Designing and Diagnosing Models for Conversational Search and Recommendation

Penha, G.

DOI
10.4233/uuid:acdfb704-6310-4b28-b884-4bd3e78b3f84

Publication date
2023

Document Version
Final published version

Citation (APA)
Penha, G. (2023). *Designing and Diagnosing Models for Conversational Search and Recommendation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:acdfb704-6310-4b28-b884-4bd3e78b3f84

Designing and Diagnosing Models for Conversational Search and Recommendation

Designing and Diagnosing Models for Conversational Search and Recommendation

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op woensdag 24 mei 2023 om 12.30 uur

door

Gustavo PENHA

Master of Science in Computer Science,
Universidade Federal de Minas Gerais, Brazilie,
geboren te Belo Horizonte, Brazilie.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. ir. G.J.P.M Houben
promotor: Dr. C. Hauff

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. G.J.P.M Houben	Delft University of Technology
Dr. C. Hauff	Delft University of Technology
Onafhankelijke leden:	
Prof. dr. L Flek	University of Marburg
Prof. dr. K. Balog	University of Stavanger
Prof. dr. E. Kanoulas	University of Amsterdam
Prof. dr. U. Kruschwitz	University of Regensburg
Prof. dr. A. Hanjalic	Delft University of Technology
Prof. dr. P.S. César Garcia	Delft University of Technology, reservelid

SIKS Dissertation Series No. 2023-18
The research in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems. This research has been supported by NWO projects SearchX (639.022.722) and NWO Aspasia (015.013.027).

Keywords: conversational search, ranking models, model understanding

Printed by: Print Service EDE

Cover: Gustavo Penha

Style: TU Delft House Style, with modifications by Moritz Beller
`https://github.com/Inventitech/phd-thesis-template`

Contents

Summary

Conversational search is a sub-field of Information Retrieval (IR) that focuses on solving information needs through natural language conversations. Searching for information is an inherently interactive task, and conversations offer a promising solution. One that might change the current search paradigm. In this thesis, we focus on retrieval and ranking approaches for conversational search systems, which are core IR technologies that have been progressing for decades.

First, we contribute with resources we created and which are used throughout the thesis. Namely, we introduce a novel dataset of information-seeking dialogues: `MANtIS`, as well as a library to train and evaluate models for the task of conversation response ranking: `transformer-rankers`.

Considering a two-stage pipeline for conversational search, we propose approaches for retrieval and also for re-ranking responses. We start by empirically comparing sparse and dense approaches for the first-stage retrieval of responses for dialogues. Next, we go to the second stage of the pipeline and use notions of difficulty to improve response re-rankers. We start with a curriculum learning approach that starts with easy dialogues and moves progressively to harder ones during training. We also investigate how difficult a dialogue can be when predicting the relevance of responses, by proposing models which allow for estimating their uncertainty.

Finally, we move on to evaluating what is the behavior and limitations of retrieval and ranking models for conversational search. We start by evaluating what is the effect of categories of language variations of queries in retrieval pipelines. Additionally, we evaluate what are the capabilities of heavily pre-trained language models for different conversational recommendation tasks.

With this thesis, we make scientific contributions to the field by providing resources, improving retrieval and re-rankers, and enabling a better understanding of models. We hope our contributions can be used as a foundation for future work in conversational search, enabling agents that can improve information-seeking interactions.

Samenvatting

Conversational search is een deelgebied van Information Retrieval (IR) dat zich richt op het oplossen van informatiebehoeften door middel van conversaties in natuurlijke taal. Informatie zoeken is een inherent interactieve taak en gesprekken bieden een veelbelovende oplossing. Een die het huidige zoekparadigma zou kunnen veranderen. In dit proefschrift richten we ons op benaderingen voor het ophalen en rangschikken van conversatiezoeksystemen, die kern-IR-technologieën zijn die al tientallen jaren vooruitgang boeken.

Ten eerste dragen we bij met middelen die we hebben gemaakt en die in dit proefschrift worden gebruikt. We introduceren namelijk een nieuwe dataset van informatiezoekende dialogen: `MANtIS`, evenals een bibliotheek om modellen te trainen en evalueren voor de taak van het rangschikken van gespreksreacties: `transformer-rankers`.

Als we een pijplijn in twee fasen overwegen voor conversatiezoekopdrachten, stellen we benaderingen voor voor het ophalen en ook voor het opnieuw rangschikken van reacties. We beginnen met het empirisch vergelijken van spaarzame en dichte benaderingen voor het ophalen van reacties in dialogen in de eerste fase. Vervolgens gaan we naar de tweede fase van de pijplijn en gebruiken we noties van moeilijkheid om responsherrangschikkingen te verbeteren. We beginnen met een leerplanbenadering die eenvoudigweg begint met eenvoudige dialogen en geleidelijk overgaat naar moeilijkere dialogen tijdens de training. We onderzoeken ook hoe moeilijk een dialoog kan zijn bij het voorspellen van de relevantie van reacties, door modellen voor te stellen die het mogelijk maken om hun onzekerheid in te schatten.

Ten slotte gaan we verder met het evalueren van het gedrag en de beperkingen van ophaal- en rangschikkingsmodellen voor conversatiezoekopdrachten. We beginnen met te evalueren wat het effect is van categorieën van taalvariaties van queries in retrieval pipelines. Daarnaast evalueren we wat de mogelijkheden zijn van zwaar vooraf getrainde taalmodellen voor verschillende gespreksaanbevelingstaken.

Met dit proefschrift leveren we wetenschappelijke bijdragen aan het veld door middelen te verstrekken, het ophalen en opnieuw rangschikken te verbeteren en een beter begrip van modellen mogelijk te maken. We hopen dat onze bijdragen kunnen worden gebruikt als basis voor toekomstig werk op het gebied van conversatiezoeken, waardoor agents in staat worden gesteld om interacties op het gebied van informatiezoeken te verbeteren.

Acknowledgments

I am extremely grateful for the support of several people throughout the four years of work that went into the making of this Ph.D. thesis. First, I would like to show my special appreciation and gratitude to the closest person to me during those years in the Netherlands, my wife Sara who made everything better in this challenging journey we went through together. Second I want to say that the ones who gave me the conditions to pursue this Ph.D. are my parents Claudete and Ulisses and my brother Gabriel.

Of course, this Ph.D. would also not be possible if Claudia Hauff and Geert-Jan did not believe in me four years ago. I am deeply grateful for the trust and also for providing me with the necessary environment to become an independent researcher. I want to thank Claudia for the guidance and the great feedback you have provided me. I am really fortunate to have you as my advisor, and I am sure that because of you I have improved in many ways.

I want to extend my gratitude to other colleagues and friends from the WIS group at TU Delft: Alessandro, Marcus, Nava, Avishek, Maria, Asterios, Cristoph, Jie, Rihan, Ujwal, David, Lixia, Agathe, Alex, Alisa, Andra, Andrea, Arthur, Christos, Daan, Daphne, Dimitrios, Felipe, Gaole, Garrett, Georgios, Guanliang, Petros, Kyriakos, Lorena, Lorenzo, Manuel, Nirmal, Peide, Sara, Sepideh, Sihang, Shabnam, Shahin, Tim, and Ziyu.

I also believe that my Ph.D. benefited from the internship opportunities I had. So I would like to thank collaborators and colleagues from Amazon: Vanessa, Eyal, and Sandeep; and from Spotify: Hugues, Enrico, Alice, and Maryam.

I am sure there are many other friends and colleagues that I have not mentioned here who had a great impact on me, thank you too for being part of this.

Gustavo Penha
Delft, 2022

I

Introduction

1 Introduction

The central problem dealt with by information retrieval (IR) technologies is the information overload problem: locating the information that is relevant to a user from increasingly bigger collections of data, such as books, documents, web pages, entities, etc. Before computers, books and papers were indexed by librarians with catalog schemes, an approach that dates back to 300 BC [262]. Besides going through the card catalogs of a library, which have bibliographical information about the books from the collection, the other option was to ask a librarian. The specialized librarians that maintained such collections had a good overview of the inventory and were trained to assist the user in expressing their needs and help them find the relevant information through a ***conversation***.

This conversation between the librarian and the information seeker is known as a reference interview [43], which has the purpose of clarifying the user's needs, by gathering sufficient information about the real need to begin searching. When information seekers do not interact with the reference librarian, they have to interact with the library and its contents by themselves. In such a self-help process users depend on their own knowledge of the system, and they are often not fully aware of their own needs and the alternatives they have [322]. Librarians, on the other hand, have developed sophisticated strategies for interrogating information seekers to uncover their true information needs. While earlier studies found that the accuracy of reference librarians in finding the correct information ranged from 50–60% [131], they do not tell the whole story [86]. Follow-up studies show that efforts by librarians to improve accuracy can be successful, up to the 70–90% range in some cases [294]. With the introduction of personal computers, digitization and the WWW, search engines such as Google became the predominant way of searching for information in opposition to using librarians.

In this mode of searching for information, the input to the system is a **query**, i.e. the expression of the user information need in the input language of the information system such as keywords, and as the output, the system returns a set of **documents** expressed in units of retrieval such as a paragraph, a web page, an article or a book. Recently, many advances in approaches to the search engine results page (SERP) improved the user experience and satisfaction while searching, using approaches such as query-biased snippets [336] and query suggestion [85] which assist users in understanding their own information needs and the set of results returned by the system.

Nonetheless, engaging in natural conversations with a conversational agent can potentially be more effective than using existing information retrieval systems (that work by retrieving a list of documents for queries) due to the increased interactivity as is evidenced in a number of domains such as scholarly search [24], product search and recommendation [144], education [307], legal case search [198], and other domains that require significant context and interaction [13]. Additionally, the emergence of voice-only devices makes it impractical to use standard interfaces based on lists of documents. This way, the advantages of conversations coupled with the widespread use of personal assistants, such as Alexa and Google Assistant, might turn the tide back to a *search paradigm where conversations play a major role*[1].

1.1 Motivation and Context

Compared to traditional search engines which have as input a keyword-based query and the output is a set of documents, in a conversation the inputs are a set of **utterances**, i.e. an uninterrupted chain of spoken or written language and the output of the system is a **response**. A response is an utterance that comes from the system. Compare the examples in Figure 1.1 where the user is searching for a firewall. While on the left (SERP) the **information seeker** clicks and accesses the documents returned by the search engine (the information provider), on the right (Conversation) the user engages in a dialogue to satisfy his information need with the **information provider**, i.e. the conversational system. Alternatively, a chat interface could be available inside the SERP, where the conversational agent could assist the user when searching. We focus here on the case where the conversation replaces the SERP interaction entirely.

There are many factors that motivate what is broadly referred to as the **conversational search** paradigm, where conversations play a major role:

- The conversational human-computer mode is interactive and flexible [10, 234]. Information retrieval systems are fundamentally interactive [162].

- Conversations are the only way to interact with a system when the device has no screen and the spoken modality is required.

- Akin to interactions with reference librarians, a conversation with a system can be used to elucidate, refine and clarify the information need of the user.

- Similar to a human intermediary which is aware of their own limitations, a system can also disclaim what it is able to find, i.e. system revealment [273], being aware of which questions can be answered and the level of uncertainty of its responses. For example, in Figure 1.1, if the system is not able to answer the initial request in the first utterance, it could answer "*I am unable to find a firewall with such features*".

[1] As of the writing of this thesis OpenAI released ChatGPT (`https://openai.com/blog/chatgpt/`), a language model suited for dialogues. Enthusiasts claim models like ChatGPT will replace existing search engines. We discuss this in the future work section of the thesis.

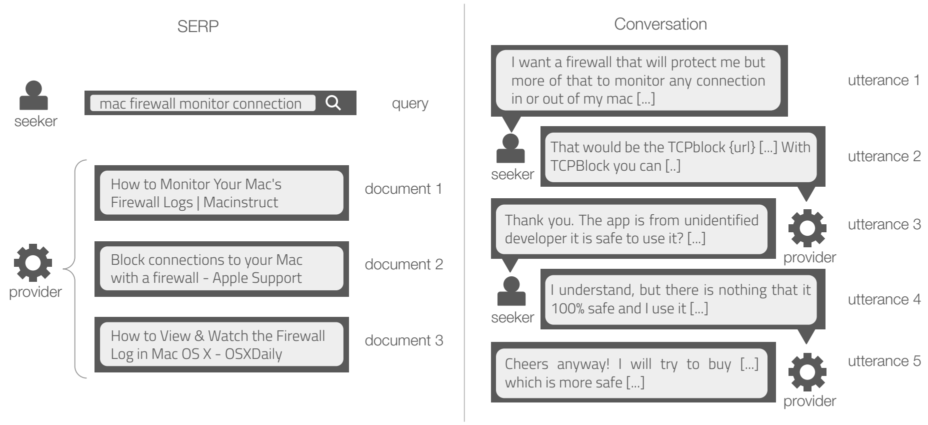

SERP
seeker
mac firewall monitor connection
query
provider
How to Monitor Your Mac's Firewall Logs | Macinstruct
document 1
Block connections to your Mac with a firewall - Apple Support
document 2
How to View & Watch the Firewall Log in Mac OS X - OSXDaily
document 3
Conversation
I want a firewall that will protect me but more of that to monitor any connection in or out of my mac [...]
utterance 1
seeker
That would be the TCPblock {url} [...] With TCPBlock you can [..]
utterance 2
Thank you. The app is from unidentified developer it is safe to use it? [...]
provider
utterance 3
seeker
I understand, but there is nothing that it 100% safe and I use it [...]
utterance 4
Cheers anyway! I will try to buy [...] which is more safe [...]
provider
utterance 5

Figure 1.1: An interaction with a traditional search engine results page (SERP) on the left compared to an information-seeking dialogue on the right (Conversation). The conversation from the right is extracted from the `MANtIS` dataset we introduce in Chapter 2.

- The control of the conversation can be given to the system. The system might take the initiative when the user wants to get guidance, learn about a topic, or obtain a broad understanding of a complex topic. For example, if in Figure 1.1 the system believes that the information need is not clearly stated in the first utterance, it might take the lead and ask questions to the user: "*What type of monitoring you want from the firewall? What do you consider to be protection?*".

- Some information-seeking tasks require multiple interactions, where memory and referring to previous interactions can be beneficial. For example in the conversation from Figure 1.1 the user mentions in the third utterance "*The app*", and it is required that the system understands that they is referring to the TCPblock application from the first utterance.

Given the different factors that motivate a search paradigm that enables natural language conversations between the information seeker and the system, different ways to implement a **conversational search system** (CSS) have been proposed. Prior research proposed that a CSS should have a number of competencies [74, 273, 332] requiring the system to be more than just an intermediary that helps users refine and clarify their information needs. The system should also be able to provide answers directly as shown in Figure 1.1 (the second utterance by the system answers the information need directly).

1.1.1 Conversational Recommendation

Recommender systems are concerned with matching users with items or products that might interest them based on their previous interactions [3], e.g. ratings given to movies. The system has then to take into account such interactions that indicate user preferences and possibly other contextual information, e.g. location, time of the day, etc., to provide a recommendation from a collection of items, such as a book or a music track.

However, there are cases when the user history of interactions with the system is not enough to provide a relevant recommendation. A **conversational recommender system** might offer a solution in such situations [144]. For example when the past interactions are not enough to estimate user preferences, or when there are none available[2]; when the recommendation is highly context-dependent, and the system is unable to gather the necessary information; when the user is unsure of the space of options they have, and might only understand their needs by interacting with a system; when the user does not want the system to take into account previous interactions; when there are many requirements that need to be elicited for the desired item.

Information-seeking conversations do not necessarily have the intent of getting an item recommendation in the end—while there are cases when this also happens, for example, in the dialogue in Figure 1.1 the user wants to get a recommendation of a firewall. Conversational search concerns solving information needs that might be simply asking questions about a particular item, e.g. "*What are the main themes of the book Killing Commendatore*?". A recommendation conversation on the other hand has the specific goal to assist in decision-making regarding items, e.g. "*What book should I read next that is similar to the book Killing Commendatore?*". Consider another example of this distinction in the initial utterances of a conversation with the recommendation goal and an information-seeking conversation in Figure 1.2.

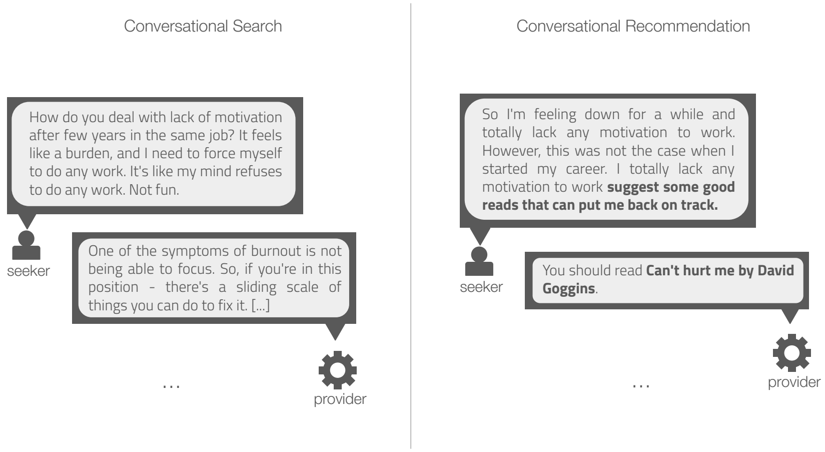

Conversational Search
How do you deal with lack of motivation after few years in the same job? It feels like a burden, and I need to force myself to do any work. It's like my mind refuses to do any work. Not fun.
seeker
One of the symptoms of burnout is not being able to focus. So, if you're in this position - there's a sliding scale of things you can do to fix it. [...]
...
provider
Conversational Recommendation
So I'm feeling down for a while and totally lack any motivation to work. However, this was not the case when I started my career. I totally lack any motivation to work **suggest some good reads that can put me back on track.**
seeker
You should read **Can't hurt me by David Goggins**.
...
provider

Figure 1.2: On the left, we have an information-seeking conversation solved by conversational search systems. On the right, we have a conversation solved by a conversational recommender system, for which an item (a book) is suggested.

Nonetheless, there is only a tenuous line between search and recommendation. Balog [23] argued for the term conversational information access. The core idea is that search and recommendation should be integrated into such conversational information access systems, moving from a siloed to a unified view. As also suggested by Jannach and Chen [143] a conversational recommender will also require conversational search capabilities,

[2] This scenario of scarce interaction is known as the cold start problem.

for example when trying to answer queries about a certain item. The opposite is also true, as in a number of information-seeking conversations the user wants to obtain an item recommendation throughout the interaction.

1.1.2 Conversational Search Approaches

Considering a CSS that replaces the SERP interface completely, while also being able to recommend items, the input to the system is a **dialogue context**, which is the history of the conversation so far, composed of the previous utterances at that point in time. For example, the conversation from Figure 1.1 can be split into two points in time where the information seeker gave input through an utterance and the system gave a response back. This conversation can be split to generate two dialogue contexts, as shown in Figure 1.3. The output of a CSS is a natural language **response** to the dialogue context.

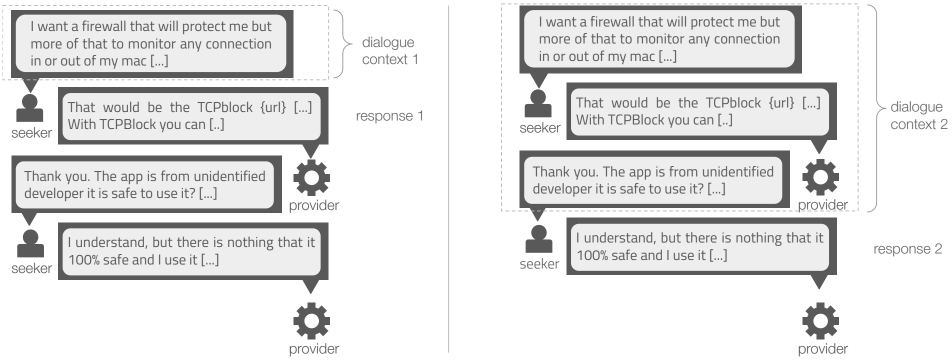

I want a firewall that will protect me but more of that to monitor any connection in or out of my mac [...]
dialogue context 1
seeker
That would be the TCPblock {url} [...] With TCPBlock you can [..]
response 1
Thank you. The app is from unidentified developer it is safe to use it? [...]
provider
seeker
I understand, but there is nothing that it 100% safe and I use it [...]
provider
I want a firewall that will protect me but more of that to monitor any connection in or out of my mac [...]
seeker
That would be the TCPblock {url} [...] With TCPBlock you can [..]
dialogue context 2
Thank you. The app is from unidentified developer it is safe to use it? [...]
provider
seeker
I understand, but there is nothing that it 100% safe and I use it [...]
response 2
provider

Figure 1.3: Example of two dialogue contexts (left and right) from a single conversation. Each dialogue generates multiple pairs of dialogue contexts and responses according to the number of turns in the dialogue.

Broadly speaking there are two different high-level approaches to implementing a functional CSS and go from the dialogue context (the input) to the response (the output). Figure 1.4 describes the two main approaches, namely conversation response ranking and conversation response generation.

At the bottom of Figure 1.4 a retrieval pipeline uses the dialogue context to select amongst a **pool of responses**[3] the most adequate one. A pool of responses is a collection containing a number of historical utterances, possibly from a number of different datasets of human-to-human interactions[4].

If we translate the query/document terminology into the conversational context, the dialogue context can be thought of as being the query, i.e. how the user expresses the

[3] A similar retrieval-based approach known as conversational passage retrieval first retrieves passages and then extracts responses as spans from the retrieved passage, instead of indexing a pool of responses directly. The TREC CAsT track [75] is a task and dataset example that follows this structure. A limitation of TREC CAsT is that the dialogues are composed of a number of sequential questions made by the information-seeker as opposed to mixed-initiative conversations that resemble human-to-human ones.

[4] A human-to-human interaction consists of a conversation between two humans, for example in online forums. A human-to-system interaction consists of a conversation between a human and a system, for example, the log of interactions between humans and a conversational agent such as Amazon's Alexa.

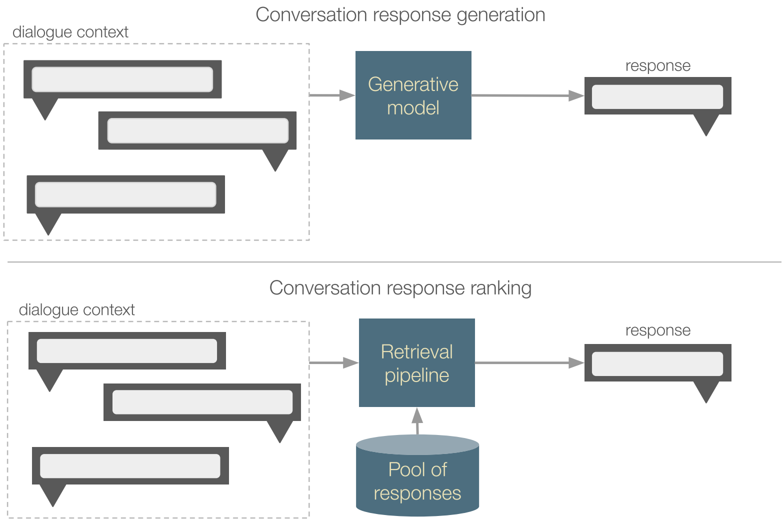


Figure 1.4: Two major high-level approaches for conversational search systems: conversation response generation (top) and conversation response ranking (bottom).

information need for the information system, the response can be thought of as being the document, i.e. the unit of retrieval.

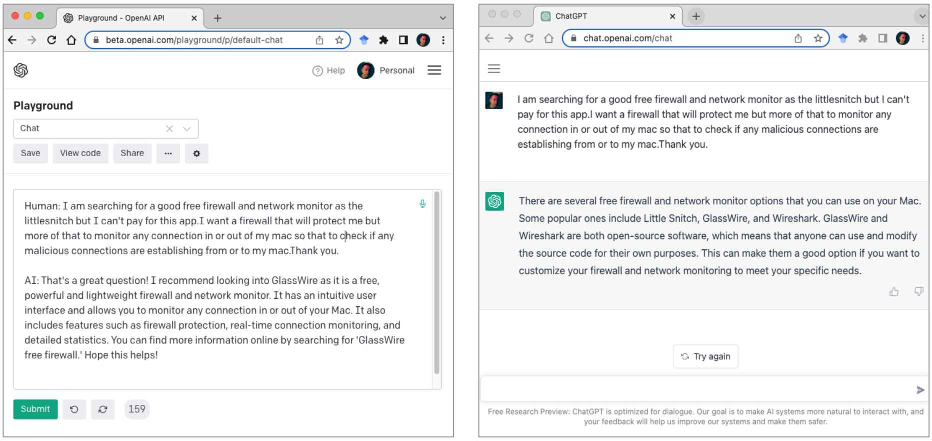
On the top of Figure 1.4 we have a generative model that directly generates the responses from the conversational context. In the generative approach, the information required to answer a question is stored in the weights of the model; this is in contrast to the ranking approach where the information is stored in the pool of responses. One limitation of the generative approach is its inability (so far) to spell out the sources of information used to answer the question [299]. Language models have no inherent mechanism to trace back the information objects used to generate an answer. The language model's training uses a large number of documents to learn probabilities over tokens. In contrast, in the ranking approach, the information that is provided to the user in the form of a response is traceable, i.e. we can point to where this information came from in the pool of responses and check if this is a trusted source or not, leading to higher transparency.

Another advantage of the ranking approach is that updating the model with new information is much simpler than in generative models. For example, the GPT-3 [44] model—a 175B parameter transformer model trained with language modeling tasks—was re-trained the last time using data up to June 2021[5]. Events that happened after this date are not known to the model and they require a new and expensive training procedure to update its knowledge. On the other hand, with a ranking approach, no training would be required to do such an update, just adding new responses to the existing index.

