tri, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng (2016a). “MS MARCO: A Human Generated MACHINE Reading COMprehension Dataset”. In: (cit. on pp. 60, 110).
- Nguyen, Tri, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng (2016b). “MS MARCO: A Human Generated MACHINE Reading COMprehension Dataset”. In: (cit. on pp. 70, 88).
- Nogueira, Rodrigo, Jannis Bulian, and Massimiliano Ciaramita (2019a). “Multi-agent query reformulation: Challenges and the role of diversity”. In: (cit. on p. 27).
- Nogueira, Rodrigo, Wei Yang, Kyunghyun Cho, and Jimmy J. Lin (2019b). “Multi-Stage Document Ranking with BERT”. In: (cit. on pp. 57, 70).
- Nogueira, Rodrigo, Wei Yang, Jimmy J. Lin, and Kyunghyun Cho (2019c). “Document Expansion by Query Prediction”. In: (cit. on p. 55).
- Nogueira, Rodrigo Frassetto and Kyunghyun Cho (2017). “Task-Oriented Query Reformulation with Reinforcement Learning”. In: (cit. on p. 27).
- OpenAI (2023). *GPT-4 Technical Report* (cit. on pp. 16, 28).
- Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe (2022a). *Training language models to follow instructions with human feedback* (cit. on p. 27).
- Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe (2022b). “Training language models to follow instructions with human feedback”. In: (cit. on pp. 27, 64).
- Over, Paul (2001). “The TREC interactive track: an annotated bibliography”. In: *Interactivity at the Text Retrieval Conference (TREC)* (cit. on p. 82).
- Ozertem, Umut, Olivier Chapelle, Pinar Donmez, and Emre Velipasaoglu (2012). “Learning to Suggest: A Machine Learning Framework for Ranking Query Suggestions”. In: *Portland, Oregon, USA* (cit. on p. 7).
- Pal, Dipasree, Mandar Mitra, and Kalyankumar Datta (2013). “Query Expansion Using Term Distribution and Term Association”. In: (cit. on p. 55).
- Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002a). “BLEU: A Method for Automatic Evaluation of Machine Translation”. In: *Philadelphia, Pennsylvania* (cit. on p. 27).
- Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002b). “Bleu: a Method for Automatic Evaluation of Machine Translation”. In: (cit. on p. 110).

- Parikh, Ankur, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das (2020). "ToTTo: A Controlled Table-To-Text Generation Dataset". In: (cit. on p. 31).
- Peng, Baolin, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong (2018). "Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning". In: (cit. on p. 48).
- Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "GloVe: Global Vectors for Word Representation". In: (cit. on p. 15).
- Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep Contextualized Word Representations". In: (cit. on p. 15).
- Petroni, Fabio, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller (2019). "Language Models as Knowledge Bases?" In: (cit. on pp. 30, 32).
- Pfeiffer, Jonas, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych (2020). "AdapterHub: A Framework for Adapting Transformers". In: (cit. on pp. 125, 126).
- Pietquin, Olivier (2004). "A Framework for Unsupervised Learning of Dialogue Strategies". PhD thesis. Faculte Polytechnique de Mons, TCTS Lab (Belgique) (cit. on p. 70).
- Pradeep, Ronak, Rodrigo Nogueira, and Jimmy Lin (2021). "The Expando-Monoduo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models". In: (cit. on pp. 39, 58, 94, 110).
- Qin, Yujia, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun (2023). *ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs* (cit. on p. 135).
- Qu, Chen, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer (2020). "Open-Retrieval Conversational Question Answering". In: Virtual Event, China (cit. on p. 93).
- Qu, Yingqi, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang (2021). "RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering". In: (cit. on p. 89).
- Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving language understanding by generative pre-training". In: (cit. on pp. 16, 24).

- Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). "Language models are unsupervised multitask learners". In: (cit. on pp. 66, 67).
- Radlinski, Filip and Nick Craswell (2017). "A Theoretical Framework for Conversational Search". In: Oslo, Norway (cit. on pp. 5, 42).
- Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020a). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer* (cit. on pp. 25, 125).
- Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020b). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: (cit. on pp. 26, 55, 58, 91, 94).
- Rajpurkar, Pranav, Robin Jia, and Percy Liang (2018). "Know What You Don't Know: Unanswerable Questions for SQuAD". In: (cit. on p. 47).
- Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: (cit. on p. 110).
- Rame, Alexandre, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord (2023). *Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards* (cit. on p. 135).
- Ranzato, Marc'Aurelio, Sumit Chopra, Michael Auli, and Wojciech Zaremba (2016). "Sequence Level Training with Recurrent Neural Networks". In: (cit. on p. 27).
- Rao, Sudha and Hal Daumé III (2019). "Answer-based Adversarial Training for Generating Clarification Questions". In: (cit. on p. 55).
- Reddy, Siva, Danqi Chen, and Christopher D. Manning (2019). "CoQA: A Conversational Question Answering Challenge". In: (cit. on p. 47).
- Reimers, Nils and Iryna Gurevych (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: (cit. on pp. 84, 88, 90, 91).
- Roberts, Adam, Colin Raffel, and Noam Shazeer (2020). "How Much Knowledge Can You Pack Into the Parameters of a Language Model?" In: (cit. on pp. 30, 32, 126).
- Robertson, Stephen and Hugo Zaragoza (2009). "The probabilistic relevance framework: BM25 and beyond". In: (cit. on p. 38).
- Roit, Paul, Johan Ferret, Lior Shani, Roei Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Leonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor (2023). *Factually Consistent Summarization via Reinforcement Learning with Textual Entailment Feedback* (cit. on p. 27).

- Salle, Alexandre, Shervin Malmasi, Oleg Rokhlenko, and Eugene Agichtein (2021). "Studying the Effectiveness of Conversational Search Refinement Through User Simulation". In: (cit. on pp. 46, 48, 81, 84, 85).
- Salton, G., A. Wong, and C. S. Yang (1975). "A Vector Space Model for Automatic Indexing". In: (cit. on p. 38).
- Sanderson, Mark (2008). "Ambiguous Queries: Test Collections Need More Sense". In: Singapore, Singapore (cit. on p. 4).
- Sanderson, Mark and W. Bruce Croft (2012). "The History of Information Retrieval Research". In: (cit. on p. 37).
- Santos, Rodrygo L. T., Craig MacDonald, and Iadh Ounis (2012). "Learning to rank query suggestions for adhoc and diversity search". In: (cit. on p. 7).
- Schatzmann, Jost, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young (2007). "Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System". In: (cit. on p. 48).
- Schatzmann, Jost, Karl Weilhammer, Matt Stuttle, and Steve Young (2006). "A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies". In: (cit. on p. 70).
- Scheffler, Konrad and Steve Young (2000). "Probabilistic simulation of human-machine dialogues". In: (cit. on p. 70).
- Schick, Timo, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom (2023). *Toolformer: Language Models Can Teach Themselves to Use Tools* (cit. on pp. 9, 120, 135).
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). "Proximal policy optimization algorithms". In: (cit. on p. 67).
- Schwartz, Bennett L. and Janet Metcalfe (2011). "Tip-of-the-tongue (TOT) states: retrieval, behavior, and experience". In: (cit. on p. 4).
- Sekulić, Ivan, Mohammad Aliannejadi, and Fabio Crestani (2021). "Towards Facet-Driven Generation of Clarifying Questions for Conversational Search". In: Virtual Event, Canada (cit. on pp. 46, 81, 84, 88, 90).
- Sekulić, Ivan, Mohammad Aliannejadi, and Fabio Crestani (2022a). "Evaluating Mixed-Initiative Conversational Search Systems via User Simulation". In: Virtual Event, AZ, USA (cit. on p. 48).
- Sekulić, Ivan, Mohammad Aliannejadi, and Fabio Crestani (2022b). "Exploiting Document-Based Features for Clarification in Conversational Search". In: (cit. on pp. 46, 82, 84).
- Shah, Chirag and Emily M. Bender (2022). "Situating Search". In: Regensburg, Germany (cit. on p. 5).
- Shannon, Claude Elwood (1948). "A Mathematical Theory of Communication". In: (visited on 04/22/2003) (cit. on p. 97).