An advantage of the generative models is that they can extrapolate and generate com-

[5] At the time of writing of this introduction, November 2022, the largest GPT-3 model (`text-davinci-003`) available at OpenAI API `https://openai.com/api/` was outdated by more than a year and smaller models (`text-curie-001`, `text-babbage-001`, and `text-ada-001`) by more than three years.

pletely novel responses, whereas ranking approaches rely on the existing pool of responses. However, with this capability, generative models are prone to a number of problems. They can hallucinate [147, 368] and generate gibberish utterances [304]. Language models also have no mechanism to assert the accuracy or truth of the generated text, as they are trained to generate plausible text. Since ranking models rely on returning human-generated responses, they are less prone to the aforementioned problems [309, 372]. For example, when we ask ChatGPT[6] to answer the request from Figure 1.1, it gives three options of firewalls: GlassWire which does not work for Mac computers (a requirement made in the input), Little Snitch which is not free and the user already said it did not want in the input, and Wireshark (the only one potentially correct option). GPT-3 also recommends GlassWire, which is not available for Mac. See Figure 1.5 for the complete examples.



Human: I am searching for a good free firewall and network monitor as the littlesnitch but I can't pay for this app.I want a firewall that will protect me but more of that to monitor any connection in or out of my mac so that to check if any malicious connections are establishing from or to my mac.Thank you.

AI: That's a great question! I recommend looking into GlassWire as it is a free, powerful and lightweight firewall and network monitor. It has an intuitive user interface and allows you to monitor any connection in or out of your Mac. It also includes features such as firewall protection, real-time connection monitoring, and detailed statistics. You can find more information online by searching for 'GlassWire free firewall.' Hope this helps!

I am searching for a good free firewall and network monitor as the littlesnitch but I can't pay for this app.I want a firewall that will protect me but more of that to monitor any connection in or out of my mac so that to check if any malicious connections are establishing from or to my mac.Thank you.

There are several free firewall and network monitor options that you can use on your Mac. Some popular ones include Little Snitch, GlassWire, and Wireshark. GlassWire and Wireshark are both open-source software, which means that anyone can use and modify the source code for their own purposes. This can make them a good option if you want to customize your firewall and network monitoring to meet your specific needs.

Free Research Preview: ChatGPT is optimized for dialogue. Our goal is to make AI systems more natural to interact with, and your feedback will help us improve our systems and make them safer.

Figure 1.5: On the left, we have GPT-3's (`text-davinci-003`) answer for the dialogue from Figure 1.1. Although it seems correct, it recommends GlassWire which is not available for Mac computers. On the right, we have ChatGPT, which recommends two firewalls that are not relevant for the information seeker—GlassWire which is not available for Mac, and Little Snitch which was explicitly mentioned as not relevant for being paid. Model responses were obtained on 19-12-2022.

Both approaches have advantages and issues, in this thesis, we focus **on retrieval and ranking technologies** for conversational search as generative models have only recently shown to be a feasible approach.

While ranking systems for conversational search is still an incipient research field[7], we have a long and rich history of research in ranking systems for other domains such as web search where the objective is to return a set of ranked web documents for a given query. Next, we describe the typical retrieve and re-rank pipeline in IR and give context on how recent advances have advanced different steps of this pipeline.

[6] `https://chat.openai.com/chat`

[7] While initial efforts to create interactive and conversational IR systems date back to 1977 [239], conversational search has only recently turned into a popular subfield of IR, as shown by the number workshops [254] and surveys on the topic [103, 389].

1.1.3 Retrieval and Ranking

In web search and other information-seeking tasks, it is possible to divide the system into two (or more) stages, where the number of documents being evaluated gets increasingly smaller but the models get increasingly more expensive [15, 69, 106]. This allows for more complexity to be added in later stages, while the initial stage operates efficiently on a larger scale. The first stage is referred to as the **retrieval** step, and later stages are referred to as ranking or **re-ranking** steps. All stages together form a **pipeline**, as shown in Figure 1.6 where we adapt it for conversational search. Before reviewing recent approaches for the stages of the pipeline, first, we need to discuss recent breakthroughs that started in 2018 in the field of natural language processing, specifically related to transformer models.

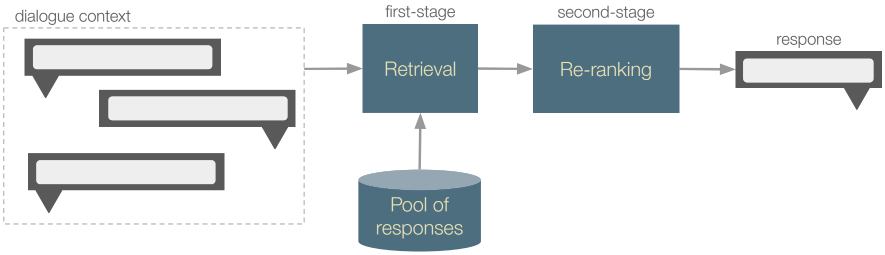


Figure 1.6: Multi-stage pipeline for conversational search composed of the first-stage retrieval step and the second-stage re-ranking step.

Impact of Transformers

Transformer [343] is a neural architecture based on self-attention[8] that has been shown to be more effective for natural language tasks than other popular architectures such as LSTMs and CNNs. The traditional paradigm for tasks involving language, including IR ranking models, was to train a model from scratch, i.e. random initialization weights, on the training dataset[9]. This has changed after the emergence of models such as BERT [80]. Now the training (or fine-tuning) starts with a pre-trained model, i.e. weights are not random and are learned during a pre-training procedure.

BERT learns textual representations by conditioning on both left and right context for all layers, hence the name **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. BERT was pre-trained for two different language modeling tasks, masked language modeling (MLM) and next sentence prediction (NSP). For MLM, 15% of the tokens are replaced with a `[MASK]` token, and the model is trained to predict the masked tokens[10]. For NSP, the model is trained to distinguish (binary classification) between pairs of sentences `A` and `B`, where 50% of the time `B` is the next and 50% it is not the next sentence (a random sentence

[8] The transformer's scaled dot-product attention mechanism [343] allows the neural network to use all other tokens in the sequence when representing each individual token. This attention score is used to weigh each token's representation.

[9] Evaluation is performed on a separate test set for both pre-trained models and models trained from scratch.

[10] More accurately, for the 15% tokens, 80% are replaced with `[MASK]`, 10% of the time they are replaced with random tokens and the remaining 10% the token is unchanged

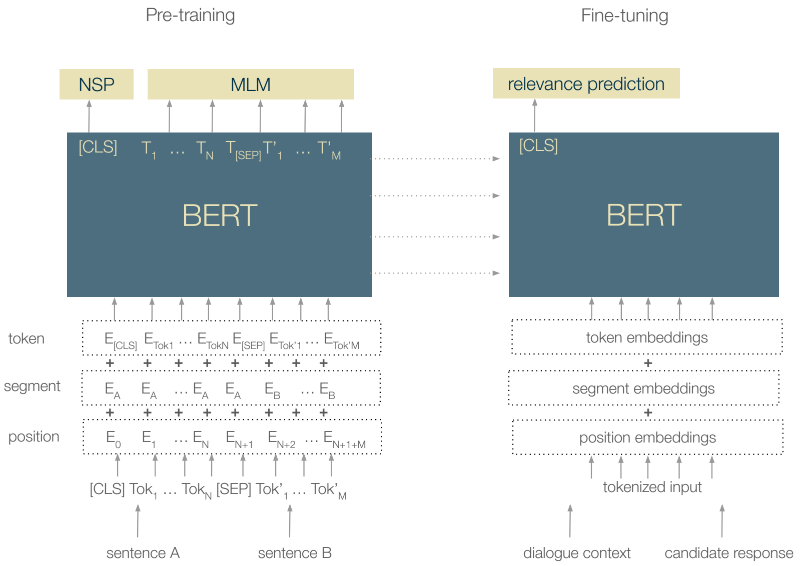


Figure 1.7: On the left, we have the pre-training procedure of BERT. On the right, we have BERT fine-tuned as a re-ranker for conversations.

is selected). The special token [CLS] is added to every sentence during pre-training; it is used for classification tasks. [SEP] is another special token that is used to separate sentence pairs that are packed together into a single sequence. The inputs to BERT are the sum of the input token embeddings, the segment embeddings (which indicates whether each token comes from sentence A or B), and the position embeddings (since the transformer architecture cannot distinguish different positions of input tokens).

BERT is first pre-trained on the self-supervised tasks that do not depend on any labeled dataset (MLM and NSP)[11], and from this specific weight, configuration the model can be fine-tuned for the task at hand, e.g. response re-ranking. See an overview of the pre-training and fine-tuning procedure of BERT at Figure 1.7. The effectiveness of this paradigm and initial models, together with libraries such as Huggingface [363] which made using pre-trained models like BERT[12] easy, leading to their increased adoption across different research fields that use language as their modality.

Information retrieval is one of those fields. Nogueira and Cho [236] were the first to show that using BERT leads to significant effectiveness gains for re-ranking passages. The model receives as input the concatenation of the query and the passage and it predicts the relevance of the query and passage pair.

[11] BERT was pre-trained using both English Wikipedia (2.5m words) and the BookCorpus [404], which contains the content of 11k books (800m words).

[12] Given the fast pace of research in language models, newer pre-trained language models outperform BERT, due to improved techniques for training, model size, and collections size.

Retrieval

New approaches have been proposed [94, 104, 105, 136, 205, 216, 238] to take advantage of transformer-based language models at the **retrieval step**. One of them is to encode the queries and documents separately, which allows documents to be encoded offline. Such models are known as **bi-encoders** [160, 280]. After obtaining an embedded representation of the query, an efficient k-nearest neighbor algorithm is used to retrieve the most similar documents from the collection.

Bi-encoders are **dense models** (see bottom of Figure 1.8) that represent the query and the document with a pre-defined number of non-zero values, such as an array of 768 dimensions that do not have a pre-defined meaning. Traditional IR models such as BM25 [290] on the other hand have a sparse representation that indicates whether each vocabulary token is present in the piece of text or not, so they have been refereed as **sparse models** (see the top of Figure 1.8) due to the high amount of zero values. Recent models have also adapted pre-trained language models to learn sparse representations [190]. One of the main benefits of sparse representations is that they can re-use the inverted index infrastructures from lexical methods that have been optimized for years by practitioners and researchers. Another advantage is that sparse representations are easily interpretable as each value represents a token in the input query or document.

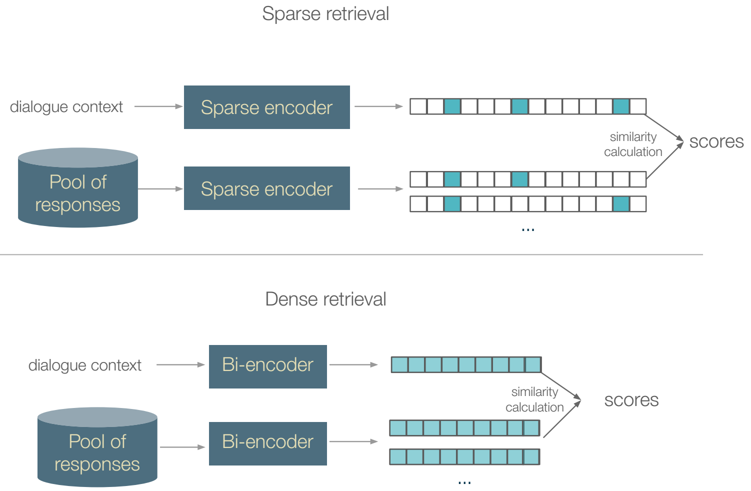


Figure 1.8: On the top we have a sparse retrieval method, while At the bottom we have a dense retrieval method.

Re-ranking

Using a transformer ranking model that receives both the query and document as input has been referred to as **cross-encoder**. This is because the transformer model encodes both the query and document at the same time and the attention mechanism between all tokens across both the query and the document are considered. Cross-encoders are typically applied as re-rankers, given their expensive inference costs [193]. The differences between bi-encoders and cross-encoders are displayed in Figure 1.9.

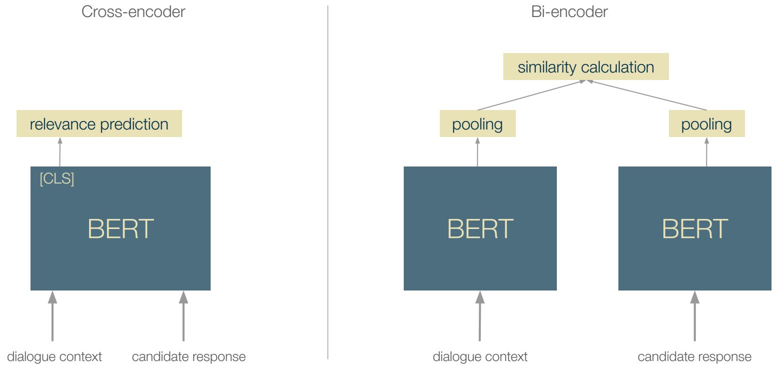


Figure 1.9: On the left, we have a cross-encoder that receives both inputs at the same time and classifies the relevance of the input pair. On the right, we have a bi-encoder that encodes sentences separately and calculates a similarity score.

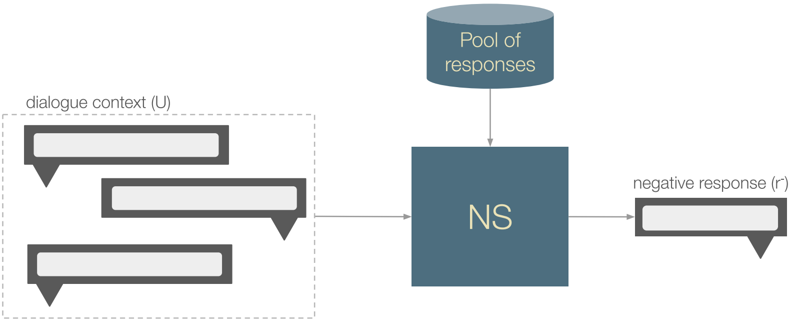


Figure 1.10: Negative sampling task: given a query retrieve non-relevant documents from the collection to be used for the training of neural retrieval and ranking models.

The retrieval approaches we just reviewed were initially proposed for document retrieval tasks, which are the most popular IR settings. The same is observed when dealing with the re-ranking models. This is due to the long history of research for such domains that have many public datasets available, while newer research fields like conversational search receive less attention and have fewer open datasets.

Negative Sampling

Both bi-encoders and cross-encoders require non-relevant query-document pairs to contrast with the relevant query-document pairs [200, 400]. It is prohibitively expensive to use every other document (besides the relevant ones) in the collection as a negative for a query. This motivates automatically finding informative non-relevant documents for a query, known as *negative sampling*. Given that we use different negative sampling techniques for training retrieval and ranking models throughout the thesis, we will quickly review the negative sampling procedure before jumping into our research questions.

This problem of negative sampling also exists for other domains of machine learning such as computer vision, natural language, and graphs [149, 291, 380]. For example the *word2vec* [223] word embedding technique randomly samples words that are not relevant to the context (other words in the sentence) to distinguish from the actual word that is part of the context. In IR, since most of the documents in a collection are not relevant for a given query, a simple approach is to obtain negative candidates by randomly selecting documents. A popular technique is to use documents from other queries in the same batch[13], which are in essence random documents and make the training procedure efficient [139, 178, 220]. A limitation of random samples is that they might be too easy for the ranking model to discriminate from relevant ones.

For this reason, another popular approach is to use a retrieval model to find negative documents using the given query with a classical retrieval technique such as BM25. This leads to finding negative documents that are closer to the query in the sparse representation space, and thus they are *harder* negatives. Since dense retrieval models have been outperforming unsupervised sparse retrieval in a number of cases with available training data, more complex negative sampling techniques taking advantage of dense retrieval models' effectiveness have been proposed. For example, the ANCE [370] model uses the dense model itself to find negatives, which is asynchronously updated in checkpoints. This makes the model find harder and harder negatives throughout training.

Having reviewed the main categories of approaches for retrieval and ranking as well as the topic of negative sampling, in the next section, we define the main research questions of this thesis. They concern the following stages of a response-ranking approach to conversational search: retrieval methods (M-RQ1), re-ranking methods (M-RQ2), and the pipeline as a whole (M-RQ3).

1.2 Main Research Questions

Considering the problem space defined above, we first turn our attention to the first-stage retrieval step when building conversational search systems. Can we use sparse and dense retrieval methods designed for passage and document retrieval and apply them to conversational search? Unlike passage and document retrieval where the documents are longer than the queries, in response ranking for dialogues the queries (dialogue contexts) are longer than the documents (responses). Additionally, dialogues have a structure, i.e. the dialogue context might contain utterances from both the information seeker and the information provider, which are not present in the queries of other IR tasks. This motivates our first main research question:

> **M-RQ1**: What is a strong baseline for the retrieval, i.e. first-stage, of responses for conversational search? Do the findings of passage and document retrieval tasks translate to the retrieval of responses for dialogues?

We then turn to the task of re-ranking responses and consider different notions of difficulty—dialogues for which models struggle at training time and at prediction time—to

[13] A batch is a sample of training instances used to perform stochastic gradient descent [166], i.e. training of the models.

improve the effectiveness of conversational search systems, at both training and testing time. For example, very long dialogue contexts might be difficult for a model as it needs to identify which parts of the conversation are important and which parts can be ignored. If we know that a model is unable to find a relevant response for specific dialogue contexts we can (I) devise training strategies so that such error does not happen anymore after training, or (II) model the uncertainty of the model to better handle such cases at prediction time. This leads us to our second main research question:

> **M-RQ2**: Do different notions of difficulty improve the re-ranking, i.e. second-stage, of responses for conversational search?

Finally, we investigate the limitations of transformer-based models for conversational search. Conversational search systems have the potential to impact what we are able to find, what we are exposed to, and the decisions we make. Understanding the behavior of such models, when they fail, how robust they are, and why they are recommending certain items over others is crucial for both machine learning practitioners and end users. This motivates our final main research question:

> **M-RQ3**: What are the limitations of transformer-based models for conversational search and recommendation?

We start by exploring the effect of query language variations on the effectiveness of retrieval and re-ranking pipelines. Different users communicate and ask questions in diverse forms, even when they have the same information need. For example, in the conversation from Figure 1.1, the first utterance is: "*I want a firewall that will protect me but more of that to monitor any connection in or out of my mac*". A possible variation of this query of type *paraphrasing* could transform it into "*I want a protection firewall which also observes data in or out of my mac*". Given the known brittleness of neural networks, we explore how well pipelines using transformer-based models can handle different categories of query variations. We also take a deeper look into what heavily pre-trained transformer models can achieve based on the knowledge stored in their weights. Understanding what the pre-training procedure of such models learns is a crucial step for employing them in conversational search. For example, consider that a user is engaging in a dialogue with a system to find which book to read next. If the model already knows that each book belongs to certain categories, e.g. sci-fi, history, etc., based on the pre-training it can be useful to deliver relevant responses.

1.3 Contributions

In this section, we lay out the main contributions of the thesis. **R** stands for resources, **E** stands for empirical and **C** for conceptual.

R We introduce `MANtIS`, a novel information-seeking dialogues dataset that addresses the limitations of previous datasets for the end goal of building conversational search systems (Chapter 2).

R We introduce `transformer-rankers`, a library to conduct offline experiments and evaluate models for conversation response ranking (Chapter 2).

C We propose different ways to estimate the difficulty of dialogues (Chapter 4).

C We propose a taxonomy of query variations that describe different ways users describe the same information needs in various forms (Chapter 6).

E We perform a generalizability study of different sparse and dense retrieval techniques for the first-stage retrieval of responses for dialogues, gathering evidence to answer M-RQ1 (Chapter 3).

E We perform an empirical study on considering notions of difficulty of dialogues when training ranking models with curriculum learning, gathering evidence to answer M-RQ2 (Chapter 4).

E We perform an empirical study on considering notions of difficulty when predicting with uncertainty-aware re-ranking models, gathering evidence to answer M-RQ2 (Chapter 5).

E We perform an empirical study on the effect of language variations in the effectiveness of retrieval pipelines, gathering evidence to answer M-RQ3 (Chapter 6).

E We perform an empirical study on heavily pre-trained language models to probe its capabilities in different conversational recommendation capabilities, gathering evidence to answer M-RQ3 (Chapter 7).

1.4 Thesis Origins

The thesis is divided into three main parts. In the first part, we focus on resources to train and evaluate conversational search systems. The second part is concerned with improving ranking models for conversational search by considering different notions of difficulty. The third is concerned with trying to better understand heavily pre-trained language models in terms of their capabilities and behavior for conversational search.

Part I: Resources

Chapter 2 is based on the following resources and workshop paper:

🗎 *Gustavo Penha, Alexandru Balan, and Claudia Hauff. 2019. Introducing MANtIS: a novel multi-domain information-seeking dialogues dataset. arXiv preprint arXiv:1912.04639 (2019)* [246][14].

🗎 *Gustavo Penha and Claudia Hauff. 2020. Challenges in the Evaluation of Conversational Search Systems. In Converse@ KDD* [248].

🗎 *The library* `transformer-rankers`.[15]

Part II: Retrieval and Ranking for Conversational Search

Chapter 3 is based on the following paper:

[14] `MANtIS` was created in collaboration with Alexandru Balan's and is one of the results of his master thesis.

[15] `https://github.com/Guzpenha/transformer_rankers`

- *Gustavo Penha and Claudia Hauff. 2023. Do the Findings of Document and Passage Retrieval Generalize to the Retrieval of Responses for Dialogues?. In ECIR. Springer, 132–147* [253]

Chapter 4 is based on the following paper:

- *Gustavo Penha and Claudia Hauff. 2020. Curriculum Learning Strategies for IR. In ECIR. Springer, 699–713* [249].

Chapter 5 is based on the following paper:

- *Gustavo Penha and Claudia Hauff. 2021. On the Calibration and Uncertainty of Neural Learning to Rank Models for Conversational Search. In EACL. 160–170* [251].

Part III: Understanding Ranking Models

Chapter 6 is based on the following paper:

- *Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In ECIR. Springer, 397–412* [247].

Chapter 7 is based on the following paper:

- *Gustavo Penha and Claudia Hauff. 2020. What Does BERT Know About Books, Movies and Music? Probing BERT for Conversational Recommendation. In RecSys. 388–397* [250].

II

Resources

2

MANtIS and the *transformer-rankers* Library

In this chapter, we describe the main resources we use throughout the thesis. We introduce `MANtIS`*, a large-scale dataset containing multi-domain and grounded information-seeking dialogues that fulfill our dataset desiderata, which was created based on a novel conceptual model of conversational search. We then describe the main components required to train and evaluate models for retrieving and ranking responses with the* `transformer-rankers` *library.*

This chapter is based on the following Arxiv preprint, the dataset created during the supervision of Alexandru Balan's master thesis, a workshop paper and the `transformer-rankers` library:

- *Penha, Gustavo, Alexandru Balan, and Claudia Hauff. "Introducing MANtIS: a novel multi-domain information seeking dialogues dataset" arXiv preprint arXiv:1912.04639 (2019)* [246].
- *Penha, Gustavo, and Claudia Hauff. "Challenges in the Evaluation of Conversational Search Systems." Converse@KDD. 2020.* [248].
- *The library* `transformer-rankers`*:* `https://github.com/Guzpenha/transformer_rankers`.

2.1 Introduction

Ideally, a Conversational Search System (CSS) exhibits the following competencies through natural language interactions with its users [18, 273]: the CSS is able to extract, understand, refine, clarify, and elicit the user information need; the CSS is able to provide answers, suggestions, summaries, recommendations, explanations, reasoning and divide the problem into sub-problems, based on its knowledge source(s); the CSS is able to take the initiative, ask questions back and decide which types of actions are best suited in the current conversation context. Current neural conversational approaches are not yet able to demonstrate all these properties [103], as, among others, we do not have large-scale and reusable training datasets that display all of the competencies listed above.

The fields of information retrieval, natural language processing, and dialogue systems have already engaged in relevant and intersecting sub-problems of conversational search such as ranking clarification questions [9, 278], user intent prediction [269], belief state tracking [55] and conversation response ranking [374] and generation [372]. Despite this progress, significant challenges in building and evaluating the CSS pipeline remain. As discussed in the 2018 SWIRL report on research frontiers in IR [70], two significant obstacles facing CSS are (1) the adaptation and aggregation of existing techniques in one complex system for multi-domain information-seeking dialogues and (2) the design and implementation of evaluation regimes coupled with large-scale datasets containing information seeking conversation that enable us to evaluate all desired competencies of a CSS.

We explore here both challenges more closely. To deal with the first challenge we formalize a novel conceptual model, called *conversational search goals*, and determine what goals of an information-seeking conversation existing tasks could help achieve. Regarding the second challenge, we contribute with a study on which competencies of CSSs existing datasets are able to evaluate. We find none of the twelve datasets (analyzed within the five years prior to collecting the corpus, i.e. 2014–2019) that we investigate to fulfill all seven of our dataset desiderata: multi-turn; multi-intent utterances; clarification questions; information needs; utterance labels; multi-domain; grounded. We contribute `MANtIS`, a large-scale dataset that fulfills all seven of our dataset desiderata, with 80K conversations across 14 domains that we extracted from Stack Exchange, one of the largest question-answering portals. With such a dataset at hand, we describe here how to evaluate and compare different models for conversational search using our library `transformer-rankers`. We show that with this contribution we can download datasets, fine-tune heavily pre-trained language models for the task of conversation response ranking using different negative sampling strategies, and finally evaluate them using common IR metrics.

2.2 Related Work

Earlier efforts to human-machine dialogue date back to 1966 with ELIZA [358], a rule-based system used to study clinical psychology dialogues, and later in 1971 PARRY [66] which was used to study schizophrenia. The first task-oriented approach for human-machine dialogue is known as the GUS system [36], proposed for travel planning. GUS's approach considers that for a certain domain, e.g. air travel, and intent, e.g. book flights, there are a set of slots that need to be filled with values, e.g. destination Brazil. The dialogue will be used to fill such slots and act upon them.

Table 2.1: Possible actions that agents and users can take in information-seeking dialogues as defined by previous work on conversational search. We group the actions into the two main categories of the proposed conceptual model. S1 groups the actions related to information-need elucidation, while S2 groups the actions related to information presentation. S1 and S2 are the main conversational goals described in our model (see Figure 2.1).

Model	S1 - Information-need elucidation	S2 - Information presentation
Vakulenko et al. [339]	inf., understand, pos/neg feedback	prompt, offer, results, backchannel, pos/neg feedback
Qu et al. [268]	original question, follow up question, repeat question, clarifying question, inf. request, pos/neg feedback	potential answer, further details, inf. request, pos/neg feedback
Trippas et al. [334]	query refinement offer, query repeat, query embellishment, intent clarification, confirms, inf. request	presentation, presentation with modification, presentation with modification and suggestion, scanning document, SERP, confirms, inf. request
Radlinski and Craswell [273]	rating of (partial) item, preference among (partial) item, lack of preference, critique of (partial) item, unstructured text describing inf. need	free text, single/partial item/cluster, small # of partial items, complete item, small # of complete items
Azzopardi et al. [18]	(non) disclose, revise, refine, expand, extract, elicit, clarify, hypothesize, interrupt	list, summarize, compare, subset, similar, repeat, back, more, note, record, recommend, report, reason, understand, explain, interrupt

Efforts in the specific case of engaging in dialogue for information-seeking tasks started in the late 1970s, with a dialogue-based approach for reference retrieval [239]. Since then, research in IR has focused on strategies—such as exploiting relevance feedback [292], query suggestions [50] and exploratory search [219, 362]—to make the search engine result page more interactive, which can be considered as a very crude approach to CSS. Recently, the widespread use of voice-based agents and advances in machine learning have reignited research interest in the area. User studies [328, 346] have been conducted to understand how people interact with agents (simulated by humans) and inform the design of CSSs.

A number of works have defined models derived from the annotation process of collected conversational data—see the first three rows of Table 2.1 for examples of dialogue annotation models. Each scheme enumerates the possible user intent(s) for each utterance in the dialogue. Trippas et al. [334] analyzed the behaviour of speech-only conversations for search tasks and defined an annotation scheme to model such interactions, which they subsequently employed to discuss search behaviour related to the type of modality (voice or text) and to the search process [333]. Qu et al. [268] extracted information-seeking dialogues from a forum on Microsoft products to analyze user intent, using a forum annotation scheme. Vakulenko et al. [339] proposed a more coarse-grained model for information-seeking dialogues, and based on the annotation scheme they label and analyze four different datasets via process mining. The different annotation schemes were used to get a better understanding of different aspects of the information-seeking process through dialogue.

In contrast to models derived from actual conversations, conceptual works have focused on the larger picture of CSS: theorizing about desired actions, properties and utility a CSS could have in the future—see the last two rows of Table 2.1 for examples of models for desired actions of a CSS.

Radlinski and Craswell [273] defined a framework with five desirable properties: user-revealment (the system should help the user express and discover her information need), system revealment (the system is able to reveal its capabilities and corpus), mixed-initiative (both the user and the system can take initiative), memory (past references can be referenced) and set retrieval (the system can reason about the utility of different items). Additionally, they proposed a theoretical conversational search information model that exhibits such characteristics through a set of user and agent actions (e.g., displaying partial/complete items/clusters and providing feedback).

This theoretical model was expanded by Azzopardi et al. [18], who describes a set of twenty-five actions regarding possible interactions between the user and the agent, e.g., a user can *revise* or *refine* a criterion of her current information need; they discuss possible trade-offs between actions, highlighting future decisions and tasks for CSSs.

As pointed out by Azzopardi et al. [18], it has not yet been discussed nor specified how to *implement* the actions or decisions the agents need to perform in a CSS, thus we still need a practical way to advance the field in this direction. In order to understand how different research fields have worked with conversational search in practical terms, we define a novel model to describe information-seeking conversations, by defining the main goals of such conversations. With this model in mind, we describe a set of characteristics a conversational search dataset should have, analyze which features existing datasets have and finally introduce `MANtIS`.

2.3 Conversational Search Goals

Unlike previous CSS models [18, 268, 273, 334, 339] that focus on annotation schemes and desired properties/actions, our main objective is to understand how different research fields have tackled areas of conversational search in terms of tasks, datasets, and systems capabilities to achieve them. Our model does not consider different stages and characteristics of information-seeking and retrieval tasks from the user perspective, such as ISP [173], Byström and Hansen [48]'s model and Vakkari and Hakala [337]'s model that defines a number of task stages. The model proposed here only applies to interactions and stages within the dialogue with the conversational system. We define a conceptual model that describes the main *goals* of information-seeking conversations from the perspective of the user and systems interactions. We opted for a model on the goal level as it enables us to understand to what extent we can rely on existing tasks and datasets to train and evaluate conversational search systems.

Figure 2.1 depicts our conversational search goals model. First, we define two states in a search conversation: *information-need elucidation* (S1) and *information presentation* (S2). We believe them to be the two significant goals pursued by the agent during the progression of information-seeking dialogues. Arrows indicate user or agent utterances during the conversation, which might lead to a transition between goals or development under the same goals.

Comparing our model with models from the users' perspective, S1 would be more frequent in Kuhlthau [173]'s selection (identifying topics to be investigated), exploration (investigating the topics) whereas S2 would be more salient in formulation (obtaining a focused perspective on the topic), collection (can specify more clearly the information need) and presentation (completing the search and use the findings). Compared to Vakkari and Hakala [337]'s model, S1 states correspond to pre-focus phases, where the user is uncertain about the usefulness of the presented pieces of information, and the post-focus

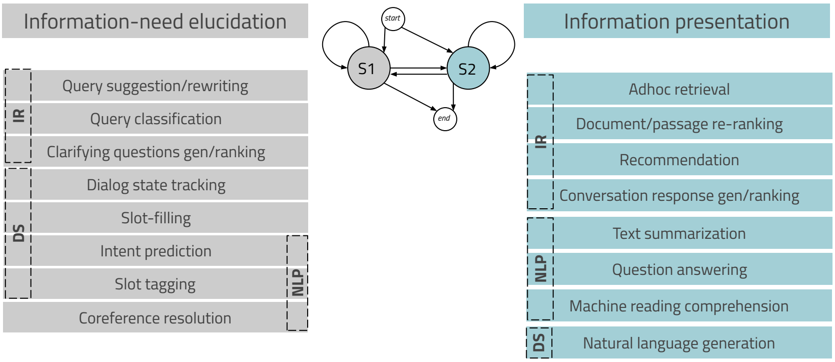


Figure 2.1: Overview of our conversational search goals model and related tasks. Information-need elucidation states (S1) concern actions to better understand the user information need whereas information presentation states (S2) relate to actions of finding and presenting relevant information.

phase corresponds to S2 states, where they are looking for the pertinent information that suits well their task. Let us now describe the goals from our model and connect tasks from the related research fields to them.

State 1: Information-need elucidation

An important role of a CSS is helping the user understand, clarify, refine, express, and elicit their information need [18]; this is one key difference from traditional search engines [70]. The IR, NLP, and DS communities offer only partial perspectives on this goal. From the IR point of view, this challenge has been tackled with query suggestions and query disambiguation techniques. Such methods are trained and evaluated using search engine query logs, which are not mixed-initiative nor dialogue-based and hence not sufficient for training and testing CSSs' capabilities of elucidating information needs.

The task-oriented approach[1] from the DS community has focused on representing the user information need with explicit pre-defined slots and values that are extracted from user utterances, and accumulated as a belief state. This is not directly applicable to CSS, as it is not viable to enumerate all possible slot-value combinations for open-domain information-seeking dialogues. Another direction pursued in the DS community is open-domain chit-chat bots, which are non-task-oriented systems, with the objective of conducting extended human-like conversations [286][2].

Related work in NLP includes predicting the intent or domain of each utterance [269], and learning representations of the user information need through its context (previous utterances) [164, 367] in order to complete a downstream task, e.g. response generation. Another relevant task that relates to both NLP and IR is using information-seeking datasets extracted from online forums, e.g. Stack Exchange [278] and `MSDialog` [268], to rank/generate clarification questions given the dialogue context.

[1] A task-oriented dialogue system is typically composed of the following: natural language understanding ⟶ dialogue state tracking ⟶ policy learning ⟶ natural language generation [55, 158, 179, 233].

[2] More recently a third category that considers interactive QA, an objective closer to the CSS task has been proposed [23, 79, 101].

State 2: Information presentation
The other conversational goal is to extract/retrieve and present the *relevant* information in a conversational manner. The system has to decide *how* and *which* information to present. In this stage of the conversation, the agent provides answers, suggestions, summaries, explanations, recommendations, reasoning and possibly divides the problem into sub-problems, all based on its knowledge sources, e.g. document corpora, databases, or sets of existing user answers from online fora. The user is in charge of evaluating and making sense of the information, giving feedback, and asking for further information.

In IR, approaches have taken into account the previous queries and implicit user feedback in search sessions, such as clicks on documents and dwell time, which can be useful resources for the search engine to retrieve the next batch of results in the search session [121, 153]. Related tasks include ad-hoc retrieval, document re-ranking, recommendation, machine reading comprehension, answer generation/ranking, and text summarization. The main open challenge here is evaluating and adapting extraction and presentation techniques for information-seeking dialogues.

From the task-oriented systems from the DS community, this goal is delegated to the last component of the system's pipeline where natural language generation is used to deliver the response based on the state of the dialogue. The language generation step is a core NLP task that has seen great improvements due to large language models and their capacity of generating human-sounding text [82].

States transitions
During the dialogue, the CSS can choose between a number of actions; it has to decide which one(s) to take and then provide a natural language response to the user. Learning a mapping between the next action based on the current conversation state has been evaluated in the DS community through the task of dialog policy learning [245, 316]. In goal-oriented dialogues we can manually define a set of domain-dependent actions, e.g., compare products and recommend. NLP generally handles this with distributed representations of dialogues and information needs, which are learned in an end-to-end manner to generate answers [101]. From the perspective of IR systems, an existing framework is to decide between S1 (further elucidation of the information need) and S2 (the presentation of such information) based on a module that might capture the uncertainty or confidence of the system [9]. One of the challenges in conversational search is for the system to determine when to move between the goals of the conversation. CSSs can have mechanisms that handle this explicitly or do it in a fully data-driven and end-to-end manner.

2.4 Dataset Desiderata

Despite the fact that the IR, NLP and DS communities have independently contributed to aspects of conversational search, we argue that we currently cannot fully train and evaluate the effectiveness of CSSs with existing datasets. Based on the existing theoretical frameworks of CSSs [18, 273] and our conversational search goals model we formally define a dataset desiderata:

- **Multi-turn dialogues**: the data must contain dialogues with more than one turn of user and agent utterances. Single-turn dialogues do not take into account the process

Table 2.2: Overview of dialogue datasets including their size and conversational search characteristics. [a] *The dialog acts were pre-defined, and the teacher in the setup chooses only one among a few options.* [b] *There are labels for a sample of 2,199 dialogues.* [c] *There are labels for a sample of 1,356 dialogues.*

Name	Venue	Field	#Dialogues	multi-turn	multi-intent	clf. questions	inf. needs	utterance labels	multi-domain	grounded
SCS [333, 334]	CHIIR	IR	39	✓	✓	✓	✓	✓	✓	
MISC [328]	CAIR workshop	IR	88	✓	✓	✓	✓		✓	
CCPE-M [272]	SIGDIAL	DS	502	✓	✓	✓		✓	✓	
Frames [16]	SIGDIAL	DS	1,369	✓	✓	✓		✓		
KVRET [88]	SIGDIAL	DS	3,031	✓	✓			✓		✓
CoQA [279]	preprint only	-	8,000	✓					✓	✓
MultiWOZ [46]	EMNLP	NLP	8,438	✓	✓	✓		✓		
QuAC [59]	EMNLP	NLP	13,594	✓				✓ [a]	✓	✓
WoW [81]	ICLR	ML	22,311	✓	✓				✓	✓
ShARC [296]	EMNLP	NLP	32,436	✓						✓
MSDialog [268]	SIGIR	IR	35,000	✓	✓	✓	✓	✓ [b]		
DSTC-7-SS [382]	DSTC7 workshop	DS	100,000	✓	✓	✓	✓			
UDC [204]	SIGDIAL	DS	930,000	✓	✓	✓	✓			
MANtIS	-	-	80,324	✓	✓	✓	✓	✓ [c]	✓	✓

of elucidating the user information-need.

- **Information needs**: the user must have an information need [323] expressed in her utterances. The conversations must be information-seeking, going beyond lookup, chit-chat and goal-oriented tasks. Conversational *search* is different from general conversational *AI* [101], as there is an underlying information need to be solved.
- **Clarification questions**: the data must present mixed-initiative conversations by going beyond the user-asks/system-responds loop. Clarification questions are essential in elucidating the user information-need.
- **Multi-intent utterances**: another indication of mixed-initiative [273] are utterances that have more than one intent, e.g. giving feedback and presenting further information.
- Having **utterance labels** is a useful resource in building CSSs by providing additional supervision signals.
- **Multi-domain**: the users' information needs can fall into more than one domain (topics of conversation, such as *physics*, *travel* and *English*). Domain specific dialogue systems do not generalize to new/unseen information needs. Thus the dataset must contain conversations from multiple domains.
- **Grounded conversations**: the agent must be able to report the source(s) of the information it is providing and the reasoning behind it. Grounding conversations in documents is a useful resource for achieving explainable agents. Moreover, using sources of information for generating responses has shown to improve the quality of the dialogues over non-grounded conversations that rely only on historical conversational data [401].

With these desiderata in mind, we explored twelve multi-turn, non-chit-chat, human-to-human, open-sourced, and datasets that were released in the five years prior to collecting the corpus, i.e. 2014 - 2019. The result—i.e. the datasets' characteristics according to our desiderata—can be found in Table 2.2. Importantly, none of the datasets have all the desirable features. SCS is the most complete one, missing only the grounding aspect. However, the very limited number of dialogues in this dataset (39) makes it not suitable to train and evaluate conversational search models. The three largest datasets, MSDialog, DSTC-7-SS and UDC, were all derived from technical forums. Their two main drawbacks are the narrow content domain (technical) and lack of a correspondence between utterances and documents where useful information to fulfill the information needs could be extracted from (i.e., grounding). This poses challenges for research on CSS: how generalizable are models trained on one or two particular domains? How can systems leverage the huge amount of available information in web documents—from diverse domains—in information-seeking conversations?

In order to study such challenges we created a novel dataset called MANtIS, short for *multi-domain information seeking dialogues dataset*. MANtIS is to our knowledge the first dataset at large-scale that fulfills all of our dataset desiderata.

2.5 MANtIS

In order to create a large-scale conversational dataset, we resort to the extraction of conversations from existing data sources— the same strategy followed by the creators of the largest datasets in Table 2.2. We take the community question-answering portal Stack Exchange as a starting point[3] as (i) the data dump is publicly available, (ii) it is large-scale (more than 20M questions), (iii) the portal covers diverse domains (so-called *sites*, 175 as of 05/2019) such as *physics*, *travel* and a range of IT and computer science domains, and (iv) the information needs are often complex as posing a question on Stack Exchange usually means that a simple web search is not enough to find a suitable answer.

For MANtIS, we consider 14 diverse domains[4]. We make the source code available at https://github.com/Guzpenha/MANtIS so that conversations from any of the 175 domains of Stack Exchange can be extracted. The examples in Figure 2.2 showcase characteristics of the conversations from our dataset.

Inclusion Criteria

We consider each question-answering thread of a Stack Exchange site as a potential conversation between an information seeker and an information provider and include it in MANtIS if the following six criteria hold:

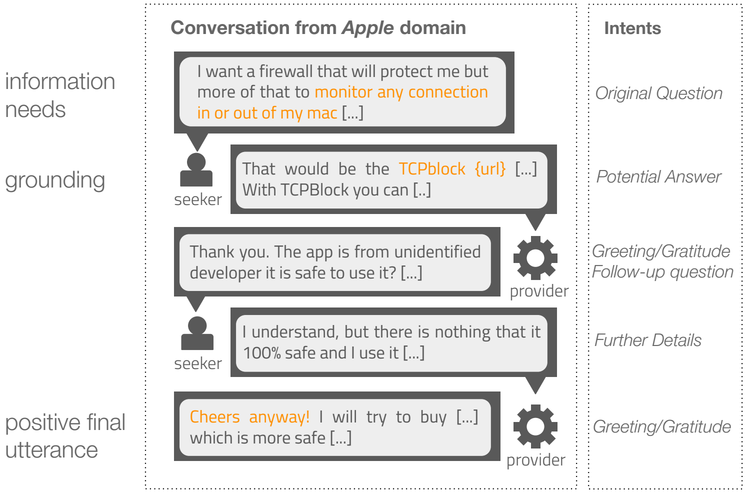
1. The entire conversation takes place between exactly two users (the information *seeker* who starts off the conversation and the information *provider*).

2. The conversation consists of at least 2 utterances per user.

3. One of the provider's utterances contains a hyperlink, providing grounding.

[3] https://archive.org/download/stackexchange data dump from 2019-03-04

[4] Specifically, we consider apple (5,645 dialogues), askubuntu (17,755), dba (5,197), diy (1,528), electronics (10,690), english (3,231), gaming (2,982), gis (9,095), physics (7,826), scifi (2,214), security (3,752), stats (7,676), travel (1,433) and worldbuilding (1,300).



I want a firewall that will protect me but more of that to monitor any connection in or out of my mac [...]

That would be the TCPblock {url} [...] With TCPBlock you can [..]

Thank you. The app is from unidentified developer it is safe to use it? [...]

I understand, but there is nothing that it 100% safe and I use it [...]

Cheers anyway! I will try to buy [...] which is more safe [...]

Figure 2.2: `MANtIS` example with document grounding (url), positive feedback from the information seeker, clarification questions and the initial information need. On the right we display the user intent labels.

4. The conversation has not been marked as *Spam* or *Offensive*.
5. The conversation has not been edited or marked as deprecated.
6. If the final utterance belongs to the seeker, it contains positive feedback.

In order to verify to what extent the existence of a hyperlink can be considered as document grounding (criterium 3), we sampled 150 conversations from `MANtIS` and manually verified whether the link contained in the information provider's utterance(s) is indeed leading to a grounding document. This was the case for 88% of the sampled conversations, which we consider a sufficiently high percentage to not further refine the grounding rule.

In order to verify whether the final say of the information seeker was a positive statement (criterium 6), we sampled 1,400 conversations (100 from each of our sites) where the last person to respond was the information seeker and manually assessed whether the final response was positive feedback (see last utterance of dialogue in Figure 2.2). Subsequently, for all conversations with a final response by the information seeker, we computed the VADER sentiment score [140]. Based on our labeled conversations, we applied a decision stump in order to obtain the optimal score threshold (separately for each site). Consequently, all the conversations with a VADER score below the optimal threshold were discarded—as we are interested in information-seeking conversations that contain a positive conclusion as we assume that in those cases the information need has been fulfilled.

Based on these criteria, we extracted a total of 80,324 conversations. The majority of the conversations have 4 utterances (60%). Some technical domains such as *electronics* and *askubuntu* have a high average number of turns, while other domains such as *worldbuilding* and *dba* have very long utterances showcasing the diversity of the domains. Our list of conditions was quite stringent, only 4.77% of all question-answering threads made it into our final dataset, and each domain contributed at least 1K conversations.

Next, we have some examples of dialogues from the different domains of `MANtIS`.

Dialogue from the `english` **domain**

u^1: I would like to describe a person who returns from a mind-relaxing break back to work by the idiom *fresh pair of eyes*. However, as per its definition on some sources, *a fresh pair of eyes* is *another person [...]*, which made me think that maybe it is not suitable. The situation I am imagining is of a person who worked longer than he/she expected to find the evident (by the incorrect outcome) mistake in his/her work, goes out for a break, then returns back to examine his work again for the mistake. I would like to describe the property of this man/lady being refreshed by the break, and in a concise and effective manner. Does the idiom ”a fresh pair of eyes” fit into this description? If not, then what else should be my phrase of choice?

u^2: Yes, one can take a break so that they return with a fresh pair of eyes, or so that they review the work with a fresh pair of eyes. However, the phrase idiomatically refers to getting someone else to have a look - someone whose preconceptions or perspectives haven’t already been tamed to match that of those close to the project. The free dictionary `http://idioms.thefreedictionary.com/a+fresh+pair+of+eyes` *another person to examine something closely in addition to anyone previously. As soon as we can get a fresh pair of eyes on this mansuscipt, we will find the last of the typos.*

u^3: Should I perhaps than say **almost* fresh pair of eyes*, or just use it without the ”almost” regardless of what it idiomatically refers to?

u^4: No, I don’t think *almost* gets you what you want here. The phrase (and in particular the word fresh) *can* be used in its literal sense, as noted in the opening sentence of my answer. You’re welcome. :)

Dialogue from the `gaming` **domain**

u^1: What kind of pokemon should I place in the gyms? There is this gym defence tier list which has pokemon with high hp in it. `https://i.stack.imgur.com/kMTSc.jpg` Is that the only thing we should look for? Does DPS, attack moves, CP matter or is hp the most important thing?

u^2: There are 3 main criteria: The first being high hp, for obvious reasons. The second criteria is charges that charge fast and have a high base damage to take full advantage of the 1.5 attackrate. The third one is a high Defense-DPS, explained in depth in `https://gaming.stackexchange.com/questions/277288/is-damage-from-defending-pokemon-normalized-for-slow-and-fast-attacks/277364#277364`. But to simplify it: a high base damge is generally better. The highest Tier actually isn’t simply the highest hp Pokémon - it is more of a coincidence that the high hp Pokémon also have a higher base damage.

u^3: So in that case is this tier list inaccurate?

u^4: Not really. I browsed the net a bit and found this reddit post `https://www.reddit.com/r/TheSilphRoad/comments/4skafz/best_attackers_and_defenders_analysis/` with a rather accurate tier list in my opinion, so feel free to check it out.

Dialogue from the `stats` domain

u^1: How do I call a forecast (more precisely, a forecasting rule) that is both accurate and precise? Is there a word that expresses both properties combined?I do not mean the forecasting rule is perfect, i.e. it does not have to produce forecasts that always perfectly coincide with their respective targets, but its accuracy is good (low bias) and its precision too (low variance).

u^2: My guess would be 'consistent forecast'. As you said: How do I call a forecast (more precisely, a forecasting rule) that is both accurate and precise? Quoting Wikipedia (`https://en.wikipedia.org/wiki/Consistency_(statistics)`) on consistency: Use of the terms consistency and consistent in statistics is restricted to cases where essentially the same procedure can be applied to any number of data items. I am taking procedure and rule to be synonymous in this case. And some more: A consistent estimator (`https://en.wikipedia.org/wiki/Consistent_estimator`) is one for which, when the estimate is considered as a random variable indexed by the number n of items in the data set, as n increases the estimates converge to the value that the estimator is designed to estimate. So if the estimate converges to the value the forecasting rule is designed to estimate then it can be called accurate and given the same information the forecasting rule must give precise forecasts.

u^3: This is rather specific (an accurate and precise forecast need not be consistent) and tangential to the part precise (a consistent forecasting rule can be imprecise in any finite sample).

u^4: By imprecise do you mean high standard deviation? I didn't get the first part of your comment I.e. an accurate and precise forecast need not be consistent.

u^5: By imprecise, yes. Consistent means it converges to a perfect forecast; mine need not converge. It can stay about as good as it is for any sample size.

u^6: That means your forecast's goodness has to be independent of sample size. What if the sample size is 1?Vis-à-vis a sample size of lets say 1000?

2.6 Evaluation

In order to evaluate models using `MANtIS` and other information-seeking datasets for conversational search, in this section we first formally define the conversation response ranking task, followed by the limitations of this evaluation scheme.

2.6.1 Conversation Response Ranking

The task of conversation response ranking [83, 115, 129, 130, 246, 319, 367, 373, 374, 384, 398, 402], concerns finding the best response given the dialogue context. Formally, let $\mathscr{D} = \{(\mathscr{U}_i, \mathscr{R}_i, \mathscr{Y}_i)\}_{i=1}^{M}$ be a dataset consisting of M triplets: dialogue context, response candidates and response relevance labels. The dialogue context $\mathscr{U}_i$ is composed of the previous utterances $\{u^1, u^2, ..., u^\tau\}$ at the turn τ of the dialogue. The candidate responses $\mathscr{R}_i = \{r^1, r^2, ..., r^n\}$ are either ground-truth responses or negative sampled candidates, indicated by the relevance labels $\mathscr{Y}_i = \{y^1, y^2, ..., y^n\}$.

Typically, the number of candidates n is way smaller, e.g. $n = 10$, than the number of responses in the collection. When n is small and the model has to score only a few candidate responses we have the re-ranking setup (second-stage retrieval of the pipeline from Figure 1.6). If we consider n to be the size of the entire collection of responses we have the retrieval setup (first-stage retrieval of the pipeline from Figure 1.6). By design the number of ground-truth responses is one, the observed response in the conversational data. The task is then to learn a ranking function $f(.)$ that is able to generate a ranked list for the set of candidate responses $\mathcal{R}_i$ based on their predicted relevance scores $f(\mathcal{U}_i, r)$.