- Sharma, Prafull and Yingbo Li (2019). *Self-Supervised Contextual Keyword and Keyphrase Retrieval with Self-Labeling* (cit. on p. 82).
- Shazeer, Noam and Mitchell Stern (2018). "Adafactor: Adaptive Learning Rates with Sublinear Memory Cost". In: (cit. on pp. 91, 94).
- Shuster, Kurt, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston (2022a). *Language Models that Seek for Knowledge: Modular Search& Generation for Dialogue and Prompt Completion* (cit. on p. 5).
- Shuster, Kurt, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston (2022b). *BlenderBot 3: a deployed conversational agent that continually learns to responsibly engage* (cit. on pp. 9, 120, 135, 136).
- Sparck Jones, Karen (1988). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". In: (cit. on p. 38).
- Stiennon, Nisan, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano (2020). "Learning to summarize with human feedback". In: (cit. on p. 68).
- Strudel, Robin, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond (2022). *Self-conditioned Embedding Diffusion for Text Generation* (cit. on p. 32).
- Su, Jianlin, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu (2022). *RoFormer: Enhanced Transformer with Rotary Position Embedding* (cit. on p. 21).
- Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014a). "Sequence to Sequence Learning with Neural Networks". In: (cit. on p. 124).
- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014b). "Sequence to Sequence Learning with Neural Networks". In: (cit. on p. 22).
- Thakur, Nandan, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych (2021). "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models". In: (cit. on pp. 88, 95).
- Thoppilan, Romal, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel

- Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le (2022a). *LaMDA: Language Models for Dialog Applications* (cit. on p. 5).
- Thoppilan, Romal, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le (2022b). *LaMDA: Language Models for Dialog Applications* (cit. on p. 120).
- Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models* (cit. on pp. 28, 29, 136).

- Trippas, Johanne R., Damiano Spina, Paul Thomas, Mark Sanderson, Hideo Joho, and Lawrence Cavedon (2020). "Towards a model for spoken conversational search". In: (cit. on p. 5).
- Vakulenko, Svitlana, Ilya Markov, and Maarten de Rijke (2017). "Conversational exploratory search via interactive storytelling". In: (cit. on p. 43).
- Valmeekam, Karthik, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati (2023). *Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change)* (cit. on pp. 32, 135).
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: (cit. on pp. 16, 17, 19, 21, 22).
- Vijayakumar, Ashwin K., Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra (2016). "Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models". In: (cit. on pp. 55, 60, 66, 67).
- Voorhees, Ellen (2019). "The Evolution of Cranfield". In: (cit. on p. 39).
- Wang, Yizhong, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi (2023). *Self-Instruct: Aligning Language Models with Self-Generated Instructions* (cit. on p. 133).
- Wang, Zekun, Ge Zhang, Kexin Yang, Ning Shi, Wangchunshu Zhou, Shaochun Hao, Guangzheng Xiong, Yizhi Li, Mong Yuan Sim, Xiuying Chen, Qingqing Zhu, Zhenzhu Yang, Adam Nik, Qi Liu, Chenghua Lin, Shi Wang, Ruibo Liu, Wenhu Chen, Ke Xu, Dayiheng Liu, Yike Guo, and Jie Fu (2023). *Interactive Natural Language Processing* (cit. on p. 134).
- Webber, William, Alistair Moffat, and Justin Zobel (2010). "A Similarity Measure for Indefinite Rankings". In: (cit. on pp. 70, 100).
- Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus (2022a). *Emergent Abilities of Large Language Models* (cit. on pp. 16, 121).
- Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus (2022b). *Emergent Abilities of Large Language Models* (cit. on pp. 28, 30, 31).
- Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou (2022c). "Chain of Thought Prompting Elicits Reasoning in Large Language Models". In: (cit. on p. 32).
- White, Ryan (2013). "Beliefs and Biases in Web Search". In: Dublin, Ireland (cit. on p. 4).