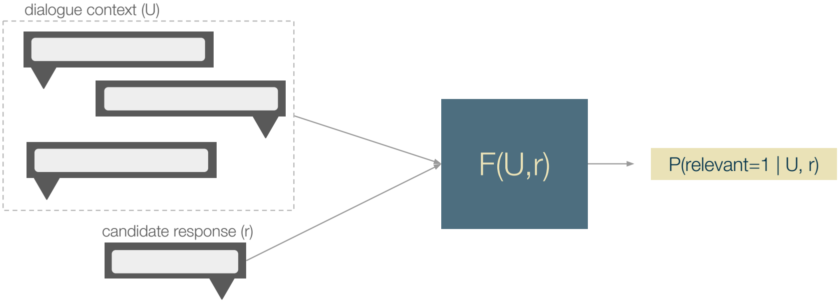


Figure 2.3: Ranking function f predicts the relevance of a candidate response r for the dialogue context $\mathcal{U}$.

Other similar ranking tasks related to conversational search are *clarification question retrieval* [277, 278], where the set of responses to be retrieved are always clarification questions and *conversation passage retrieval* [75, 194]. A successful model for the ranking tasks retrieves the ground-truth response(s) first in the ranked list, and thus the evaluation metrics employed are IR metrics such as MAP and $R_N@K$ (where N is the number of candidate responses and K is the list cutoff threshold).

Premises and Limitations

There are a number of premises and limitations that we would like to highlight in this offline evaluation next.

There is a complete pool of adequate responses that endure over time

Our ranking task assumes access to a pool of responses that contain at least one appropriate answer to a given information need. If we resort only to historical responses the maximum effectiveness of a system would be very low. For example, in popular benchmarks such as UDC [204] and MSDialog [268] the number of responses that exactly match with historical responses are less than 11% and 2% respectively. We also see that such exact matches are often uninformative: 40% are utterances for which the intent is to show gratitude, e.g. '*Thank you!*', compared to the 20% overall rate in MSDialog. Another concern is that responses that were never given before, e.g. questions about a recent Windows update, would not be answerable by such a system even though this information might be available on the web.

The correct answer is always in the candidate responses list.

Neural ranking models are generally employed for the task of re-ranking a set of documents, obtained from a recall-oriented and efficient first-stage ranker [387]. While such a multi-stage approach offers a practical approach for conversational response ranking, benchmarks always include the relevant response in the candidate list to be retrieved.

The effectiveness of models for small collections generalizes to large collections.
While in ad-hoc retrieval we have to rank from a pool of millions of documents, current benchmarks require models to retrieve responses from a list of 10–100 candidates (12 out of 13 use less than 100 candidates, and 7 use only 10 candidates). This makes the task unreasonably easy, as demonstrated by the 80% drop in performance from subtask 5 (120000 candidates) and subtask 2 (100 candidates) of DTSC7-NOESIS [118]. In Chapter 3 we evaluate the effectiveness of models for the full-rank task where the number of candidates is the complete collection.

Test instances from the same dialogue are considered as independent.
When creating conversational datasets [129, 204, 268] the default is to generate multiple instances from one dialogue: one instance for each answer provided by the information provider composed of the last information seeker utterance, and the dialogue history—see Figure 1.3. Even though multiple utterances come from the same dialogue, they are evaluated independently, e.g. an inappropriate response at the beginning of a conversation does not change the evaluation of a response given later by the system in the same dialogue. Benchmarks evaluate instances from the same dialogue independently. In a real-world scenario, if a model fails at the start of the conversation, it has to recover from unsatisfactory responses.

There is only one adequate answer.
Traditional offline evaluation cannot handle counterfactuals [37] such as what would have happened if another response was given instead of the ground-truth one. Due to the high cost of human labels, it is common to use only one relevant response per context (the observed human response). However, multiple responses could be correct for a given context with different levels of relevance. Multiple answers can be right because they provide semantically similar responses or because they are different but appropriate responses to an information need.

2.7 The `transformer-rankers` Library

In this section, we describe the three main modules of the `transformer-rankers` library[5]: datasets, transformer rankers, and negative sampling. The core task supported is *conversation response ranking* as defined in Section 2.6. For example, it is possible to download the `MANtIS` *dataset* and fine-tune a `BERT` [80] *re-ranker model* with BM25 to obtain *negative samples* with a few lines of code[6].

2.7.1 Dialogue Datasets

It is possible to download a number of datasets in `transformer-rankers`[7], including three information-seeking dialogue datasets used in most chapters of the thesis:

- `MANtIS` [246] the dataset introduced in Section 2.5.

[5] `https://github.com/Guzpenha/transformer_rankers`

[6] See for example this Google Colab notebook: `https://colab.research.google.com/drive/1wGmaO3emC7Sg-tA7nGehIQ2vjOLN9S5e?usp=sharing`.

[7] See all datasets here: `https://github.com/Guzpenha/transformer_rankers/blob/master/transformer_rankers/datasets/downloader.py`

- MSDialog [268] which contains 246K context-response pairs, built from 35.5K information-seeking conversations from the Microsoft Answer community, a QA forum for several Microsoft products;
- UDC-DSTC8 [175] which contains 184k context-response pairs of disentangled Ubuntu IRC dialogues.

See for example how to download those three datasets with transformer-rankers:

```python
from transformer_rankers.datasets import downloader
data_folder = './datasets'
for name in ['mantis', 'msdialog', 'ubuntu_dstc8']:
    dataDownloader = downloader.DataDownloader(name, data_folder)
    dataDownloader.download_and_preprocess()
```

2.7.2 Transformer for Ranking

The multi-stage pipeline described in the introduction to produce a retrieval-based conversational search system requires a first-stage retrieval system that selects a number of candidates from the entire pool of responses that can be re-ranked later. In Chapter 3 we describe approaches for the first-stage retrieval, whereas in later chapters we focus on re-ranking. Re-ranking with a transformer model that has as input both the query and the document, also known as a cross-encoder, has been a really successful approach to numerous IR tasks, including conversation response ranking[8]. A strong baseline for the task is BERT [80], which is used throughout this thesis[9]. The transformer-rankers implementation relies on the Hugging Face library [363].

In Figure 2.4 we show how a dialogue context and a candidate response are concatenated as input to a BERT re-ranker in order to obtain a prediction of relevance. Each dialogue context $\mathscr{U}$ contains only one utterance per seeker/provider for each conversational turn. In cases where the concatenation of ($\mathscr{U}$[SEP]r) is bigger than the input limit, we truncate $\mathscr{U}$ from the left to the right.

2.7.3 Negative Sampling

With transformer-rankers there are three different negative sampling approaches implemented[10]: random, BM25, and dense retrieval. Note that since they are required to perform retrieval, they are capable of doing the first-stage step in pipelines. Section 1.1.3 gives an overview of negative sampling procedures.

[8] See for example https://github.com/JasonForJoy/Leaderboards-for-Multi-Turn-Response-Selection

[9] See the following report on getting baseline results using transformer-rankers BERT re-rankers for all the three information-seeking datasets employed here: https://wandb.ai/guz/library-crr-bert-baseline/reports/BERT-ranker-baselines-for-CRR--Vmlldzo0NDcyMzU.

[10] https://github.com/Guzpenha/transformer_rankers/blob/master/transformer_rankers/negative_samplers/negative_sampling.py

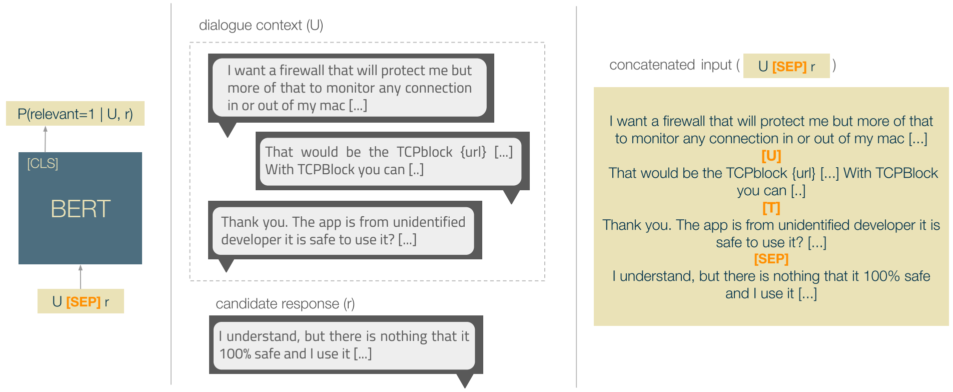

P(relevant=1 | U, r)
[CLS]
BERT
U [SEP] r
dialogue context (U)
I want a firewall that will protect me but more of that to monitor any connection in or out of my mac [...]
That would be the TCPblock {url} [...] With TCPBlock you can [..]
Thank you. The app is from unidentified developer it is safe to use it? [...]
candidate response (r)
I understand, but there is nothing that it 100% safe and I use it [...]
concatenated input (U [SEP] r)
I want a firewall that will protect me but more of that to monitor any connection in or out of my mac [...]
[U]
That would be the TCPblock {url} [...] With TCPBlock you can [..]
[T]
Thank you. The app is from unidentified developer it is safe to use it? [...]
[SEP]
I understand, but there is nothing that it 100% safe and I use it [...]

Figure 2.4: Using cross-encoder BERT re-ranker to estimate the relevance of a pair of dialogue context $\mathcal{U}$ and the candidate response r. On the left, we have a diagram of the inputs and outputs of the model. In the middle, we have an example dialogue context and candidate response. On the right, we have the same example as the input to the model. The input is their concatenation with a `[SEP]` token. The dialogue context $\mathcal{U}$ is represented by the concatenation of its utterances, separated by the special end of utterances and turns tokens: `[U]` and `[T]`.

2.8 Conclusions

We proposed here a model of conversational search that focuses on the main goals of the agent and user interactions. We identified two major challenges: (1) the collaboration of efforts in the research fields of IR, NLP, and DS, and (2) the lack of publicly available large-scale conversational search datasets. Based on a set of dataset desiderata, we introduce `MANtIS`, a large-scale dataset that contains more than 80K conversations across 14 domains that are multi-turn, centered around complex information needs, and are mixed-initiative.

We also describe the core task that we use for the evaluation of ranking models throughout this thesis. We introduce the `transformer-rankers` library to train and evaluate transformer models for the task, going through the main components of datasets, transformer rankers, and negative sampling.

Having described the main resources used in this thesis, next, we dive into retrieval and ranking models for conversational search. We begin with the first-stage retrieval of responses for dialogues in the next chapter.

III

Retrieval and Ranking for Conversational Search

3

Representations for First-Stage Retrieval of Responses

In this chapter, we focus on the first stage of the multi-stage pipeline for conversational search. The predominance of the re-ranking task in previous work has led to a great deal of attention to building neural re-rankers*, while the first-stage retrieval step has been overlooked. Since the correct response is always available in the candidate list of n, this artificial re-ranking evaluation setup assumes that there is a first-stage retrieval step that is always able to rank the correct response in its top-n list. In this chapter, we focus on the more realistic task of* full-rank retrieval of responses*, where n can be up to millions of responses. We investigate both dialogue context and response expansion techniques to augment sparse representations for retrieval, as well as zero-shot and fine-tuned dense representations for retrieval. Our findings—based on three different information-seeking dialogue datasets—reveal that a learned response expansion technique is a solid baseline for sparse retrieval. We find the best-performing method overall to be dense retrieval with intermediate training—a step after the language model pre-training where sentence representations are learned—followed by fine-tuning on the target conversational data. We also look into hypotheses that could explain why we observed the phenomena of harder negatives sampling techniques leading to worse results for the fine-tuned dense retrieval models. The code required to reproduce this chapter is available at* `https://github.com/Guzpenha/transformer_rankers/tree/full_rank_retrieval_dialogues`.

This chapter is based on the following paper:

- *Gustavo Penha and Claudia Hauff. 2023. Do the Findings of Document and Passage Retrieval Generalize to the Retrieval of Responses for Dialogues? In ECIR. Springer, 132–147. [253]*

3.1 Introduction

The offline evaluation of neural ranking models for conversational response ranking is to rank the ground-truth response over a limited set of n responses and measure the number of relevant responses found in the first K positions—$Recall_n@K$ [399]. Since the entire collection of available responses is typically way bigger[1] than such set of candidates, this setup is in fact a re-ranking problem, where we have to select the best response out of a few options. Additionally, in existing benchmarks the correct response is traditionally always amongst the n responses to re-rank [248]. This is thus an artificial evaluation that overlooks the first-stage retrieval step, which needs to retrieve the n responses that will be later re-ranked. If the first-stage model, e.g. BM25 [290], fails to retrieve relevant responses, the *retrieve then re-rank* pipeline will also fail.

In this chapter, we contribute a novel comparison of supervised and unsupervised, dense and sparse retrieval models[2] for the overlooked problem of *full-rank retrieval of responses for dialogues.* We adapt prominent techniques for the problem, i.e. effective in other ranking tasks such as passage retrieval, including document expansion for the task of ranking responses for dialogue contexts.

We contribute here an empirical evidence to the following open questions when setting up a full-rank retrieval system for conversation response ranking. What is the effectiveness of sparse and dense retrieval when ranking responses from the *entire collection*? How do dense models compare with strong sparse baselines? What is their effectiveness in a zero-shot setup? What is the effect of adding an intermediate representation learning step between the language model pre-training and the training with conversational data?

We also shed light on the important problem of selecting negative samples when training dense retrieval models, which is known to have a great effect on the final effectiveness in different ranking tasks [370, 392]. Unlike previous work that studies sampling out of a few random conversational responses in the re-ranking setup of a cross-encoder model [184], we study the harder problem of sampling negative responses from the entire collection. We are the first to investigate different hypotheses in the context of negative sampling of responses for dialogues that can explain difficulties in using harder negatives in the training of dense retrievers. Our main findings in building retrieval models of responses for dialogues in the full-rank setting are:

- While dialogue context expansion is not successful for sparse retrieval, supervised response expansion through the proposed $resp2ctxt_{lu}$ is a strong baseline for full-rank retrieval of responses for dialogues.

- Dense retrieval without access to the target dialogue data, i.e. the zero-shot scenario, is able to beat a strong sparse baseline only when it has access to a large amount of out-of-domain supervision data.

[1] While for most benchmarks [399] we have only 10–100 candidates, a working system with the Reddit dataset from PolyAI `https://github.com/PolyAI-LDN/conversational-datasets` for example would need to retrieve from 3.7 billion responses.

[2] Although we evaluate them as standalone methods for the full-rank retrieval problem, they can also be employed as first-stage retrievers followed by a re-ranking step.

- Dense retrieval models that have intermediate training followed by fine-tuning with the target data are the best-performing models, even with a simple random sampling approach for obtaining negative responses.

- Harder negative sampling techniques lead to worse effectiveness. We found evidence indicating that false positives strongly contribute to this phenomenon. Denoising is an effective approach for taking advantage of harder negative samples.

3.2 Related Work

In this section, we analyze previous work pertinent to this paper by first discussing current research in (un)supervised dense and sparse retrieval followed by reviewing work on re-ranking and retrieval models for responses.

3.2.1 Dense and Sparse Retrieval

The proposed conceptual framework by Lin [190] argues for categorizing retrieval models into two dimensions: supervised vs. unsupervised and dense vs. sparse representations[3]. An *unsupervised* sparse representation model such as BM25 and TF-IDF [156] represents each document and query with a sparse vector with the dimension of the collection's vocabulary, having many zero weights due to non-occurring terms. Since the weights of each term are based on term statistics they are unsupervised methods.

A *supervised* sparse retrieval model such as COIL [105], SPLADE [94], TILDE [406] and DeepImpact [216] can take advantage of the effectiveness of transformer-based language models by changing the terms' weights from collection statistics to something that is learned. DeepCT [71] for example learns term weights with a transformer-based regression model from the supervision of the `MSMarco` dataset. Approaches that only modify non-zero weights however are not able to address the vocabulary mismatch problem [98], as terms with zero weight will not be affected. One way to address such a problem in sparse retrieval is by using query expansion methods. RM3 [1] has been shown to be a competitive query expansion technique that uses pseudo-relevance feedback to add new terms to the queries followed by another final retrieval step using the modified query.

Document expansion has also been shown to be an effective technique to improve sparse retrieval, which is able to address the vocabulary mismatch problem. The core idea is to create pseudo documents that have expanded terms and use them instead when doing retrieval. doc2query [238] is an effective approach to document expansion that uses a language model to predict the queries which might be issued to find the document. The predictions of this model are used to create the augmented pseudo documents. Expansion techniques are able to modify non-zero weights by adding terms that did not exist in the query or document.

Supervised dense retrieval models, such as ANCE [370], RocketQA [271], PAIR [281] and coCodenser [104], represent query and documents in a smaller fixed-length space, for example of 768 dimensions, which can naturally capture semantics. They are thus able to address the vocabulary mismatch problem. While dense retrieval models have

[3] A distinction can also be made of cross-encoders and bi-encoders, where the first encode the query and document jointly as opposed to separately [325]. Cross-encoders are applied in a re-ranking step due to their inefficiency and thus are not our focus.

shown to consistently outperform BM25, this is not so easily the case when dense retrieval models do not have access to training data from the target task, known as the *zero-shot scenario* [282, 326]. The BEIR benchmark [326] showed that BM25 was superior to dense retrieval from 9–18 (depending on the model) out of the 18 datasets under this evaluation scheme. While the zero-shot scenario offers a fairer comparison of dense models with unsupervised sparse models, learned dense retrieval models should also be compared with learned sparse models, e.g. BM25+doc2query.

Unlike previous work that compares supervised and unsupervised, dense and sparse retrieval models for other tasks such as passage ranking, we provide a novel and comprehensive comparison for the problem of *full-rank retrieval of responses for dialogues*.

3.2.2 Re-Ranking and Retrieval of Responses for Dialogues

Early neural models for response ranking were based on matching the representations of the concatenated dialogue context and the representation of a response in a single-turn manner with architectures such as CNN and LSTM [159, 204]. Researchers later explored matching each utterance in the dialogue context with the response with more complex neural architectures [114, 195, 367, 371, 402].

Using heavily pre-trained language models for ranking was first shown to be effective by Nogueira and Cho [236]. They used a BERT model to re-rank the responses of a first-stage retrieval system on the MSMarco passage retrieval task and showed significant improvements in effectiveness. Such language models for ranking have quickly become a predominant approach in information retrieval [193]. This was also shown to be effective for re-ranking responses in conversations. We were amongst the first to show a way of using a BERT-based re-ranking model for the dialogues domain (see Chapter 4).

One limitation of transformer-based language models is that they do not take into account the structure of the dialogue. Gu et al. [113] proposed adding another embedding layer to BERT that takes into account the speaker of the dialogue. Dialogue-aware training has also been further explored, for example, both by Han et al. [126] and Whang et al. [361] who proposed different modifications to the conversational data to improve the fine-tuning of language models. Building better re-ranking models for dialogue tasks is still an active research field as seen by recent surveys on the topic [318, 399].

In contrast, full-rank retrieval of responses, i.e. the first-stage retrieval step, has been under-explored [248]. Lan et al. [178] showed that a BERT-based dense retrieval model outperforms BM25 on the full-rank task. Tao et al. [317] later proposed a mutual learning model that trains both the dense retrieval bi-encoder model and the cross-encoder re-ranker model at the same time. They also showed that such a dense model is more effective than BM25 without expansion techniques for the full-rank problem of retrieving responses for dialogues.

A limitation of previous work is that a strong sparse retrieval baseline model, e.g. BM25+*dialogue context expansion* or BM25+*response expansion*, was not compared. Such methods are capable of mitigating the vocabulary mismatch and thus the question if dense models are able to outperform sparse ones when using expansion techniques is still unanswered. We expand on the analysis of previous work [178, 317] by looking into stronger sparse baselines, evaluating the effect of intermediate training, testing zero-shot effectiveness of dense models, and studying the effect of other negative sampling methods.

3.3 Full-rank Retrieval for Dialogues

In this section, we first describe the problem of full-rank retrieval of responses, followed by the proposed sparse and then dense approaches.

3.3.1 Problem Definition

The full-rank retrieval of responses for dialogue is a particular case of the conversation response ranking task (defined in Section 2.6.1), where the candidate list is the entire set of responses from the collection. In previous work, the number of candidates is limited, typically $n = 10$. Since we are concerned with the full-rank task and not the re-ranking setting, in our experiments n is the number of responses available in the collection.

3.3.2 Sparse Retrieval

In order to do sparse retrieval of responses we rely on classical retrieval methods with query and document expansion techniques. One of the limitations of sparse retrieval is that, since it represents each dialogue context and response using the existing terms in a bag-of-words manner, the vocabulary mismatch problem might occur. Such expansion techniques are able to overcome this problem if they append new words to the dialogue contexts and responses.

For this reason, we propose here to do *dialogue context expansion* with RM3 [1], a competitive unsupervised method that assumes that the top-ranked responses by the sparse retrieval model are relevant. From such pseudo-relevant responses, words are selected and an expanded dialogue context is generated, and then used by the sparse retrieval method to generate the final ranked list.

In order to expand the responses to be retrieved, we propose $resp2ctxt$. This is an adaptation of the effective *doc2query* [238] approach for dialogues. Formally, we fine-tune a generative transformer model for the task of generating the dialogue context $\mathcal{U}_i$ from the ground-truth response r^+. This model is then used to generate expansions for all responses in the collection. They are appended to the responses and the sparse retrieval method itself is not modified. $resp2ctxt$ allows for two things: term re-weighting (adding terms that already exist in the document) and the addition of new terms (to deal with the vocabulary mismatch problem).

Unlike most ad-hoc retrieval problems where the queries are smaller than the documents, full-rank retrieval of responses for dialogues is the exact opposite. For example, while the `TREC-DL-2020` passage and document retrieval tasks the queries have between 5–6 terms on average and the passages and documents have over 50 and 1000 terms respectively, the dialogue contexts (queries) have between 70 and 474 terms on average depending on the dataset while the responses (documents) have between 11 and 71 terms on average, as seen in the first two rows of Table 3.2. This is a challenge for the generative model since generating larger pieces of text is a more difficult problem than smaller ones, e.g. more room for error.

Motivated by this, we also explored an adaptation of $resp2ctxt$ that aims to generate only the last utterance of the dialogue context: $resp2ctxt_{lu}$. This model is trained to generate u^{τ} from $r+$. The underlying premise is that the part that needs to be answered by the dialogue context is the last utterance, and if this is correctly generated by $resp2ctxt_{lu}$, the sparse retrieval method will be able to find the correct response from the collection.

3.3.3 Dense Retrieval

In order to do dense retrieval we rely on methods that learn to represent the dialogue context and the responses separately in a dense embedding space. Responses are then ranked by their similarity to the dialogue context. We rely here on pre-trained language transformer models, such as BERT [80], RoBERTa [201] or MPNet [308], to obtain such representations of the dialogue context and response. This approach is generally referred as a *bi-encoder* model [193].

Intermediate Training

The first step of the pipeline is to train the representations of the language model with intermediate[4] data that does not contain the target domain data. Such intermediate data contains triplets of query, relevant document, and negative document and can include multiple datasets. The main advantage of adding this step before fine-tuning the bi-encoder for the target conversational data is to reduce the gap between the pre-training, often including language modeling, and the downstream task at hand.

The intermediate training step learns to represent texts (query and documents) by doing a mean pooling function over the transformer's final layer, which is then used to calculate the dot-product similarity. The relevant document representation is used to contrast with the representations of the document that is not relevant. Such a procedure learns better text representations than a naive approach of simply using the `[CLS]` token representation of BERT for the dialogue contexts and responses [4, 280].

The loss function employs multiple negative texts to learn the representations in a contrastive manner, also known as in-batch negative sampling. This model is then able to do zero-shot retrieval for the full-rank retrieval of responses to dialogue contexts since it does not have access to the target domain data.

The function $f(\mathcal{U}, r)$ can be defined as $dot(\eta(concat(\mathcal{U})), \eta(r))$, where η is the representation obtained with the mean pooling of all the output vectors of the transformers language model, and $concat(\mathcal{U}) = u^1 \mid \texttt{[U]} \mid u^2 \mid \texttt{[T]} \mid ... \mid u^\tau$, where | indicates the concatenation operation. The utterances from the context $\mathcal{U}$ are concatenated with special separator tokens `[U]` and `[T]` indicating end of utterances and turns[5].

Fine-tuning

The second step in the pipeline is to fine-tune the model with data from the target domain: dialogue contexts and responses. Since we do not have labeled negative responses and only relevant ones, the remaining responses can be thought of as non-relevant to the dialogue context. Computing the probability of the correct response over all other responses in the dataset would give us $P(r \mid \mathcal{U}) = \frac{P(\mathcal{U},r)}{\sum_k P(\mathcal{U},r_k)}$. Since this computation is prohibitively expensive to calculate, we approximate it using only a few negative samples retrieved by a negative sampling approach.

The *negative sampling* task is then to: given the dialogue context $\mathcal{U}$ find challenging responses that are not relevant. This can be seen as a retrieval task as well, where one can

[4] We differentiate this intermediate step to a pre-training step. The transformer-based language models were first pre-trained for their respective language modeling tasks. For example, BERT is pre-trained for next-sentence prediction and masked language modeling and can be later trained to represent queries and documents.

[5] The special tokens `[U]` and `[T]` will not have any meaningful representation in the zero-shot setting, but they can be learned in the fine-tuning step.

use a retrieval model to find negatives by applying $f(\mathcal{U}, r)$ for every r in the collection, sorting, and removing r^+ from the resulting top negatives. With such a dataset at hand, we continue the training—after the intermediate step—in the same manner as done by the intermediate training step, with the following cross-entropy loss function[6] for a batch with size B: $\mathcal{J}(\mathcal{U}, \mathbf{r}, \theta) = -\frac{1}{B}\sum_{i=1}^{B}\left[f(\mathcal{U}_i, r_i) - \log\sum_{j=1, j!=i}^{B} e^{f\left(\mathcal{U}_i, r_j\right)}\right]$, where $f(\mathcal{U}, r)$ is the dot-product of the mean pooled representation of the transformer model.

3.4 Experimental Setup

In order to compare the different sparse and dense approaches we consider three large-scale information-seeking conversation datasets introduced in Section 2.7.1: `MANtIS`, `MS-Dialog`, and `UDC-DSTC8`.

3.4.1 Implementation Details

For BM25 and BM25+RM3 we rely on the pyserini implementations [192]. In order to train *resp2ctxt* expansion methods we rely on the Huggingface transformers library [363], using the `t5-base` model. For all methods, we use default hyperparameters from either the original paper or library and perform no parameter optimization. We fine-tune the T5 model for 2 epochs, with a learning rate of 2e-5, weight decay of 0.01, and batch size of 5. When augmenting the responses with *resp2ctxt* we follow docT5query [238] and append three different context predictions, using sampling and keeping the top-10 highest probability vocabulary tokens.

For the zero-shot dense retrieval models, we rely on the SentenceTransformers [280] model releases[7]. The library uses Hugginface transformers for the pre-trained models such as BERT [80], RoBERTa [201], MPNet [308]. When fine-tuning the dense retrieval models, we rely on the *MultipleNegativesRankingLoss*, which accepts a number of hard negatives and also uses the remaining in-batch random negatives to train the model. We use a total of 10 negative samples for dialogue context.

We fine-tune the dense models for a total of 10k steps, and for every 100 steps, we evaluate the models on a re-ranking task that selects the relevant response out of 10 responses. We use the re-ranking validation MAP to select the best model from the whole training to use in evaluation. We use a batch size of 5, with 10% of the training steps as warmup steps. The learning rate is 2e-5 and a weight decay of 0.01. We use the FAISS [155] library to perform the similarity search.

In the follow-up experiments to investigate negative sampling, we denoise negatives (E2) using lists of 100 responses and keep the bottom 10 as negatives. We expand the collection with an external corpus (E5) using ConvoKit [54]. We choose datasets that have similar topics to the information-seeking datasets we use[8], amounting to a total of 17M non-empty candidate responses. For experiment E6 we generate the negative candidates

[6] We refer to this loss as MultipleNegativesRankingLoss.

[7] `https://www.sbert.net/docs/pretrained_models.html`

[8] Namely `movie-corpus`, `wiki-corpus`, `subreddit-Ubuntu`, `subreddit-microsoft`, `subreddit-apple`, `subreddit-Database`, `subreddit-DIY`, `subreddit-electronics`, `subreddit-ENGLISH`, `subreddit-gis`, `subreddit-Physics`, `subreddit-scifi`, `subreddit-statistics`, `subreddit-travel` and `subreddit-worldbuilding`.

using Hugginface [363] conversational pipelines, with the pre-trained models `DialoGPT-large` and `blenderbot-400M-distill`.

3.4.2 Evaluation

To evaluate the effectiveness of the retrieval systems, instead of resorting to the standard evaluation metric in conversation response ranking [113, 319, 384] which is recall at position K with n candidates[9] $R_n@K$, we set n to be the entire collection of answers, and thus we evaluate the model's effectiveness in finding the correct response out of the whole possible set of responses: $R@K$. While the first-stage retrieval component can be coupled with another re-ranking stage that focuses on precision, we consider here the case where we do not have a re-ranking stage and evaluate the capability of the approaches to perform the task as stand-alone models. For this reason, we use R@1 and R@10. We perform Student t-tests at a confidence level of 0.95 with Bonferroni correction to compare the statistical significance of methods.

3.5 Results

In this section, we first report on both dense and sparse retrieval results. Then we analyze the negative sampling procedure used to train the dense retrieval models.

3.5.1 Sparse Retrieval

In order to compare supervised and unsupervised sparse retrieval methods as well as zero-shot and fine-tuned dense retrieval models, we divided them into four categories as shown in Table 3.1. Each row is a retrieval approach, containing the effectiveness in terms of R@1 and R@10 for each of the three datasets.

Does dialogue context expansion via RM3 lead to improvements over no expansion for sparse retrieval?

`BM25+RM3` (row 1b) does not improve over `BM25` (1a) on any of the three conversational datasets analyzed. A thorough hyperparameter fine-tuning was performed and no combination of the `RM3` hyperparameters outperformed `BM25`.

A manual analysis of the new terms appended to a sample of 60 dialogue contexts reveals that only 18% of them have at least one relevant term added based on our best judgment. Unlike web search where the query is often incomplete, under-specified, and ambiguous, in the information-seeking datasets employed here the dialogue context (query) is quite detailed and has more terms than the responses (documents).

We hypothesize that because the dialogue contexts are already quite descriptive the task of expansion is trickier in this domain and thus we observe many dialogues for which the terms added are just noise.

Does response expansion, i.e. *resp2ctxt*, lead to improvements over no expansion for sparse retrieval?

We find that response expansion helped in two of the three datasets tested. `BM25+`*resp2ctxt* (2a) outperforms `BM25` (1a) in two of the three datasets. Predicting only the last utterance of

[9] For example $R_{10}@1$ indicates the number of relevant responses found at the first position when the model has to rank 10 candidate responses.

Table 3.1: Effectiveness of sparse and dense retrieval for the retrieval of responses for dialogues. Bold values indicate the highest recall for each type of approach. Superscripts indicate statistically significant improvements using Students t-test with Bonferroni correction. *†=other methods from the same group;* 1*=best from unsupervised sparse retrieval ;* 2*=best from supervised sparse retrieval;* 3*=best from zero-shot dense retrieval.*

		MANtIS		MSDialog		UDC-DSTC8	
		R@1	**R@10**	**R@1**	**R@10**	**R@1**	**R@10**
(0)	Random	0.000	0.000	0.000	0.001	0.000	0.001
Unsupervised sparse							
(1a)	BM25	**0.133**†	**0.299**†	**0.064**†	**0.177**†	**0.027**†	**0.070**†
(1b)	BM25 + RM3	0.073	0.206	0.035	0.127	0.011	0.049
Supervised sparse							
(2a)	BM25 + $resp2ctxt$	0.135	0.309	0.074	**0.208**	0.028	0.067
(2b)	BM25 + $resp2ctxt_{lu}$	**0.147**$^{\dagger 1}$	**0.325**$^{\dagger 1}$	**0.075**^{1}	0.202^{1}	**0.029**	**0.076**
Zero-shot dense **Model**$_{IntermediateData}$							
(3a)	ANCE$_{600K\text{-MSMarco-PR}}$	0.048	0.111	0.050	0.124	0.010	0.028
(3b)	TAS-B$_{400K\text{-MSMarco-PR}}$	0.062	0.143	0.060	0.157	0.019	0.050
(3c)	Bi-encoder$_{500k\text{-MSMarco-QA}}$	0.038	0.098	0.043	0.113	0.014	0.040
(3d)	Bi-encoder$_{215M-mul}$	0.138	0.297	0.108	0.277	0.023	0.076
(3e)	Bi-encoder$_{1.17B-mul}$	**0.155**$^{\dagger 1}$	**0.341**$^{\dagger 12}$	**0.147**$^{\dagger 12}$	**0.339**$^{\dagger 12}$	**0.041**†	**0.097**$^{\dagger 12}$
Fine-tuned dense **Model**$_{NegativeSampler}$							
(4a)	Bi-encoder$_{Random(0)}$	**0.130**†	**0.307**†	**0.168**$^{\dagger 123}$	**0.387**$^{\dagger 123}$	**0.050**$^{\dagger 12}$	**0.128**$^{\dagger 123}$
(4b)	Bi-encoder$_{\text{BM25}(1a)}$	0.112	0.271	0.128	0.316	0.027	0.087
(4c)	Bi-encoder$_{Bi-encoder(3e)}$	0.065	0.146	0.144	0.306	0.018	0.051

the dialogue ($resp2ctxt_{lu}$) performs better than predicting the whole utterance, as shown by BM25+$resp2ctxt_{lu}$'s (2b) higher recall values. For example, in the MANtIS dataset the R@10 goes from 0.309 when using the model trained to predict the dialogue context, to 0.325 when using the one trained to predict only the last utterance of the dialogue context.

In order to understand what the response expansion methods are doing most—term re-weighting or adding novel terms—we present the percentage of novel terms added by both methods in Table 3.2. The table shows that $resp2ctxt_{lu}$ does more term re-weighting than adding new words when compared to $resp2ctxt$ (53% and 70% on average are new words respectively and thus 47% vs 30% are changing the weights by adding existing words), generating overall smaller augmentations (115.45 vs 431.17 on average respectively).

In terms of sparse retrieval, the experiments so far reveal that using a response augmentation technique is a much better baseline than using BM25, which has been used as a strong baseline for comparison with dense models in dialogue benchmarks [178, 317].

3.5.2 Dense Retrieval

Can zero-shot dense retrieval outperform a strong sparse baseline?

Zero-shot dense retrieval, i.e. no access to target data, beats the strong sparse baseline BM25+$resp2ctxt$ (2b) *only when it is fine-tuned on large datasets containing diverse data*

Table 3.2: Statistics of the augmentations for the response (document) expansion methods *resp2ctxt* and *resp2ctxt*$_{lu}$.

	MANtIS	MSDialog	UDC-DSTC8
Context avg length	474.12	426.08	76.95
Response avg length	42.58	71.38	11.06
Augmentation average length - *resp2ctxt*	494.23	596.99	202.3
Augmentation average length - *resp2ctxt*$_{lu}$	138.5	135.29	72.57
Percentage of new words - *resp2ctxt*	71%	69%	71%
Percentage of new words - *resp2ctxt*$_{lu}$	59%	37%	63%

including dialogues, as we see by comparing rows (3a–c) and (3e–d) with row (2b) in Table 3.1. For example, while the zero-shot dense retrieval models based only on the MSMarco dataset (3a–c) perform on average 35% worse than the strong sparse baseline (2b) in terms of R@10 for the MSDialog dataset, the zero-shot model trained with 1.17B instances on diverse data (3e) is 68% better than the strong sparse baseline (2b). When using a bigger amount of intermediate training data[10], we see that the zero-shot dense retrieval model (3e) is able to outperform the sparse retrieval baseline by margins of 33% of R@10 on average across the datasets.

As expected, the closer the intermediate training data distribution is to the target domain, the better the dense retrieval model performs. The results indicate that a good zero-shot retrieval model needs to be trained for representation learning on a large set of datasets to outperform strong sparse retrieval baselines. Our results match previous empirical evidence on the effect of the intermediate training step on dense retrieval for different retrieval tasks [240].

Is intermediate training of dense retrieval models helpful or is it sufficient to fine-tune a dense model on the target data?
Intermediate training on a large set of training instances is quite important for learning dense representations. Table 3.3 compares the dense models using either different pre-trained language models with and without using the intermediate data, with a different number of negative sampling procedures.

We see that if we fine-tune mpnet-base directly on the target data, and do not do any intermediate training step the effectiveness drops are significant and substantial as shown when comparing results of 1.17B mul. sources (rows 1—3) vs no intermediate data (rows 4–6) in Table 3.3. For example, in the MANtIS dataset the R@10 goes from 0.307 to 0.172 when using random negative sampling. This also happens for other language models and intermediate datasets, e.g. for bert-base and MSMarco the R@10 goes from 0.205 to 0.092 the MANtIS dataset.

What is the effect of fine-tuning the dense model after the intermediate training?
First, we see that simply using random sampling to find negatives and then fine-tuning the dense retrieval model that had already gone through intermediate training—row (4a)

[10] For the full description of the intermediate data see https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

Table 3.3: Effectiveness of fine-tuned dense retrieval models when using different language models and intermediate training for each negative sampling procedures from Table 3.1. Bold indicates the highest value within different negative sampling methods for the same setting. We observe the same phenomena of decreasing effectiveness for better negative sampling methods when using different language models and whether using intermediate training or not.

			MANtIS		MSDialog		UDC-DSTC8	
Model	**Intermediate data**	**Neg. Sampler**	**R@1**	**R@10**	**R@1**	**R@10**	**R@1**	**R@10**
Bi-encoder$_{mpnet-base}$	1.17B mul.sources	Random (0)	**0.130**	**0.307**	**0.168**	**0.387**	**0.050**	**0.128**
		BM25 (1a)	0.112	0.271	0.128	0.316	0.027	0.087
		Bi-encoder (3e)	0.065	0.146	0.144	0.306	0.018	0.051
	-	Random (0)	**0.070**	**0.172**	**0.114**	**0.308**	**0.021**	**0.063**
		BM25 (1a)	0.043	0.118	0.091	0.256	0.009	0.027
		Bi-encoder (3e)	0.032	0.087	0.083	0.205	0.002	0.019
Bi-encoder$_{bert-base}$	500K MSMarco-QA	Random (0)	**0.085**	**0.205**	**0.138**	**0.339**	**0.030**	**0.079**
		BM25 (1a)	0.051	0.130	0.116	0.287	0.007	0.022
		Bi-encoder (3e)	0.043	0.106	0.107	0.242	0.008	0.030
	-	Random (0)	**0.029**	**0.092**	**0.063**	**0.200**	**0.012**	**0.038**
		BM25 (1a)	0.017	0.057	0.040	0.144	0.002	0.006
		Bi-encoder (3e)	0.011	0.041	0.034	0.119	0.000	0.009

in Table 3.1—achieves the best overall effectiveness we obtain in two of the three datasets. Having access to the target conversational data as opposed to only a diverse set of questions and answers means that the representations learned by the model are closer to the true distribution of the data.

We hypothesize that fine-tuning Bi-encoder$_{mpnet-base}$ (3e) for MANtIS (4a) is harmful because the intermediate data contains multiple Stack Exchange responses. In this way, the subset of dialogues of Stack Exchange that MANtIS encompasses might be serving only to overfit the intermediate representations. As evidence for this hypothesis, we found that (I) the learning curves flatten quickly (as opposed to other datasets) and (II) fine-tuning another language model that does not have Stack Exchange data (MSMarco) in their fine-tuning, Bi-encoder$_{bert-base}$ (3c), improves the effectiveness with statistical significance from 0.092 R@10 to 0.205 R@10, as shown in Table 3.3.

Do harder negative samples lead to more effective fine-tuning of dense models?

Surprisingly we found that using more effective models to select negative candidates is detrimental to the effectiveness of the dense retrieval model (rows 4a–c). We observe this phenomenon when using different language models and whether using intermediate training or not for all datasets tested, as shown in Table 3.3. We performed an experiment with an alternative contrastive loss [125] that employs in-batch negative sampling, and we observe that the same behaviour regardless of the loss function[11].

Based on brainstorming sessions and discussions the authors of the paper that originated this chapter had with other IR researchers a set of hypotheses was formed that could explain why this phenomenon might be happening. Next, we explore the three resulting hypotheses with six additional experiments.

[11] Other loss functions were also tested and resulted in the same effectiveness for the negative samplers: Random >> BM25 >> Bi-encoder.

3.5.3 Dense Retrieval: Negative Sampling

We investigated the following hypotheses that could explain the observed phenomena of decreasing effectiveness for better negative sampling functions:

H1: False negative samples increase when using better negative sampling methods. False negatives are responses that are potentially valid for the context. Such relevant responses sampled will lead to unlearning relevant matches between context and responses as they receive negative labels. Example retrieved by the Bi-encoder model (line 3e of Table 3.1):

> **Dialogue context ($\mathcal{U}$)**: hey... how long until dapper comes out? `[U]` 14 days [...] `[U]` i thought it was coming out tonight
>
> **Correct response (r^{+})**: just kidding couple hours
>
> **Negative sample (r^{-})**: there is a possibility dapper will be delayed [...] meanwhile, dapper discussions should occur in ubuntu+1

H2: Confusing negative samples increases when using better negative sampling methods. They are not relevant, i.e. a valid response to the context, but they are semantically or lexically identical to (or exact matches or part of) the context. Such samples will lead to representations of similar sentences being far apart in the embedding space. Example of a partial match retrieved by BM25 (line 1a of Table 3.1):

> **Dialogue context ($\mathcal{U}$)**: can any one help me im trying to install some thing and i get this error GTK... configure: error: Package requirements (gtk+-2.0 `[U]` perhaps sudo apt-get install libgtk2.0-dev `[U]` any way to tell it to install all dependencies too
>
> **Correct response (r^{+})**: what do you mean, it won't compile if you don't have the dependencies
>
> **Negative sample (r^{-})**: sudo apt-get install libgtk2.0-dev

H3: There is a lack of informative negative samples, i.e. responses that are more informative than random negative responses for training, for the dialogue contexts in each dataset. Informative negative samples are ideally the ones that (I) have lexical matches with the dialogue context[12] and are not semantically relevant or (II) give the impression that it is a natural and fluent response to the last utterance of the dialogue context but are not semantically relevant. Examples of potentially informative negative samples[13]:

[12] Unlike H2, they are not subsets of continuous parts of the context.

[13] The negative from a different collection was selected from *reddit/r/onedrive* dialogues. The generated negative sample was made using *DialoGPT-large* for the dialogue context.

Dialogue context ($\mathscr{U}$): I had my iPhone swapped out by Apple and after reinstalling my apps, signing in, etc, I noticed my OneDrive app was saying ”Be sure you’re connected to cellular or wifi”.... and it is. I’ve signed out and back in... removed and re added the app... etc no dice. Anyone have any suggestions?

Correct response (r^+): Hi, I realized the inconvenience you are experiencing. Is the issue specific to OneDrive app or with other apps as well? First, update iOS on your device. Then, make sure you’ve installed any available updates to the app. [...]

Different collection negative sample (r^-): I love OneDrive, have used it for years with no issues. I believe a lot of people have issues because they don’t understand how it works, they don’t read the instructions [...].

Generated negative sample (r^-): I had the same problem. I had to uninstall and reinstall the app.

In order to test our hypotheses we perform the following experiments, each one geared towards investigating one hypothesis:

E1: Annotate the relevance of a subset of negative samples to check whether the number of false negatives increases with better negative sampling functions (**H1**).

E2: Instead of using the top-ranked responses as negative responses, we use the bottom responses of the top-ranked responses as negatives[14]. This decreases the chances of obtaining false positives and if k is small such as 100, it will not render the sampling procedure to random (**H1**).

E3: Remove negative samples that are subsets of the context when training dense models and compare their effectiveness with the original negative samples (**H2**).

E4: Use only the last utterance to retrieve negative samples, this will make it less likely that a response is an exact match with the entire dialogue context (**H2**).

E5: Compare the effectiveness of models when using a corpus of responses for negative sampling which has additional responses from external corpora, that are potentially more informative than the ones from the original dataset (**H3**).

E6: Generate negative samples using a generative language model and compare the effectiveness of this model against using retrieved negative samples (**H3**).

Our findings for the six experiments (E1–E6) are displayed in Table 3.4. Bold values indicate positive evidence for their respective hypothesis. **In the first experiment (E1)**, we manually annotated the relevance of 270 pairs of dialogue context and negative samples (3 datasets × 3 dialogue contexts × 10 negative samples × 3 negative sampling method). We found that indeed the number of false positives increases when using better negative sampling approaches, providing positive evidence for the hypothesis that false positives are detrimental to the training of the dense retrieval models. **For the second experiment (E2)** we employ a denoising technique that uses the bottom negative samples from the top-k list instead of the first. We found that the effectiveness improves by large margins when

[14] As an example, when we retrieve $k = 100$ responses, instead of using responses ranked 1 to 10 we use responses ranked 91 to 100.

Table 3.4: Experiments to examine why better negatives sampling procedures lead to worse dense retrieval results. Bold indicates positive evidence for the corresponding hypothesis. We present the R@1 and R@10 for the condition presented and the absence of the condition for E2–E5.

E1

Negative Sampler		MANtIS	MSDialog	UDC-DSTC8
Random (0)		0	0	0
BM25 (1a)	false r^- count	0	4	2
Bi-encoder (3e)		11	4	15

E2

		MANtIS		MSDialog		UDC-DSTC8	
Negative Sampler	**Condition**	**R@1**	**R@10**	**R@1**	**R@10**	**R@1**	**R@10**
Random (0)		0.130	0.307	0.168	0.387	0.050	0.128
BM25 (1a)	no denoising	**0.112**	**0.271**	0.128	0.316	0.027	0.087
	denoising	0.101	0.257	**0.151**	**0.358**	**0.041**	**0.121**
Bi-encoder (3e)	no denoising	0.065	0.146	0.144	0.306	0.018	0.051
	denoising	**0.146**	**0.316**	**0.184**	**0.397**	**0.042**	**0.106**

E3

		MANtIS		MSDialog		UDC-DSTC8	
Negative Sampler	**Condition**	**R@1**	**R@10**	**R@1**	**R@10**	**R@1**	**R@10**
BM25 (1a)	r^- not subset of $\mathcal{U}$	**0.112**	**0.271**	0.128	0.316	**0.027**	**0.087**
	r^- subset of $\mathcal{U}$	0.095	0.239	**0.138**	**0.331**	0.025	0.077
Bi-encoder (3e)	r^- not subset of $\mathcal{U}$	0.065	0.146	**0.144**	**0.306**	**0.018**	**0.051**
	r^- subset of $\mathcal{U}$	**0.078**	**0.180**	0.127	0.266	0.015	0.047

E4

		MANtIS		MSDialog		UDC-DSTC8	
Negative Sampler	**Condition**	**R@1**	**R@10**	**R@1**	**R@10**	**R@1**	**R@10**
BM25 (1a)	$\mathcal{U}$ to retrieve candidate	0.112	**0.271**	0.128	0.316	0.027	**0.087**
	$\mathcal{U}_{lu}$ to retrieve candidate	**0.123**	0.270	**0.160**	**0.360**	**0.030**	0.078
Bi-encoder (3e)	$\mathcal{U}$ to retrieve candidate	0.065	0.146	0.144	0.306	0.018	0.051
	$\mathcal{U}_{lu}$ to retrieve candidate	**0.146**	**0.319**	**0.151**	**0.348**	**0.040**	**0.098**

E5

		MANtIS		MSDialog		UDC-DSTC8	
Negative Sampler	**Corpus to retrieve**	**R@1**	**R@10**	**R@1**	**R@10**	**R@1**	**R@10**
Random (0)	target only	0.130	0.307	**0.168**	**0.387**	**0.050**	**0.128**
	expanded	**0.136**	**0.312**	0.150	0.361	0.046	0.122
BM25 (1a)	target only	**0.112**	**0.271**	0.128	0.316	0.027	0.087
	expanded	0.104	0.257	**0.140**	**0.347**	**0.035**	**0.110**
Bi-encoder (3e)	target only	0.065	0.146	0.144	0.306	0.018	0.051
	expanded	**0.110**	**0.259**	**0.172**	**0.364**	**0.035**	**0.101**

E6

	MANtIS		MSDialog		UDC-DSTC8	
Negative Sampler	**R@1**	**R@10**	**R@1**	**R@10**	**R@1**	**R@10**
Random (0)	**0.130**	**0.307**	**0.168**	**0.387**	**0.050**	0.128
BM25 (1a)	0.112	0.271	0.128	0.316	0.027	0.087
Bi-encoder (3e)	0.065	0.146	0.144	0.306	0.018	0.051
GenNegatives$_{blenderbot-400M-distill}$	0.109	0.267	0.142	0.348	0.050	**0.134**
GenNegatives$_{DialoGPT-large}$	0.103	0.260	0.154	0.363	0.046	0.123

using the dense model to find negatives in all three datasets. In two datasets (MANtIS and MSDialog) we find that the denoised negative sampling of the Bi-encoder yields statistically significant improvements over Random (0.316 R@10 vs 0.307 R@10 for MANtIS and 0.397 R@10 vs 0.387 for MSDialog). The results for the second experiment are thus additional positive evidence for the hypothesis that false positives are detrimental.

In the third experiment (E3), by allowing the negative samples to be subsets of the dialogue context, we expect the effectiveness of the model to drop by large margins since the number of confusing negative samples increases. This was not the case. The results indicate that possibly confusing negative samples with exact matches with the dialogue context was not detrimental. **For the fourth experiment (E4)** we expected that when using only the last utterance of the dialogue to find negatives, we would decrease the number of confusing negatives. This was the case for training the model with the bi-encoder as the negative sampler.

In the final two experiments, we tested whether we could find more informative samples **by using an expanded corpus of responses (E5) and by using generated negative responses (E6)**. We found that using the larger corpus was beneficial when using the bi-encoder negative sampler, showing that we can possibly find more informative negative samples when using larger data. We found however that the generated negative responses from both models were not effective, as random samples from the corpus lead to better effectiveness when training the dense retrieval model.

Overall we see that we have the most evidence for the first hypothesis (H1) of false negatives degrading the training procedure. The problems of false negatives when using harder negatives has been discussed before for other retrieval tasks [104, 271], and we find evidence here on the conversational task that matches prior works on denoising the hard negatives. Other hypotheses (H2 and H3) had partial positive evidence, which suggests that they could also be a potential source of difficulty when training dense models with harder negatives. In conclusion, we demonstrate that a denoising strategy to remove false negative samples is required to train dense models for ranking responses for conversations when taking into account hard negative samples.

3.6 Limitations

One of the limitations of our study is that recent and more complex techniques that improve supervised sparse retrieval were not considered. doc2query does term re-weighting and expansion of the documents, but it does not modify the queries. Approaches that perform weighting and expansion for both the queries and documents [94]—predicting the weights of every token in the vocabulary regardless if they appear in the inputs or not—might be able to achieve better performance and close the gap or even surpass dense retrieval models in our domain.