- White, Ryen W., Susan Dumais, and Jaime Teevan (2008). "How Medical Expertise Influences Web Search Interaction". In: Singapore, Singapore (cit. on p. 4).
- Wildemuth, Barbara M. (2004). "The Effects of Domain Knowledge on Search Tactic Formulation". In: (cit. on p. 4).
- Williams, Ronald J. and David Zipser (1989a). "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". In: (cit. on p. 25).
- Williams, Ronald J. and David Zipser (1989b). "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". In: (cit. on p. 91).
- Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush (2020). "Transformers: State-of-the-Art Natural Language Processing". In: (cit. on p. 110).
- Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush (2019a). *HuggingFace's Transformers: State-of-the-art Natural Language Processing* (cit. on pp. 91, 94).
- Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew (2019b). "HuggingFace's Transformers: State-of-the-art Natural Language Processing". In: (cit. on p. 72).
- Workshop, BigScience et al. (2023). *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model* (cit. on p. 28).
- Wu, Bin, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu (2018). "Query Suggestion with Feedback Memory Network". In: (cit. on p. 7).
- Wu, Fangzhao, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou (2020). "MIND: A Large-scale Dataset for News Recommendation". In: (cit. on pp. 48, 84, 85).
- Xiao, Guangxuan, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis (2023). *Efficient Streaming Language Models with Attention Sinks* (cit. on p. 32).
- Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan (2023). *Tree of Thoughts: Deliberate Problem Solving with Large Language Models* (cit. on p. 32).
- Ye, Jiacheng, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang (2021). "One2Set: Generating Diverse Keyphrases as a Set". In: (cit. on pp. 55, 66, 75).

- Yuan, Xingdi, Tong Wang, Rui Meng, Khushboo Thaker, Daqing He, and Adam Trischler (2018). "Generating Diverse Numbers of Diverse Keyphrases". In: (cit. on pp. 66, 67).
- Zamani, Hamed, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck (2020a). "Generating Clarifying Questions for Information Retrieval". In: (cit. on pp. 7, 8, 46, 47, 54, 81, 84).
- Zamani, Hamed, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell (2020b). "MIMICS: A Large-Scale Data Collection for Search Clarification". In: Virtual Event, Ireland (cit. on p. 7).
- Zamani, Hamed, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N. Bennett, Nick Craswell, and Susan T. Dumais (2020c). "Analyzing and Learning from User Interactions for Search Clarification". In: (cit. on pp. 5, 56).
- Zamani, Hamed, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N. Bennett, Nick Craswell, and Susan T. Dumais (2020d). "Analyzing and Learning from User Interactions for Search Clarification". In: (cit. on p. 56).
- Zamani, Hamed, Johanne R Trippas, Jeff Dalton, Filip Radlinski, et al. (2023). "Conversational information seeking". In: (cit. on pp. 5, 41, 45).
- Zamani, Hamed, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski (2022). "Conversational Information Seeking". In: (cit. on p. 5).
- Zhai, ChengXiang (2020). "Interactive Information Retrieval: Models, Algorithms, and Evaluation". In: (cit. on pp. 42, 137).
- Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen (2023). *A Survey of Large Language Models* (cit. on pp. 23, 28).
- Zhou, Chunting, Pengfei Liu, Puxin Xu, Srinu Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy (2023). *LIMA: Less Is More for Alignment* (cit. on p. 26).
- Zhou, Chunting, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad (2021). "Detecting Hallucinated Content in Conditional Neural Sequence Generation". In: (cit. on p. 31).
- Zhou, Kun, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen (2020). "Towards Topic-Guided Conversational Recommender System". In: (cit. on pp. 48, 84, 85, 93).

Ziegler, Daniel M., Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving (2019). "Fine-Tuning Language Models from Human Preferences". In: (cit. on p. 27).

Zou, Jie, Aixin Sun, Cheng Long, Mohammad Aliannejadi, and Evangelos Kanoulas (2023). "A user study on asking clarifying questions in web search". In: (cit. on p. 45).