3.7 Conclusions

We explored sparse and dense techniques that retrieve responses out of the entire collection available—in contrast to most prior work in response ranking for dialogues which are typically set up as a re-ranking task. The expansion of responses, i.e. $resp2ctxt_{lu}$ showed to be a strong baseline for sparse retrieval. We also find that dense retrieval needs large datasets in order to beat a strong sparse retrieval baseline in the zero-shot setting.

Our findings also suggest that fine-tuning a bi-encoder dense retrieval model after intermediate training is the best-performing method for the task of full-rank retrieval of responses for dialogues. We finish our experiments with a thorough analysis of negative sampling methods, exploring different hypotheses that could explain why harder negatives lead to worse effectiveness for the dense methods.

This chapter answers our first main research question of the thesis (M-RQ1), showing that a bi-encoder model is a strong baseline for the retrieval of responses for conversational search. We showed that most findings from other tasks such as passage retrieval translate to the retrieval of responses for dialogues. In terms of the multi-stage pipeline described in Figure 1.6, we focused in this chapter on first-stage approaches for conversational search. For the next two chapters, we move to the second main research question (M-RQ2) of the thesis and focus on the second-stage re-ranking step.

4

Difficulty Notions when Training Response Re-rankers

In this chapter, we focus on the second stage of the multi-stage pipeline for conversational search and explore how different notions of difficulty can improve re-rankers. In order to do so we rely on curriculum learning. This technique can be used to improve neural models' effectiveness by sampling batches non-uniformly, going from easy to difficult instances during training. In the context of neural information retrieval curriculum learning has not been explored yet, and so it remains unclear (1) how to measure the difficulty of training instances and (2) how to transition from easy to difficult instances during training. In order to deal with challenge (1), we explore scoring functions to measure the difficulty of conversations based on different input spaces. To address challenge (2) we evaluate different pacing functions, which determine the velocity at which we go from easy to difficult instances. We find that, overall, by just intelligently sorting the training data (i.e., by performing curriculum learning) we can improve the retrieval effectiveness by up to 2%. The code required to reproduce this chapter is available at `https://github.com/Guzpenha/transformers_cl`.

This chapter is based on the following paper:

🗎 *Gustavo Penha and Claudia Hauff. 2020. Curriculum Learning Strategies for IR. In ECIR. Springer, 699–713* [249].

4.1 Introduction

Curriculum Learning (CL) is motivated by the way humans teach complex concepts: teachers impose a certain order of the material during students' education. Following this guidance, students can exploit previously learned concepts to learn new ones. This idea was initially applied to machine learning over two decades ago [87] as an attempt to use a similar strategy in the training of a recurrent network by *starting small* and gradually learning more difficult examples. More recently, Bengio et al. [31] provided additional evidence that curriculum strategies can benefit neural network training with experimental results on different tasks such as shape recognition and language modeling. Since then, empirical successes were observed for several computer vision [124, 357] and natural language processing (NLP) tasks [295, 313, 396].

In supervised machine learning, a function is learned by the learning algorithm (the *student*) based on inputs and labels provided by the *teacher*. The teacher typically samples randomly from the entire training set. In contrast, CL imposes a structure on the training set based on a notion of difficulty of instances, presenting to the student easy instances before difficult ones. When defining a CL strategy we face two challenges that are specific to the domain and task at hand [124]: (1) arranging the training instances by a sensible measure of *difficulty*, and, (2) determining the *pace* in which to present instances—going over easy instances too fast or too slow might lead to ineffective learning.

We conduct here an empirical investigation into those two challenges in the context of IR. Estimating relevance—a notion based on human cognitive processes—is a complex and difficult task at the core of IR, and it is still unknown *to what extent CL strategies are beneficial for neural ranking models*. This is the question we aim to answer in our work.

Given a set of queries—for instance user utterances, search queries, or questions in natural language—and a set of documents—for instance responses, web documents, or passages—neural ranking models learn to distinguish relevant from non-relevant query-document pairs by training on a large number of labeled training pairs. Neural models had for some time struggled to display significant and additive gains in IR [375]. In a short time though, BERT [80] (released in late 2018) and its derivatives (e.g. XLNet [379], RoBERTa [201]) have proven to be remarkably effective for a range of NLP tasks. The recent breakthroughs of these large and heavily pre-trained language models have also benefited IR [376, 377, 381].

In our work we focus on the challenging IR task of conversation response ranking [367], where the query is the dialogue history and the documents are the candidate responses of the agent. The set of responses is not generated on the go, they must be retrieved from a comprehensive dialogue corpus. A number of deep neural ranking models have recently been proposed for this task [320, 367, 374, 398, 402], which is more complex than retrieval for single-turn interactions, as the ranking model has to determine where the important information is in the previous user utterances and how it is relevant to the current information need of the user. Due to the complexity of the relevance estimation problem displayed in this task, we argue it to be a good test case for curriculum learning in IR.

In order to tackle the first challenge of CL (determine what makes an instance difficult) we contribute different *scoring functions* that determine the difficulty of query-document pairs based on four different input spaces: conversation history $\{\mathcal{U}\}$, candidate responses $\{\mathcal{R}\}$, both $\{\mathcal{U},\mathcal{R}\}$, and $\{\mathcal{U}, \mathcal{R}, \mathcal{Y}\}$, where $\mathcal{Y}$ are relevance labels for the responses. To address the second challenge (determine the pace to move from easy to difficult instances) we contribute different *pacing functions* that serve easy instances to the learner for more or less time during the training procedure. We empirically explore how the curriculum strategies perform for two different response ranking datasets when compared against vanilla (no curriculum) fine-tuning of BERT for the task. Our main findings are that (i) CL improves retrieval effectiveness when we use difficulty criteria based on a supervised model that uses all the available information $\{\mathcal{U}, \mathcal{R}, \mathcal{Y}\}$, (ii) it is best to give the model more time to assimilate harder instances during training by introducing difficult instances in earlier iterations, and, (iii) the CL gains over the no CL baseline are spread over different conversation domains, lengths of conversations and measures of conversation difficulty.

4.2 Related Work

In this section, we first review neural ranking models followed by curriculum learning approaches in diverse fields.

4.2.1 Neural Ranking Models

Over the past few years, the IR community has seen a great uptake of the many flavors of deep learning for all kinds of IR tasks such as ad-hoc retrieval, question answering, and conversation response ranking. Unlike traditional learning to rank (LTR) [200] approaches in which we manually define features for queries, documents and their interaction, neural ranking models learn features directly from the raw textual data. Neural ranking approaches can be roughly categorized into representation-focused [138, 301, 349] and interaction-focused [120, 350]. The former learn query and document representations separately and then computes the similarity between the representations. In the latter approach, first, a query-document interaction matrix is built, which is then fed to neural net layers. Estimating relevance directly based on interactions, i.e. interaction-focused models, has shown to outperform representation-based approaches on several tasks [137, 235].

Transfer learning via large pre-trained Transformers [343]—the prominent case being BERT [80]—has led to remarkable empirical successes on a range of NLP problems. The BERT approach to learning textual representations has also significantly improved the performance of neural models for several IR tasks [270, 297, 376, 377, 381], that for a long time struggled to outperform classic IR models [375]. In this work, we use the no-CL BERT as a strong baseline for the conversation response ranking task.

4.2.2 Curriculum Learning

Following a curriculum that dictates the ordering and content of the educational material is prevalent in the context of human learning. With such guidance, students can exploit previously learned concepts to ease the learning of new and more complex ones. Inspired by cognitive science research [293], researchers posed the question of whether a machine learning algorithm could benefit, in terms of learning speed and effectiveness, from a similar curriculum strategy [31, 87]. Since then, positive evidence for the benefits of curriculum training, i.e. training the model using easy instances first and increasing the difficulty during the training procedure, has been empirically demonstrated in different machine learning problems, e.g. image classification [110, 124], machine translation [171, 260, 396] and answer generation [199].

Processing training instances in a meaningful order is not unique to CL. Another related branch of research focuses on *dynamic* sampling strategies [42, 53, 174, 302], which, unlike CL that requires a definition of what is easy and difficult before training starts, estimates the importance of instances during the training procedure. Self-paced learning [174] simultaneously selects easy instances to focus on and updates the model parameters by solving a biconvex optimization problem. A seemingly contradictory set of approaches give more focus to difficult or more uncertain instances. In active learning [53, 65, 331], the most uncertain instances with respect to the current classifier are employed for training. Similarly, hard example mining [302] focuses on difficult instances, measured by the model loss or magnitude of gradients for instance. Boosting [42, 394] techniques give more weight to difficult instances as training progresses. In this work, we focus on CL, which has been more successful in neural models, and leave the study of dynamic sampling strategies in neural IR as future work.

The most critical part of using a CL strategy is defining the difficulty metric to sort instances by. The estimation of instance difficulty is often based on our prior knowledge of what makes each instance difficult for a certain task and thus is domain-dependent (cf. Table 4.1 for curriculum examples). CL strategies have not been studied yet in neural ranking models. To our knowledge, CL has only been employed in IR within the LTR framework, using LambdaMart [47], for ad-hoc retrieval by Ferro et al. [93]. However, no effectiveness improvements over randomly sampling training data were observed. The representation of the query, document, and their interactions in the traditional LTR framework is dictated by the manually engineered input features. We argue that neural ranking models, which learn how to represent the input, are better suited for applying CL in order to learn increasingly more complex concepts.

Table 4.1: Difficulty measures used in the curriculum learning literature.

Difficulty criteria	**Tasks**
Sentence length	machine translation [260], language generation [313], reading comprehension [383]
Word rarity	machine translation [260, 396], language modeling [31]
External model confidence	machine translation [396], image classification [124, 357], ad-hoc retrieval [93]
Supervision signal intensity	facial expression recognition [116], ad-hoc retrieval [93]
Noise estimate	speaker identification [275], image classification [56]
Human annotation	image classification [335] (through weak supervision)

4.3 Curriculum Learning: Easy First Difficult Later

Before introducing our experimental framework (i.e., the scoring functions and the pacing functions we investigate), let us first formally introduce the specific IR task we explore—a choice dictated by the complex nature of the task (compared to e.g. ad-hoc retrieval) and the availability of large-scale training resources such as MSDialog [268].

4.3.1 Problem Definition: Re-ranking

This is the typical conversation response ranking problem as defined in Section 2.6.1, with a small set of candidate responses, i.e. re-ranking step.

4.3.2 Framework

When training neural networks, the common training procedure is to divide the dataset $\mathcal{D}$ into $\mathcal{D}_{train}, \mathcal{D}_{dev}, \mathcal{D}_{test}$ and randomly (i.e., uniformly—every sample has the same likelihood of being sampled) sample mini-batches $\mathcal{B} = \{(\mathcal{U}_i, \mathcal{R}_i, \mathcal{Y}_i)\}_{i=1}^{b}$ of b instances from $\mathcal{D}_{train}$ where b is way smaller than the collection size, and perform an optimization procedure sequentially in $\{\mathcal{B}_1, ..., \mathcal{B}_B\}$. The CL framework employed here is inspired by previous works [260, 357]. It is defined by two functions: the *scoring function* which determines the difficulty of instances and the *pacing function* which controls the pace with which to transition from easy to hard instances during training. More specifically, the scoring function $f_{score}(\mathcal{U}_i, \mathcal{R}_i, \mathcal{Y}_i)$, is used to sort the training dataset. The pacing function $f_{pace}(s)$ determines the percentage of the sorted dataset available for sampling according to the current training step s (one forward pass plus one backward pass of a batch is considered to be one step). The neural ranking model samples uniformly from the initial $f_{pace}(s) * |D_{train}|$ instances sorted by f_{score}, while the rest of the dataset is not available for sampling. During training $f_{pace}(s)$ goes from δ (percentage of initial training data) to 1 when $s = T$. Both δ and T are hyperparameters. We provide an illustration of the process in Figure 4.1.

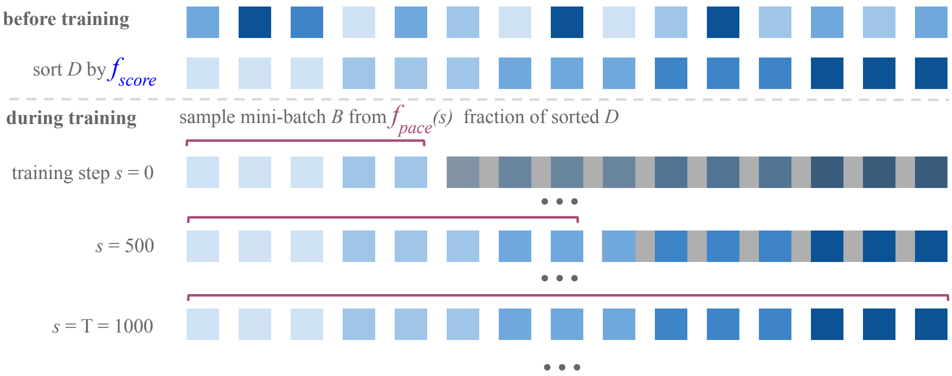


Figure 4.1: Our curriculum learning framework is defined by two functions. The scoring function $f_{score}(instance)$ defines the instances' difficulty (darker/lighter blue indicate higher/lower difficulty). The pacing function $f_{pace}(s)$ indicates the percentage of the dataset available for sampling according to the training step s.

Table 4.2: Overview of our curriculum learning scoring functions.

Input Space	**Name**	**Definition**	**Difficulty notion**
baseline	*random*	$f_{score} = Uniform(0,1)$	
$(\mathcal{U})$	$\#_{turns}$ $\overline{\#_{\mathcal{U}\, words}}$	$f_{score}(\mathcal{U}) = \lvert\mathcal{U}\rvert$ $f_{score}(\mathcal{U}) = \frac{\sum_{i=0}^{\lvert\mathcal{U}\rvert} word\_count(u_i)}{\lvert\mathcal{U}\rvert}$	information spread
$(\mathcal{R})$	$\overline{\#_{\mathcal{R}\, words}}$	$f_{score}(\mathcal{R}) = \frac{\sum_{i=0}^{\lvert\mathcal{R}\rvert} word\_count(r_i)}{\lvert\mathcal{R}\rvert}$	distraction in responses
$(\mathcal{U},\mathcal{R})$	σ_{SM} σ_{BM25}	$f_{score}(\mathcal{U},\mathcal{R}) = \sqrt{\frac{\sum_{i=0}^{\lvert\mathcal{R}\rvert}(SM(\mathcal{U},r_i) - \overline{SM(\mathcal{U},\mathcal{R})})^2}{\lvert\mathcal{R}\rvert - 1}}$ $f_{score}(\mathcal{U},\mathcal{R}) = \sqrt{\frac{\sum_{i=0}^{\lvert\mathcal{R}\rvert}(BM25(\mathcal{U},r_i) - \overline{BM25(\mathcal{U},\mathcal{R})})^2}{\lvert\mathcal{R}\rvert - 1}}$	responses heterogeneity
$(\mathcal{U},\mathcal{R},\mathcal{Y})$	$BERT_{pred}$ $\overline{BERT_{loss}}$	$f_{score}(\mathcal{U},\mathcal{R},\mathcal{Y}) =$ $-(BERT\_pred(\mathcal{U},r_i^+) - BERT\_pred(\mathcal{U},r_i^-))$ $f_{score}(\mathcal{U},\mathcal{R},\mathcal{Y}) = \frac{\sum_{i=0}^{\lvert\mathcal{R}\rvert} BERT\_loss(\mathcal{U},r_i)}{\lvert\mathcal{R}\rvert}$	model confidence

4.3.3 Scoring Functions

In order to measure the difficulty of a training triplet composed of $(\mathcal{U}_i, \mathcal{R}_i, \mathcal{Y}_i)$, we define scoring functions that use different parts of the input space: functions that leverage (i) the text in the dialogue history $\{\mathcal{U}\}$ (ii) the text in the response candidates $\{\mathcal{R}\}$ (iii) interactions between them, i.e., $\{\mathcal{U},\mathcal{R}\}$, and, (iv) all available information including the labels for the training set, i.e., $\{\mathcal{U},\mathcal{R},\mathcal{Y}\}$. The seven[1] scoring functions we propose are defined in Table 4.2; we now provide intuitions of why we believe each function to capture some notion of instance difficulty.

- $\#_{turns}(\mathcal{U})$ and $\overline{\#_{\mathcal{U}\, words}}(\mathcal{U})$: The important information in the context can be spread over different utterances and words. Bigger dialogue contexts mean there are more places where the important part of the user information need can be spread over. $\overline{\#_{\mathcal{R}\, words}}(\mathcal{R})$: Longer responses can distract the model as to which set of words or sentences are more important for matching. Previous work shows that it is possible to fool machine reading models by creating longer documents with additional distracting sentences [148].

- $\sigma_{SM}(\mathcal{U},\mathcal{R})$ and $\sigma_{BM25}(\mathcal{U},\mathcal{R})$: Inspired by query performance prediction [303], we use the variance of retrieval scores to estimate the amount of heterogeneity of information, i.e. diversity, in the response candidate. Homogeneous ranked lists are considered to be easy. We deploy a semantic matching model (SM) and BM25 to capture both semantic correspondences and keyword matching [276]. SM is the average cosine similarity between the first k words from $\mathcal{U}$ (concatenated utterances) with the first k words from r using pre-trained word embeddings.

[1] The function *random* is the baseline—instances are sampled uniformly (no CL).

- $BERT_{pred}(\mathcal{U},\mathcal{R},\mathcal{Y})$ and $\overline{BERT_{loss}}(\mathcal{U},\mathcal{R},\mathcal{Y})$: Inspired by CL literature [124], we use external model prediction confidence scores as a measure of difficulty[2]. We fine-tune BERT [80] on $\mathcal{D}_{train}$ for the conversation response ranking task. For $BERT_{pred}$ easy dialogue contexts are the ones that the BERT confidence score for the positive response r^{+} candidate is higher than the confidence for the negative response candidate r^{-}. The higher the difference the easier the instance is. For $\overline{BERT_{loss}}$we consider the loss of the model to be an indicator of the difficulty of an instance.

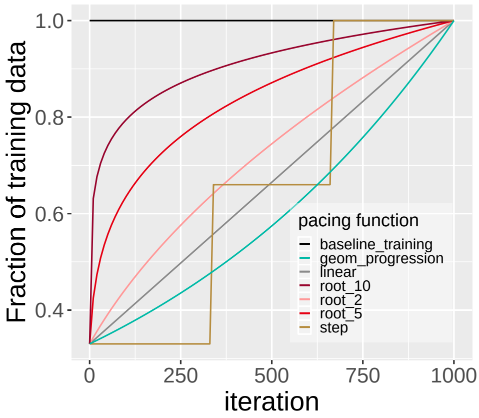


Figure 4.2: Example of pacing functions with $\delta = 0.33$ (fraction of data used at the beginning of training) and $T = 1000$ (total of iterations).

4.3.4 Pacing Functions

Assuming that we know the difficulty of each instance in our training set, we still need to define how are we going to transition from easy to hard instances. We use the concept of pacing functions $f_{pace}(s)$; they should each have the following properties [260, 357]: (i) start at an initial value of training instances $f_{pace}(0) = \delta$ with $\delta > 0$, so that the model has a number of instances to train in the first iteration, (ii) be non-decreasing, so that harder instances are added to the training set, and, (iii) eventually all instances are available for sampling when it reaches T iterations, $f_{pace}(T) = 1$.

As intuitively visible in Figure 4.2, we opted for pacing functions that introduce more difficult instances at different paces—while *root*_10 introduces difficult instances very early (after 125 iterations, 80% of all training data is available), *geom_progression* introduces them very late (80% is available after ~ 800 iterations). We consider four different types of pacing functions, formally defined in Table 4.3. The *step* function [31, 124, 310] divides the data into S fixed-sized groups, and after $\frac{T}{S}$ iterations a new group of instances is added, where S is a hyperparameter. A more gradual transition was proposed by Platanios et al. [260], by adding a percentage of the training dataset linearly with respect to the total of CL iterations T, and thus the slope of the function is $\frac{1-\delta}{T}$ (*linear* function). They also

[2] We note, that using BM25 average precision as a scoring function failed to outperform the baseline.

Table 4.3: Overview of our curriculum learning pacing functions. δ and T are hyperparameters.

Pacing function	Definition
baseline_training	$f_{pace}(s) = 1$
step	$f_{pace}(s) = \begin{cases} \delta, & \text{if } s \leq T * 0.33 \\ 0.66, & \text{if } s > T * 0.33, s \leq T * 0.66 \\ 1, & \text{if } s > T * 0.66 \end{cases}$
root	$f_{pace}(s, n) = min\left(1, \left(s\frac{1-\delta^n}{T} + \delta^n\right)^{\frac{1}{n}}\right)$
linear	$f_{pace}(s, n) = root(s, 1)$
root_n	$f_{pace}(s, n) = root(s, n)$
geom_progression	$f_{pace}(s) = min\left(1, 2^{\left(s\frac{log_2 1 - log_2 \delta}{T} + log_2 \delta\right)}\right)$

proposed $root\_n$ functions motivated by the fact that difficult instances will be sampled less as the training data grows in size during training. By making the slope inversely proportional to the current training data size, the model has more time to assimilate difficult instances. Finally, we propose the use of a geometric progression that instead of quickly adding difficult examples, gives easier instances more training time.

4.4 Experimental Setup

In order to test curriculum learning approaches we consider two[3] large-scale information-seeking conversation datasets introduced in Section 2.7.1: `MANtIS` and `MSDialog`.

4.4.1 Implementation Details

As a strong neural ranking model for our experiments, we employ BERT [80] for the conversational response ranking task. We follow recent research in IR that employed fine-tuned BERT for retrieval tasks [236, 377] and obtain strong baseline (i.e., no CL) results for our task. The best model by Yang et al. [374], which relies on external knowledge sources for `MSDialog`, achieves a MAP of 0.68 whereas our BERT baselines reach a MAP of 0.71 (cf. Table 4.4). We fine-tune BERT[4] for sentence classification, using the `[CLS]` token[5]; the input is the concatenation of the dialogue context and the candidate response separated by `[SEP]` tokens. When training BERT we employ a balanced number of relevant and non-relevant context and response pairs[6]. We use cross entropy loss and the Adam optimizer [168] with a learning rate of $5e-5$ and $\epsilon = 1e-8$, the default hyperparameters.

For σ_{SM}, as word embeddings, we use pre-trained fastText[7] embeddings with 300 di-

[3] The experiments of this chapter were performed when the `UDC-DSTC8` dataset was not yet released.

[4] We use the PyTorch-Transformers implementation `https://github.com/huggingface/pytorch-transformers` and resort to *bert-base-uncased* with default settings.

[5] The BERT authors suggest `[CLS]` as a starting point for sentence classification tasks [80].

[6] We observed similar results to training with a 1 to 10 ratio in initial experiments.

[7] `https://fasttext.cc/docs/en/crawl-vectors.html`

mensions and a maximum length of $k = 20$ words of dialogue contexts and responses. For σ_{BM25}, we use default values[8] of $k_1 = 1.5$, $b = 0.75$ and $\epsilon = 0.25$. For CL, we fix T as 90% percent of the total training iterations—this means that we continue training for the final 10% of iterations after introducing all samples—and the initial number of instances δ as 33% of the data to avoid sampling the same instances several times.

4.4.2 Evaluation

To compare our strategies with the baseline where no CL is employed, for each approach, we fine-tune BERT five times with different random seeds—to rule out that the results are observed only for certain random weight initialization values—and for each run, we select the model with best-observed effectiveness on the development set. The best model of each run is then applied to the test set. We report the effectiveness with respect to Mean Average Precision (MAP) like prior works [367, 374]. We perform paired Student's t-tests[9] between each scoring/pacing-function variant and the baseline run without CL.

4.5 Results

We first report the results for the pacing functions (Figure 4.3) followed by the main results (Table 4.4) comparing different scoring functions. We finish with an error analysis to understand when CL outperforms our no-curriculum baseline.

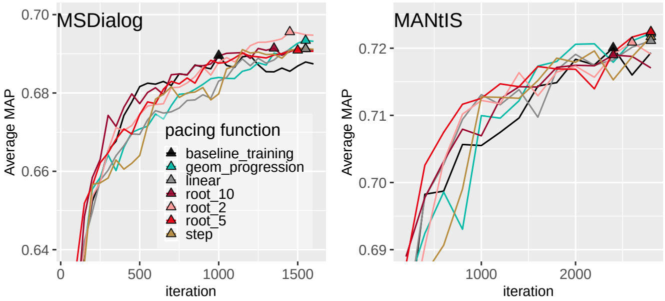


Figure 4.3: Average development MAP for 5 different runs, using different curriculum learning pacing functions. △ is the maximum observed MAP. On the left, we have results for the `MSDialog` dataset, and on the right for the `MANtIS` dataset.

4.5.1 Pacing Functions

In order to understand how CL results are impacted by the pace we go from easy to hard instances, we evaluate the different proposed *pacing functions*. We display the evolution of the development set MAP (average of 5 runs) during training in Figure 4.3 (we use development MAP to track effectiveness during training). We fix the scoring function as $BERT_{pred}$; this is the best performing scoring function, more details in the next section. We see that the pacing functions with the maximum observed average MAP are *root_2*

[8] `https://radimrehurek.com/gensim/summarization/bm25.html`

[9] Unlike other chapters, we do not apply Bonferoni correction here due to having a single baseline (no CL).

and $root\_5$ for MSDialog and MANtIS respectively[10]. The other pacing functions, *linear*, *geom_progression* and *step*, also outperform the standard training baseline with statistical significance (using Student's t-test and confidence level of 99%) on the test set and yield similar results to the *root_2* and *root_5* functions.

Our results are aligned with previous research on CL [260], that giving more time for the model to assimilate harder instances (by using a root pacing function) is beneficial to the curriculum strategy and is better than no CL with statistical significance. For the rest of our experiments, we fix the pacing function as $root\_2$, the best pacing function for MSDialog. Let's now turn to the impact of the scoring functions.

4.5.2 Scoring Functions

Table 4.4: Test set MAP results of 5 runs using different curriculum learning scoring functions. Superscripts $^{\dagger}/^{\ddagger}$ denote statistically significant improvements over the baseline where no curriculum learning is applied ($f_{score} = random$) at 95%/99% confidence intervals. Bold indicates the highest MAP for each line.

Run	$random$	$\#_{turns}$	$\overline{\#_{\mathcal{U} words}}$	$\overline{\#_{\mathcal{R} words}}$	σ_{SM}	σ_{BM25}	$BERT_{pred}$	$\overline{BERT_{loss}}$
				MSDialog				
1	0.7142	0.7220 †	0.7229 †	0.7182	0.7239 †‡	0.7175	**0.7272** †‡	0.7244 †‡
2	0.7044	0.7060	0.7053	0.6968	0.7032	0.7003	**0.7159** †‡	0.7194 †‡
3	0.7126	0.7215 †	0.7163	0.7171	0.7174	0.7159	**0.7296** †‡	0.7225 †‡
4	0.7031	0.7065	0.7043	0.6993	0.7026	0.6949	0.7154 †‡	**0.7204** †‡
5	0.7148	0.7225 †	0.7203	0.7169	0.7171	0.7134	0.7322 †‡	**0.7331** †‡
AVG	0.7098	0.7157	0.7138	0.7097	0.7128	0.7084	**0.7241**	0.7240
SD	0.0056	0.0086	0.0086	0.0106	0.0095	0.0101	0.0079	0.0055
				MANtIS				
1	0.7203	0.7192	0.7198	0.7194	0.7166	0.7200	0.7257 †‡	**0.7268** †‡
2	0.6984	0.6993	0.6989	0.6996	0.6964	0.7009	**0.7067** †‡	0.7051 †‡
3	0.7200	0.7197	0.7134	0.7206	0.7153	0.7153	**0.7282** †‡	0.7221
4	0.7114	0.7117	0.7002	0.6978	0.7140	0.7084	**0.7240** †‡	0.7184 †‡
5	0.7156	0.7174	0.7193 †	0.7162	0.7147	0.7185	**0.7264** †‡	0.7258 †‡
AVG	0.7131	0.7135	0.7103	0.7107	0.7114	0.7126	**0.7222**	0.7196
SD	0.0090	0.0085	0.0102	0.0111	0.0084	0.0079	0.0088	0.0088

The most critical challenge of CL is defining a measure of the difficulty of instances. In order to evaluate the effectiveness of our scoring functions we report the test set results across both datasets in Table 4.4. We observe that the scoring functions which do not use the relevance labels $\mathcal{Y}$ are not able to outperform the no CL baseline (*random* scoring function). They are based on features of the dialogue context $\mathcal{U}$ and responses $\mathcal{R}$ that we hypothesized make them difficult for a model to learn. Differently, for $\overline{BERT_{loss}}$ and $BERT_{pred}$ we observe statistically significant results on both datasets across different runs. They differ in two ways from the unsuccessful scoring functions: they have access to the

[10] If we increase the n of the root function to bigger values, e.g. $root\_10$, the results drop and get closer to not using CL. This is due to the fact that higher n generate root functions with a similar shape to standard training, giving the same amount of time to easy and hard instances (cf. Figure 4.2).

training labels $\mathcal{Y}$ and the difficulty of an instance is based on what a previously trained model determines to be hard, and thus not our intuition.

Our results bear resemblance to Born Again Networks [97], where a student model which is identical in parameters and architecture to the teacher model outperforms the teacher when trained with knowledge distillation [133], i.e., using the predictions of the teacher model as labels for the student model. The difference here is that instead of transferring the knowledge from the teacher to the student through the labels, we transfer the knowledge by imposing a structure/order on the training set, i.e. curriculum learning.

4.5.3 Error Analysis

In order to understand when CL performs better than random training samples, we fix the scoring ($BERT_{pred}$) ad pacing function (*root_2*) and explore the test set effectiveness along several dimensions (Figures 4.4 and 4.5). We report the results only for MSDialog, but the trends hold for MANtIS as well.

We first consider the number of turns in the conversation in Figure 4.4. CL outperforms the baseline approach for the types of conversations appearing most frequently (2-5 turns in MSDialog). The CL-based and baseline effectiveness drops for conversations with a large number of turns. This can be attributed to two factors: (1) employing pre-trained BERT in practice allows only a certain maximum number of tokens as input, so longer conversations can lose important information due to truncating; (2) for longer conversations it is harder to identify the important information to match in the history.

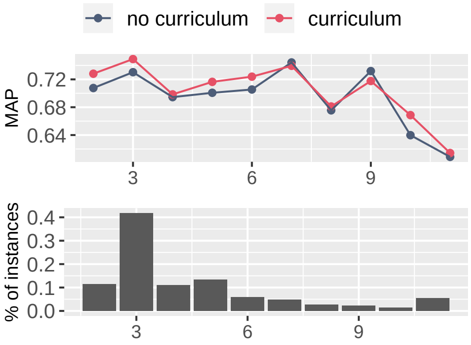


Figure 4.4: On the top we have the MSDialog test set MAP of curriculum learning and baseline (no curriculum) by number of turns. On the bottom, we have the number of instances per number of turns.

Next, we look at different conversation domains in Figure 4.5 (left), such as *windows10* and *word* for MSDialog—are the gains in effectiveness limited to particular domains? The error bars indicate the confidence intervals with a confidence level of 95%. We list only the most common domains in the test set. The gains of CL are spread over different domains as opposed to concentrated on a single domain.

Lastly, using our scoring functions we sort the test instances and divide them into three buckets: first 33% instances, 33%–66%, and 66%–100%. In Figure 4.5 (right), we see the effectiveness of CL against the baseline for each bucket using $\overline{\#_{\mathcal{U}\,words}}$ (the same trend holds for the other scoring functions). As we expect, the bucket with the most difficult

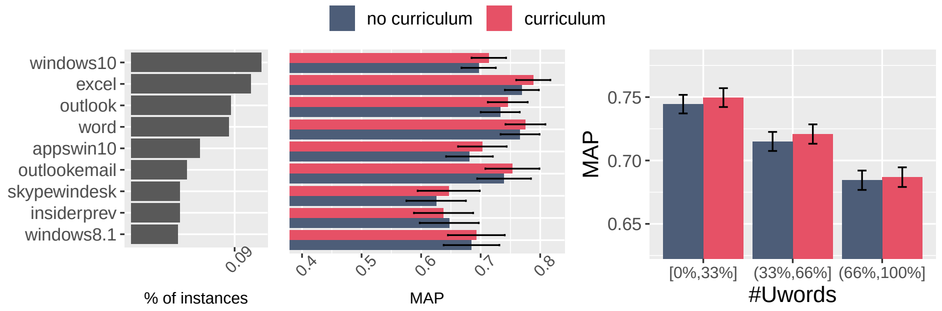


Figure 4.5: Test set MAP for `MSDialog` across different domains (left) and instances' difficulty (right) according to $\#_{\mathcal{R}words}$ for curriculum learning and the baseline.

instances according to the scoring function is the one with the lowest MAP values. Finally, the improvements of CL over the baseline are again spread across the buckets, showing that CL is able to improve over the baseline for different levels of difficulty.

4.6 Limitations

A limitation of our study is that we only consider a single BERT model for re-ranking. While the focus of this chapter is on the re-ranking step, the findings might also generalize to the retrieval step and other model architectures. For example, Zeng et al. [391] showed in a subsequent study[11] that a curriculum learning approach is effective for the first-stage retrieval step, by employing it to control the level of difficulty of the teacher supervision for a dense retriever. In the domain of conversational data, a dense retriever was also shown to benefit from curriculum learning in another subsequent study [218].

A second concern is that even though the method is simple to implement as it only changes the order of the training instances, the size of the effectiveness improvements we obtained was small. We believe that more sophisticated scoring functions and different ways of applying curriculum learning, e.g. through different tasks, might lead to higher effectiveness gains.

4.7 Conclusions

In this work, we studied whether CL strategies are beneficial for neural ranking models. We find supporting evidence for curriculum learning in IR. Simply reordering the instances in the training set using difficulty criteria leads to effectiveness improvements, requiring no changes to the model architecture—a similar relative improvement in MAP has justified novel neural architectures in the past [320, 367, 398, 402]. Our experimental results on two conversation response ranking datasets reveal (as one might expect) that it is best to use all available information ($\mathcal{U}, \mathcal{R}, \mathcal{Y}$) as evidence for instance difficulty.

This chapter provides evidence for the second research question of the thesis (M-RQ2), showing that different notions of the difficulty of a dialogue can be used to improve a re-

[11] Curriculum learning was also shown in 2022 [391] to be helpful for dense retrieval. This study was published after the paper [249] (2020) which originated this chapter.

ranking model for conversational search. We rely here specifically on a CL method, but other approaches could be used to take advantage of the difficulty estimations as proposed by the scoring functions. In terms of the multi-stage pipeline described in Figure 1.6, we focused in this chapter on second-stage approaches for conversational search, with a cross-encoder model that is more powerful but less efficient than the approaches outlined in the previous chapter. In the next chapter we continue to evaluate M-RQ2, still working with cross-encoder re-ranking models for the second stage of the pipeline. We take a different route to calculate and employ the difficulty of dialogues, relying on stochastic rankers and using the such model's uncertainty estimates.

5

Difficulty Notions when Predicting with Response Re-rankers

In this chapter, we continue our exploration of the second stage of the multi-stage pipeline for conversational search and turn our attention to difficult dialogues when predicting relevance. According to the Probability Ranking Principle (PRP), ranking responses in decreasing order of their probability of relevance leads to an optimal ranking. The PRP holds when two conditions are met: ***[C1]*** *the models are well calibrated, and,* ***[C2]*** *the probabilities of relevance are reported with certainty. We know however that deep neural networks (DNNs) are often not well calibrated and have several sources of uncertainty, and thus* ***[C1]*** *and* ***[C2]*** *might not be satisfied by neural rankers. Given the success of neural re-ranking models—and here, especially BERT-based cross-encoder approaches—we first analyze under which circumstances they are calibrated for conversational search problems. Then, motivated by our findings we use two techniques to model the uncertainty of neural rankers leading to the proposed stochastic rankers, which output a predictive distribution of relevance as opposed to point estimates. Our experimental results reveal that (i) BERT-based rankers are not robustly calibrated and that stochastic BERT-based rankers yield better calibration; and (ii) uncertainty estimation is beneficial for both risk-aware neural ranking, i.e. taking into account the uncertainty when ranking responses, and for predicting unanswerable conversational contexts. The code required to reproduce this chapter is available at* `https://github.com/Guzpenha/transformer_rankers/tree/uncertainty_estimation`.

This chapter is based on the following paper:

Gustavo Penha and Claudia Hauff. 2021. On the Calibration and Uncertainty of Neural Learning to Rank Models for Conversational Search. In EACL. 160–170 [251].

5.1 Introduction

According to the Probability Ranking Principle (PRP) [289], ranking documents in decreasing order of their probability of relevance leads to an optimal document ranking for ad-hoc retrieval[1]. Gordon and Lenk [112] discussed that for the PRP to hold, a ranking model must at least meet the following conditions: **[C1]** assign well-calibrated probabilities of relevance, i.e. if we gather all documents for which the model predicts relevance with a probability of e.g. 30%, the number of relevant documents should be 30%, and **[C2]** report certain predictions, i.e. only point estimates, for example, 80% probability of relevance.

DNNs have been shown to outperform classic information retrieval ranking models over the past few years in setups where considerable training data is available. It has been shown that DNNs are not well calibrated in the context of computer vision [119]. If the same is true for neural models for IR, e.g. transformer models for ranking [236], **[C1]** is not met. Additionally, there are a number of sources of uncertainty in the training process of neural networks [99] that make it unreasonable to assume that neural ranking models fulfill **[C2]**: *parameter uncertainty* (different combinations of weights that explain the data equally well), *structural uncertainty* (which neural architecture to use for neural ranking), and *aleatoric uncertainty* (noisy data). Given these sources of uncertainty, using point estimate predictions and ranking according to the PRP might not achieve the optimal ranking. While the effectiveness benefits of risk-aware models [352, 353], which take into account the risk[2], i.e. the uncertainty of the document's prediction scores, have been shown for non-neural IR approaches, the same was not explored for neural L2R models.

We first contribute an analysis of the calibration of neural rankers, specifically BERT-based rankers for IR tasks to understand how calibrated they are. Then, to model the uncertainty of BERT-based rankers, we contribute with *stochastic* neural ranking models (see Figure 5.1), by applying different techniques to model the uncertainty of DNNs, namely MC Dropout [100] and Deep Ensembles [177] which are agnostic to the particular DNN architecture. In our experiments, we test models under *distributional shift*, i.e. the test data distribution is different from the training data, also referred to as out-of-distribution (OOD) examples [181]. In real-world settings, there are often inputs that are shifted due to factors such as non-stationarity and sample bias. Additionally, this experimental setup provides a way of measuring whether the DNN *"knows what it knows"* [242], e.g. by outputting high uncertainty for OOD examples.

We find that BERT-based rankers are not robustly calibrated. Stochastic BERT-based rankers have 14% less calibration error on average than BERT-based rankers. Uncertainty estimation from stochastic BERT-based rankers is advantageous for downstream applications as shown by our experiments for risk-aware neural ranking (2% more effective on average relative to a model without risk-awareness) and for predicting unanswerable conversational contexts (improves classification by 33% on average of all conditions).

[1] Standard retrieval task where the user specifies his information need through a query which initiates a search by the system for documents that are likely relevant [19].

[2] We use risk and uncertainty interchangeably here.

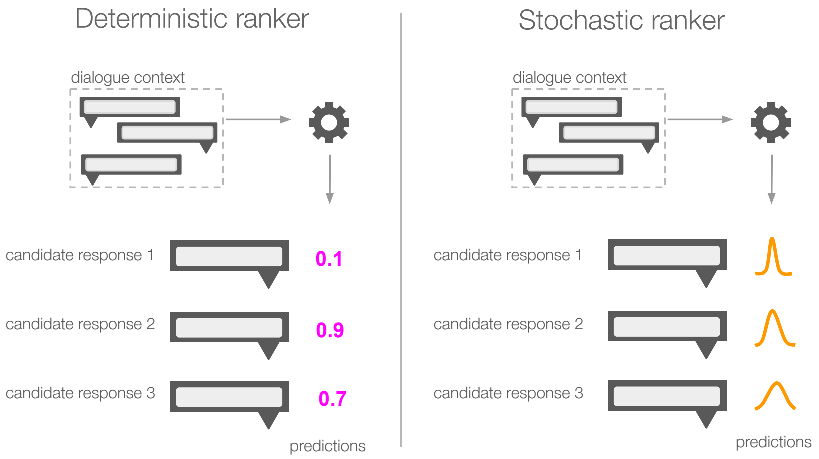


Figure 5.1: While deterministic neural rankers (left) output a point estimate probability (magenta values) of relevance for a combination of dialogue context and candidate response, stochastic neural rankers (right) output a predictive distribution (orange curves). The dispersion of the predictive distribution provides an estimation of the model uncertainty.

5.2 Related Work

In this section we first analyze previous efforts in the topics of calibration and uncertainty within information retrieval, followed by the field of bayesian neural networks.

5.2.1 Calibration and Uncertainty in IR

Even though optimally ranking documents according to the PRP [289] requires the model to be calibrated [112] (**[C1]**), the calibration of ranking models has received little attention in IR. In contrast, in the machine learning community, there have been a number of studies about calibration [215, 242], due to the larger decision-making pipelines DNNs are often part of and their importance for model interpretability [327]. For instance, in the automated medical domain it is important to provide a calibrated confidence measure besides the prediction of a disease diagnosis to provide clinicians with sufficient information [150]. Guo et al. [119] have shown that DNNs are not well calibrated in the context of computer vision, motivating our study of the calibration of neural L2R models.

The second condition (**[C2]**) for optimal retrieval when ranking according to the PRP [112] is that models report predictions with certainty. While the (un)certainty has not been studied in neural L2R models, there are classic approaches in IR that model the uncertainty. Such approaches have been mostly inspired by economics theory, treating variance as a measure of uncertainty [342]. Following such ideas, non-neural ranking models that take uncertainty into account (i.e. risk-aware models), and thus do not follow the PRP [289], have been proposed [353, 403], showing significant effectiveness improvements compared to the models that do not model uncertainty. Uncertainty estimation is a difficult task that has other applications in IR besides improving the ranking effectiveness: it can be employed to decide between asking clarifying questions and providing a potential answer in conversational search [9]; to perform dynamic query reformulation [194] for queries where the intent is uncertain; and to predict questions with no correct answers [92].

5.2.2 Bayesian Neural Networks

Unlike standard algorithms to train neural networks, e.g. SGD, that fit point estimate weights given the observed data, Bayesian Neural Networks (BNNs) infer a distribution over the weights given the observed data. Denker et al. [78] contains one of the earliest mentions of choosing probability over the weights of a model. An advantage of the Bayesian treatment of neural networks [35, 214, 230] is that they are better at representing existing uncertainties in the training procedure. One limitation of BNNs is that they are computationally expensive compared to DNNs. This has led to the development of techniques that scale well and do not require modifications of the neural net architecture and training procedure. Gal and Ghahramani [100] proposed a way to approximate Bayesian inference by relying on dropout [312]. While dropout is a regularization technique that ignores units with probability p during every training iteration and is disabled at test time, `Dropout` [100] employs dropout at both train and test time and generates a predictive distribution after a number of forward passes. Lakshminarayanan et al. [177] proposed an alternative: they employ ensembles of models (`Ensemble`) to obtain a predictive distribution. Ovadia et al. [242] showed that `Ensemble` are able to produce well-calibrated uncertainty estimates that are robust to dataset shift.

5.3 Risk-Aware Neural Ranking

In this section, we introduce the methods used for answering the following research questions: **RQ1** *How calibrated are deterministic and stochastic BERT-based rankers?* **RQ2** *Are the uncertainty estimates from stochastic BERT-based rankers useful for risk-aware ranking?* **RQ3** *Are the uncertainty estimates obtained from stochastic BERT-based rankers useful for identifying unanswerable queries?* We first describe how to measure the calibration of neural rankers (**[C1]**), followed by our approach for modeling and ranking under uncertainty (**[C2]**), and then we describe how we evaluate their robustness to distributional shift.

5.3.1 Measuring Calibration

To evaluate the *calibration* of neural rankers (**RQ1**) we resort to the Empirical Calibration Error (ECE) [228]. ECE is an intuitive way of measuring to what extent the confidence scores from neural networks align with the true correctness likelihood. It measures the difference between the observed reliability curve [77] and the ideal one[3]. More formally, we sort the predictions of the model, divide them into c buckets $\{B_0, ..., B_c\}$, and take the weighted average between the average predicted probability of relevance $avg(B_i)$ and the fraction of relevant[4] documents $\frac{rel(B_i)}{|B_i|}$ in the bucket:

$$ECE = \sum_{i=0}^{c} \frac{|B_i|}{n} \left| avg(B_i) - \frac{rel(B_i)}{|B_i|} \right|,$$

[3] See examples of reliability diagrams in Figure 5.2.

[4] We consider here binary relevance.

where n is the total number of test examples.

5.3.2 Modeling Uncertainty

First we define the ranking problem we focus on, followed by the BERT-based ranker baseline model (BERT). Having set the foundations, we move to the methods we propose to answer **RQ2** and **RQ3**: a stochastic BERT-based ranker to *model uncertainty* (S-BERT) and a risk-aware BERT-based ranker to *take into account uncertainty provided by* S-BERT *when ranking* (RA-BERT).

Conversation Response Ranking

This is the typical conversation response ranking problem as defined in Section 2.6.1, with a small set of candidate responses, i.e. re-ranking step.

Deterministic BERT Ranker

We use BERT for learning the function $f(\mathcal{U}_i, r)$, based on the representation of the [CLS] token. The input for BERT is the concatenation of the context $\mathcal{U}_i$ and the response r, separated by [SEP] tokens. This is the equivalent of early adaptations of BERT for ad-hoc retrieval [377] transported to conversation response ranking. Formally the input sentence to BERT is $concat(\mathcal{U}_i, r) = u^1$ | [U] | u^2 | [T] | ... | u^τ | [SEP] | r, where | indicates the concatenation operation. The utterances from the context $\mathcal{U}_i$ are concatenated with special separator tokens [U] and [T] indicating the end of utterances and turns. The response r is concatenated with the context using BERT's standard sentence separator [SEP]. We fine-tune BERT on the target conversational corpus and make predictions as follows: $f(\mathcal{U}_i, r) = \sigma(FFN(BERT_{CLS}(concat(\mathcal{U}_i, r))))$, where $BERT_{CLS}$ is the pooling operation that extracts the representation of the [CLS] token from the last layer and FFN is a feed-forward network that outputs logits for two classes (relevant and non-relevant). We pass the logits through a softmax transformation σ that gives us a probability of relevance. We use the cross entropy loss for training. The learned function $f(\mathcal{U}_i, r)$ outputs a point estimate and we refer to it as BERT.

Stochastic S-BERT Ranker

In order to obtain a predictive distribution, $R_r = \{f(\mathcal{U}_i, r)^0, f(\mathcal{U}_i, r)^1, ..., f(\mathcal{U}_i, r)^n\}$, which allows us to extract uncertainty estimates, we rely on two techniques, namely Ensemble [177] and Dropout [100]. Both techniques scale well and do not require modifications on the architecture or training of BERT.

Using Deep Ensembles (S-BERT^E) We train M models using different random seeds without changing the training data, each with its own set of parameters $\{\theta_m\}_{m=1}^M$ and make predictions with each one of them to generate M predicted values:

$$R_r^E = \{f(\mathcal{U}_i, r)^0, f(\mathcal{U}_i, r)^1, ..., f(\mathcal{U}_i, r)^M\}$$

The mean of the predicted values is used as the predicted probability of relevance: S-BERT$^E(\mathcal{U}_i, r) = E[R_r^E]$, and the variance $var[R_r^E]$ gives us a measure of the uncertainty.

Using MC Dropout (S-BERT^D) We train a single model with parameters θ and employ dropout at test time and generate stochastic predictions of relevance by conducting T forward passes: $R_r^D = \{f(\mathcal{U}_i, r)^0, f(\mathcal{U}_i, r)^1, ..., f(\mathcal{U}_i, r)^T\}$. The mean of the predicted values is used as the predicted probability of relevance: S-BERT$^D(\mathcal{U}_i, r) = E[R_r^D]$, and the variance $var[R_r^D]$ gives us a measure of the uncertainty.

Risk-Aware RA-BERT Ranker

Given the predictive distribution R_r, obtained either by Ensemble or Dropout, we use the following function to rank responses with risk awareness:

$$\begin{aligned}\text{RA-BERT}(\mathcal{U}_i, r) = E[R_r] - b * var[R_r] \\ -2b \sum_{i}^{n-1} cov[R_r, R_{r_i}],\end{aligned}$$

where $E[R_r]$ is the mean of the predictive distribution, and b is a hyperparameter that controls the aversion or predilection towards risk. Unlike [409], we are not combining different runs that encompass different model architectures. We instead take a Bayesian interpretation of the process of generating a predictive distribution from a single model architecture. We refer to the rankers as RA-BERT^D and RA-BERT^E, when using S-BERT^D's predictive distribution and S-BERT^E's predictive distribution respectively.

5.3.3 Robustness to Distributional Shift

In order to evaluate whether we can trust the model's calibration and uncertainty estimates, similar to [242] we evaluate how robust the models are to different types of shifts in the test data. We do so by training the model using one setting and applying it in a different setting. Specifically for all three research questions we test the models under two settings— cross-domain and cross-negative sampling—which we describe next.

Cross Domain

We train a model using the training set from one domain known as the source domain $\mathcal{D}_{\mathcal{S}}$ and evaluate it on the test set of a different domain, known as the target domain $\mathcal{D}_{\mathcal{T}}$. This is also known as the problem of domain generalization [117].

Cross Negative Sampling

Pointwise L2R models are trained on pairs of query and relevant document and pairs of query and non-relevant document [207]. Selecting the non-relevant documents requires a *negative sampling* (NS) strategy. For the cross-NS condition, we test models on negative documents that were sampled using a different NS strategy than during training, evaluating the generalization of the models on a shifted distribution of candidate documents. The dataset on the other hand is always the same for the cross-NS condition. We use three NS strategies. In NS$_{random}$ we randomly select candidate responses from the list of all responses. For NS$_{BM25}$ we retrieve candidate responses using the conversational context $\mathcal{U}_i$ as a query to a lexical retrieval model (here BM25) and all the responses r as documents. In NS$_{sentenceBERT}$ we represent both $\mathcal{U}_i$ and all the responses with a sentence embedding technique and retrieve candidate responses using a similarity measure. Unless stated otherwise, we use $\mathtt{NS}_{BM25}$ as the negative sampling strategy.

5.4 Experimental Setup

In order to answer our research questions we consider three large-scale information-seeking conversation datasets introduced in Section 2.7.1: MANtIS, MSDialog, and UDC-DSTC8.

5.4.1 Implementation Details

We fine-tune BERT [80] (*bert-base-cased*) for conversation response ranking using the *huggingface-transformers* [363]. We follow recent research in IR that employed fine-tuned BERT for retrieval tasks [236, 377], including conversation response ranking [249, 344, 360]. When training BERT we employ a balanced number of relevant and non-relevant—sampled using BM25 [290]—context and response pairs. The sentence embeddings we use for cross-NS is sentenceBERT [280] and we employ dot product calculation from FAISS [155]. We consider each dataset as a different domain for cross-NS. We use the default hyperparameters: Adam optimizer [168] with $lr = 5^{-6}$ and $\epsilon = 1^{-8}$, we train with a batch size of 6 and fine-tune the model for 1 epoch. This baseline BERT-based ranker setup yields comparable effectiveness with SOTA methods[5].

5.4.2 Evaluation

To evaluate the *effectiveness* of the neural rankers we resort to a standard evaluation metric in conversation response ranking [113, 319, 384]: recall at position K with n candidates $R_n@K$. To evaluate the *calibration* of the models, we resort to the Empirical Calibration Error (see Section 5.3.1, using $C = 10$). Throughout, we report the test set results for each dataset. To evaluate the *quality of the uncertainty estimation* we rely on two downstream tasks. The first is to improve conversation response ranking itself via Risk-Aware ranking (see Section 5.3.2). The second, which fits well with conversation response ranking, is to predict unanswerable conversational contexts. Formally the task is to predict whether there is a correct answer in the candidates list $\mathcal{R}$ or not. In our experiments, for half of the instances, we remove the relevant response from the list, setting the label as *None Of The Above* (NOTA). The other half of the data has the label *Answerable* (ANSW) indicating that there is a suitable answer in the list of candidates, for which we remove one of the negative samples instead. Similar to Feng et al. [92], who proposed to use the outputs (logits) of a LSTM-based model in order to predict NOTA, we use the uncertainties as additional features to the classifier for NOTA prediction. The input space with the additional features is fed to a learning algorithm (Random Forest), and we evaluate it with a 5-fold cross-validation procedure using F1-Macro.

[5] We obtain 0.834 $R_{10}@1$ on UDC-DSTC8 with our baseline BERT model, c.f. Table 5.1, while SA-BERT [113] achieves 0.830. The best-performing model of the DSTC8 [167] also employed a fine-tuned BERT

5.5 Results

5.5.1 Calibration of Neural Rankers

In order to answer our first research question about the calibration of neural rankers, let us first analyze BERT under standard settings (no distributional shift). Our results show that BERT is both effective and calibrated under no distributional shift conditions. In Table 5.1 we see that when the target data (*Test on* ⟶) is the same as the source data (*Train on* ↓)—indicated by underlined values—we obtain the highest effectiveness (on average 0.70 R_{10}@1) and the lowest calibration error (on average 0.036 ECE). When plotting the calibration curves of the model in Figure 5.2, we observe the curves to be almost diagonal (i.e. having near perfect calibration) when there are an equal number of relevant and non-relevant candidates (#-non-rel = 1).

Table 5.1: Calibration (ECE, lower is better) and effectiveness (R_{10}@1, higher is better) of BERT for conversation response ranking in cross-domain, and cross-NS conditions. All models were trained using NS_{BM25}. ECE is calculated using a balanced number of relevant and non relevant documents. Underlined values indicate no distributional shift ($\mathscr{D}_{\mathcal{S}} = \mathscr{D}_{\mathcal{T}}$ and train NS = test NS). For the cross-NS conditions the train dataset is the same as the test dataset, and models trained with NS_{BM25} are tested against NS_{random} and $\text{NS}_{sentenceBERT}$.

	cross-domain						**cross-NS**			
Test on ⟶	MANtIS		MSDialog		UDC-DSTC8		NS_{random}		$\text{NS}_{sentenceBERT}$	
Train on ↓ (NS_{BM25})	**R_{10}@1**	**ECE**	**R_{10}@1**	**ECE**	**R_{10}@1**	**ECE**	**R_{10}@1**	**ECE**	**R_{10}@1**	**ECE**
MANtIS	0.615	0.003	0.653	0.010	0.422	0.028	0.263	0.011	0.310	0.009
MSDialog	0.398	0.009	0.652	0.006	0.495	0.014	0.298	0.029	0.239	0.027
UDC-DSTC8	0.349	0.016	0.306	0.023	0.834	0.002	0.318	0.050	0.182	0.045

However, when we make the conditions more realistic[6] by having multiple non-relevant candidates for each conversational context, we observe in Figure 5.2 that the calibration errors start to increase, moving away from the diagonal. Additionally, when we challenge the model in cross-domain and cross-NS settings, the calibration error increases significantly as evident in Table 5.1. On average, the ECE is **4.6** times higher for cross-domain and **7.9** times higher for cross-NS. Thus **answering the first part of our first research question about the calibration of deterministic BERT-based rankers, indicating that they do not have robust calibrated predictions**, failing on the scenarios where there is a distributional shift.

In order to answer the remaining part of RQ1, on how calibrated *stochastic* BERT-based rankers are, let us consider Tables 5.2 and 5.3. They display the improvements (relative drop in ECE) over BERT in terms of calibration. S-BERT^{E} is on average **14%** better (has less calibration error) than BERT, while S-BERT^{D} is on average **10%** better than BERT, **answering our first research question: stochastic BERT-based rankers are better calibrated than deterministic BERT-based ranker**. We hypothesize that S-BERT^{E} leads to less ECE than S-BERT^{D} because it better captures the model uncertainty in the training procedure since it combines different weights that explain equally well the prediction of relevance

[6] In a production system, the retrieval stage would be executed over all candidate responses. As a consequence, the data is highly unbalanced, i.e. only a few relevant responses among potentially millions of non-relevant responses.

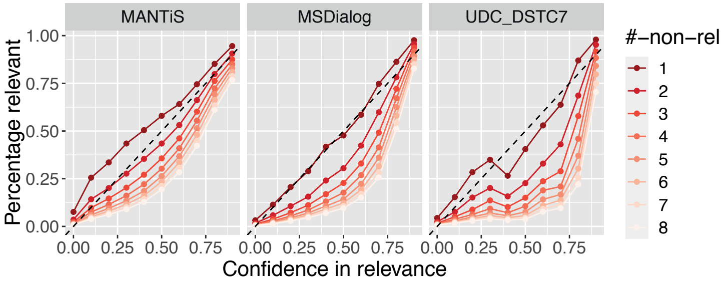


Figure 5.2: Calibration of BERT trained on a balanced set of relevant and non-relevant documents, and tested data with more non-relevant (#-non-rel) than relevant (1 per query) documents. A fully calibrated model is represented by the dotted diagonal: for every bucket of confidence in relevance, the % of relevant documents in that bucket is exactly the confidence. The calibration error is the difference between the curves and the diagonal.

given the inputs. In the next section, we focus on evaluating the effectiveness of such models that are better calibrated and also taking into account uncertainty when ranking.

Table 5.2: Relative decreases of ECE (lower is better) of S-BERT^E and S-BERT^D over BERT for the cross-domain condition. Superscript † denote significant improvements (95% confidence interval) using Student's t-tests.

	cross-domain					
Test on →	MANtIS		MSDialog		UDC-DSTC8	
Train on ↓ (NS$_{BM25}$)	S-BERT^E	S-BERT^D	S-BERT^E	S-BERT^D	S-BERT^E	S-BERT^D
MANtIS	-35.13%†	-56.14%†	-03.42%	-26.89%†	-04.94%	-00.83%
MSDialog	$_+$25.05%	$_+$08.27%	-43.11%	-11.54%	$_+$22.77%	$_+$05.85%
UDC-DSTC8	-54.95%†	-09.98%†	-25.78%†	-09.15%	$_+$24.77%	-01.84%

Table 5.3: Relative decreases of ECE (lower is better) of S-BERT^E and S-BERT^D over BERT for the cross-NS condition. Superscript † denote significant improvements (95% confidence interval) using Student's t-tests.

	cross-NS			
Test on →	NS$_{random}$		NS$_{sentenceBERT}$	
Train on ↓ (NS$_{BM25}$)	S-BERT^E	S-BERT^D	S-BERT^E	S-BERT^D
MANtIS	-31.35%	-18.65%†	-37.65%†	-02.79%
MSDialog	-15.91%	-10.58%	-17.17%	-12.93%
UDC-DSTC8	-08.05%	-01.78%	-04.81%	-01.28%

5.5.2 Uncertainty Estimates for Risk-Aware Neural Ranking

In order to evaluate the quality of the uncertainty estimations, we first resort to using them as a measure of the risk through risk-aware neural ranking (RA-BERT^D and RA-BERT^E).

5

Figure 5.3 displays the effectiveness in terms of $R_{10}@1$ gains over BERT for the different settings (cross-domain and cross-NS) when varying the risk aversion b.

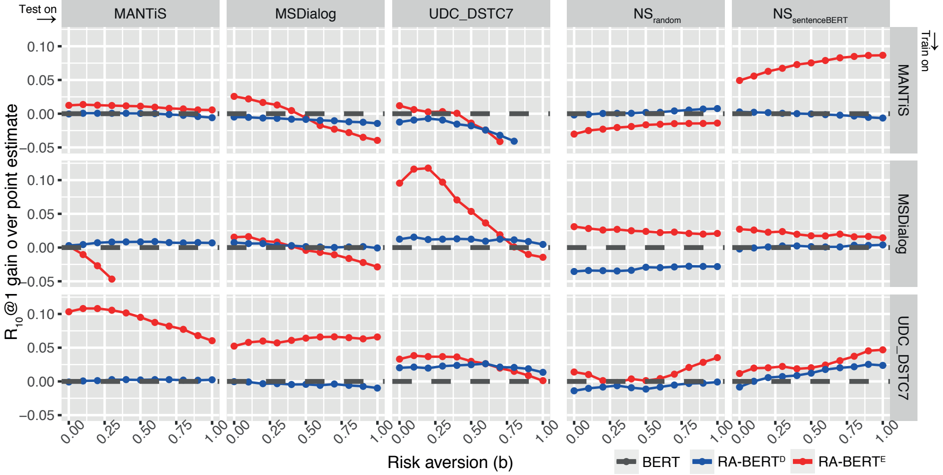


Figure 5.3: Gains of the Risk-Aware BERT-ranker for different values of risk aversion b (the importance of the uncertainty estimation for the final ranking).

We note that when $b = 0$, we are using the mean of the predictive distribution and disregard the risk, which is equivalent to S-BERT^D and S-BERT^E. The ensemble-based average S-BERT^E is more effective than the baseline BERT for almost all combinations and S-BERT^D is equivalent to the baseline. This is in line with previous work that ensemble and stacking approaches are more effective than using single models [2, 40, 41] and in line with public leaderboards and machine learning competitions [176].

When using $b < 0$, we are ranking with risk predilection (the opposite of risk aversion), and in all conditions, we found that the effectiveness was significantly worse than when $b = 0$ and thus $b < 0$ is not displayed in Figure 5.3.

When increasing the risk aversion ($b > 0$), we see that it has different effects depending on the combination of domain and NS. For instance, when training in MSDialog and applying on UDC-DSTC8, increasing the risk aversion improves the effectiveness of RA-BERT^E until b reaches 0.25, and after that the effectiveness drops.

In order to investigate whether ranking with risk aversion is more effective than using the predictive distribution mean, we select b based on the best value observed on the validation set. Tables 5.4 and 5.5 display the results of this experiment, showing the improvements of RA-BERT^D and RA-BERT^E over S-BERT^D and S-BERT^E respectively. The results show that in a few cases (8 out of 30) the best value of b is 0, for which risk-aversion is not the best option in the development set. We obtain effectiveness improvements primarily on the cross-NS condition (up to 17.2% improvement of $R_{10}@1$), which is the hardest condition (when the models are mostly ineffective, c.f. Table 5.1). **This answers our second research question, indicating that the uncertainties obtained from stochastic neural rankers are useful for risk-aware ranking, especially in the cross-NS set-**

ting where the baseline model is quite ineffective. RA-BERT^E is on average 2% more effective than S-BERT^E, while RA-BERT^D is on average 1.7% more effective than S-BERT^D.

Table 5.4: Relative improvements (higher is better) of R_{10}@1 of RA-BERT^E and RA-BERT^D over the mean of stochastic BERT predictions (S-BERT^E and S-BERT^D) for the cross-domain condition. Superscript † denote statistically significant improvements over the S-BERT ranker at 95% confidence interval using Student's t-tests.

	cross-domain					
Test on →	MANtIS		MSDialog		UDC-DSTC8	
Train on ↓ (NS$_{BM25}$)	RA-BERT^E	RA-BERT^D	RA-BERT^E	RA-BERT^D	RA-BERT^E	RA-BERT^D
MANtIS	-0.14%	+0.16%†	+0.00%	+0.00%	+0.00%	+0.00%
MSDialog	-2.74%	+0.39%	-1.05%	-0.66%	+5.08%†	-0.10%
UDC-DSTC8	+0.00%	+0.00%	+0.00%	+0.00%	+0.42%	-0.06%

Table 5.5: Relative improvements (higher is better) of R_{10}@1 of RA-BERT^E and RA-BERT^D over the mean of stochastic BERT predictions (S-BERT^E and S-BERT^D) for the cross-NS condition. Superscript † denote statistically significant improvements over the S-BERT ranker at 95% confidence interval using Student's t-tests.

	cross-NS			
Test on →	NS$_{random}$		NS$_{sentenceBERT}$	
Train on ↓ (NS$_{BM25}$)	RA-BERT^E	RA-BERT^D	RA-BERT^E	RA-BERT^D
MANtIS	+4.73%†	+4.58%†	+9.68%†	-2.68%
MSDialog	-7.61%	+3.29%	-0.61%	+0.63%
UDC-DSTC8	+6.32%†	+3.83%†	+16.39%†	+17.18%†

5.5.3 Uncertainty Estimates for NOTA Prediction

Besides using the uncertainty estimation for risk-aware ranking, we also employ it for the NOTA (None of the Above) prediction task. We compare here different input spaces for the NOTA classifier. $E[R^D]$ stands for the input space that only uses the mean of the predictive distribution for the k candidate responses in $\mathscr{R}$ using S-BERT^D, $+var[R^E]$ uses both $E[R^D]$ and the uncertainties of S-BERT^E for the k candidates and $+var[R^D]$ uses both the scores $E[R^D]$ and the uncertainties of S-BERT^D. Our results show that the uncertainties from S-BERT^D and of S-BERT^E significantly improve the F1 for NOTA prediction for both cross-domain (Table 5.6, improvement of 24% on average when using S-BERT^D) and cross-NS settings (Table 5.7, improvement of 46% on average when using S-BERT^D). **We can thus answer our last research question: the uncertainty estimates from stochastic neural rankers do improve the effectiveness of the NOTA prediction task (by an average of 33% across all conditions considered).**

Table 5.6: Results of the *cross-domain* condition for the NOTA prediction task, using a Random Forest classifier and different input spaces. The F1-Macro and standard deviation over the 5 folds of the cross validation are displayed. Superscript † denote statistically significant improvements over $E[R^D]$ at 95% confidence interval using Student's t-tests. Bold indicates the most effective approach.

	cross-domain					
Test on →	MANtIS			MSDialog		
Train on ↓ (NS$_{BM25}$)	$E[R^D]$	$+var[R^E]$	$+var[R^D]$	$E[R^D]$	$+var[R^E]$	$+var[R^D]$
MANtIS	0.635 (.02)	0.686 (.01)†	**0.792 (.02)**†	0.669 (.03)	0.731 (.04)	**0.855 (.02)**†
MSDialog	0.561 (.02)	0.598 (.02)†	**0.633 (.02)**†	0.662 (.04)	**0.702 (.01)**†	0.699 (.06)†
UDC-DSTC8	0.527 (.04)	0.665 (.02)†	**0.738 (.03)**†	0.523 (.05)	0.691 (.03)†	**0.757 (.04)**†

Table 5.7: Results of the cross negative sampling condition for the NOTA prediction task, using a Random Forest classifier and different input spaces. The F1-Macro and standard deviation over the 5 folds of the cross validation are displayed. Superscript † denote statistically significant improvements over $E[R^D]$ at 95% confidence interval using Student's t-tests. Bold indicates the most effective approach.

	cross-NS					
Test on →	NS$_{random}$			NS$_{sentenceBERT}$		
Train on ↓ (NS$_{BM25}$)	$E[R^D]$	$+var[R^E]$	$+var[R^D]$	$E[R^D]$	$+var[R^E]$	$+var[R^D]$
MANtIS	0.557 (.01)	0.604 (.02)†	**0.698 (.02)**†	0.534 (.03)	0.587 (.02)†	**0.647 (.05)**†
MSDialog	0.505 (.02)	0.606 (.02)†	**0.702 (.05)**†	0.522 (.03)	0.611 (.07)†	**0.653 (.04)**†
UDC-DSTC8	0.565 (.03)	0.800 (.02)†	**0.942 (.04)**†	0.506 (.05)	0.755 (.05)†	**0.821 (.05)**†

5.6 Limitations

One of the limitations of this work is that we only use a single ranker to test our hypotheses. We believe our findings might generalize to other neural ranking architectures, as well as other tasks. Additionally, the experiments focus on the re-ranking procedure and the same could be tested for retrieval. Our out-of-domain evaluation is limited as all datasets were extracted from online forums. How a method trained in such a dataset would generalize to other types of datasets, e.g. extracted through a wizard-of-oz experiment, is unknown.

5.7 Conclusions

In this work, we study the calibration and uncertainty estimation of neural rankers, specifically BERT-based rankers. We first show that the deterministic BERT-based ranker is not robustly calibrated for the task of conversation response ranking and we improve its calibration with two techniques to estimate uncertainty through *stochastic neural ranking*. We also show the benefits of estimating uncertainty using risk-aware neural ranking and for predicting unanswerable conversational contexts.

This chapter provides further evidence for the second main research question (M-RQ2), showing that different notions of the difficulty of a dialogue can be used to improve a re-ranking model for conversational search. Specifically, we show how to model the uncertainty of a cross-encoder model. This notion of difficulty can be then used by the re-ranker model as shown in our risk-aware model to consider both the relevance prediction and the uncertainty to produce the final ranked list. We finish here the chapters of the thesis related to improvements to the multi-stage pipeline for conversational search. Next, we start an investigation of the limitations of such pipelines for conversational search and recommendation in order to answer our third main research question (M-RQ3).

IV

Understanding Ranking Models for Conversational Search and Recommendation

6

Evaluating Retrieval Pipelines with Language Variations of Questions

In this chapter, we start to explore the limitations of multi-stage retrieval pipelines. IR benchmarks evaluate the effectiveness of retrieval pipelines based on the premise that a single query, or utterance in the case of conversational search, is used to instantiate the underlying information need. However, previous research has shown that (I) queries generated by users for a fixed information need are extremely variable, and, in particular, (II) neural models are brittle and often make mistakes when tested with modified inputs. Motivated by those observations we aim to answer the following question: how robust are retrieval pipelines with respect to different variations in queries that do not change the queries' semantics? In order to obtain queries that are representative of users' querying variability, we first created a taxonomy based on the manual annotation of transformations occurring in a dataset (`UQV100`*) of user-created query variations. For each syntax-changing category of our taxonomy, we employed different automatic methods that when applied to a query generate a query variation. Our experimental results across two datasets for two IR tasks reveal that retrieval pipelines are not robust to these query variations, with effectiveness drops of* ≈ 20% *on average. The code required to reproduce this chapter is available at* `https://github.com/Guzpenha/query_variation_generators`.

This chapter is based on the following paper:

🗎 *Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In ECIR. Springer, 397–412* [247]. This paper received the best paper award at ECIR 2022 🏆.

6.1 Introduction

Heavily pre-trained transformers for language modeling such as BERT [80] have been shown to be remarkably effective for a wide range of Information Retrieval (IR) tasks [236, 249, 377]. Commonly, IR benchmarks organized as part of TREC or other evaluation campaigns, evaluate the effectiveness of ranking models—neural or otherwise—based on small sets of topics and their corresponding relevance judgments. Importantly, each topic is typically represented by a single query[1]. However, previous research has shown that queries created by users given a fixed information need may vary widely [22, 410]. In the UQV100 [21] dataset for instance, crowd workers on average created 57.7 unique queries for a given information need as instantiated as a backstory, e.g. *"You have heard quite a lot about cheap computing as being the way of the future, including one recent model called a Raspberry Pi. You start thinking about buying one, and wonder how much they cost."*

Table 6.1: Examples of BERT effectiveness drops (nDCG@10 Δ) when we replace the original query from TREC-DL-2019 by an automatic (except for the first two lines that were produced manually) query variation. We focus here on transformations that change the query syntax, but not its semantics.

Original Query	**Query Variation**	**nDCG@10 Δ**
popular food in switzerland	popular food in zurich *gen./specialization*	
cost of interior concrete flooring	concrete flooring finishing *aspect change*	
what is theraderm used for	what is **thrraderm** used for *misspelling*	-1.00 (-100%)
anthropological definition of environment	anthropological definition ~~**of**~~ environment *naturality*	-0.15 (-26%)
right pelvic pain causes	**causes** pelvic pain **right** *ordering*	-0.18 (-46%)
define visceral	**what is** visceral *paraphrasing*	-0.26 (-38%)

We thus argue that it is necessary to investigate the robustness of retrieval pipelines in light of *query variations* (i.e., different expressions of the same information need) that are *likely to occur in practice*. That different query variations lead to vastly different ranking qualities is anecdotally shown in Table 6.1 for a vanilla BERT model for ranking [236]. If, for example, the word order of the original query from TREC-DL-2019 *right pelvic pain causes* is changed to *causes pelvic pain right*, the retrieval effectiveness of the resulting ranking drops by 46%. Similarly, paraphrasing *define visceral* to *what is visceral* reduces the retrieval effectiveness by 38%.

In this chapter, we quantify the extent to which different retrieval pipelines (composed of first-stage retrieval and second-stage re-ranking as described in the introduction of this thesis) are susceptible to different types of query variations as measured by their drop in

[1] The same procedure is taken for conversational search and recommendation tasks, where each information-need dialogue is represented by unique utterances

retrieval effectiveness. Also, different from other chapters we consider here a simpler case of one-shot interactions with the system (queries) as opposed to conversations[2].

In contrast to prior works that either analyze behaviour of models when faced with modifications to the documents [209], analyze models through the lens of IR axioms [49, 283] or analyze NLP models via general natural language text adversarial examples [108, 285], we instantiate our *query variations* based on user-created data. Concretely, we manually label a large fraction of UQV100 queries[3] and extract six types of frequently occurring query transitions: *gen./specialization*, *aspect change*, *misspelling*, *naturality*, *ordering* and *paraphrasing*—an example of each is shown in Table 6.1. The last four of these categories change the query syntax but not its semantics. For each of the syntax-changing categories, we develop automated approaches that enable us to generate query variations of each category for any input query. With these *query variation generators* in place, we contribute extensive empirical work on the recent TREC-DL-2019 [68] and ANTIQUE [128] datasets to answer the following research question: *Are retrieval pipelines robust to different variations in queries that do not change its semantics?* To this end we consider seven ranking approaches: two traditional lexical models (BM25 [290] and RM3 [1]), two neural re-ranking approaches that do not make use of transformers (KNRM [369] and CKNRM [73]) and three transformer-based re-ranking approaches (EPIC [210], BERT [236] and T5 [237]). Additionally, motivated by the fact that certain query variations can improve the retrieval effectiveness compared to using the original query [27, 33], we contribute with a study of the combination of automatic query variations with rank fusion [67].

Our main findings are as follows:

- The four types of syntax-changing query variations differ in the extent to which they degrade retrieval effectiveness: *misspellings* have the largest effect (with an average drop of 0.25 nDCG@10 points across seven retrieval models for TREC-DL-2019) while the *word ordering* has the least effect (with an average drop of nDCG@10 smaller than 0.01 for TREC-DL-2019).
- Different types of ranking models make similar mistakes. For example, effectiveness decreases for models based on transformer language models are higher for *naturality* query variations compared to decreases when using traditional lexical models.
- While rank fusion mitigates the drops in retrieval effectiveness when compared to using a single query variation, it does not achieve the full potential of the combination of query variations. An oracle that always selects the best query achieves gains of 0.08 and 0.06 nDCG@10 points on TREC-DL-2019 and ANTIQUE respectively.

Our work indicates that more research is required to improve the robustness of retrieval pipelines. Evaluation benchmarks should aim to have multiple query variations for the same information need in order to evaluate whether ranking pipelines are indeed robust, and we provide here a number of methods to automatically generate such query variations for any dataset.

[2] We believe that the results from this chapter would generalize to the first utterance in information-seeking dialogues, but leave this exploration as future work.

[3] To our knowledge, UQV100 is the only publicly available dataset that contains a large number of query variations for a set of information needs.

6.2 Related Work

To put our work in context, we now describe prior research into query variations and then move on to research analyzing neural (IR) models.

6.2.1 Query Variation

A number of studies have argued that evaluation in IR tasks should take into account multiple instantiations of the same information need, i.e. query variations, due to their impact on the effectiveness of ranking models [20–22, 26, 45, 224, 311, 410]. Zuccon et al. [410] proposed a mean-variance framework to explicitly take into account query variations when comparing different IR systems. Bailey et al. [22] argued that a model should be consistent with different query variations, and proposed a measure of consistency that gives additional information to effectiveness measurements.

Besides a better evaluation of models, query variations can also be employed to improve the overall effectiveness of ranking models, for instance by combining the different rankings obtained from them [27, 33] or by modeling relevance of multiple query variations [206]. They have also shown to be helpful for query performance prediction [390].

Different methods to automatically generate query variations have been proposed. Benham et al. [32] proposed to obtain query expansions through a relevance model which is built by issuing the original query against an external corpus and expanding it with additional terms from the set of external feedback documents. Lu et al. [206] employed a query-URL click graph and generated query variations automatically using a two-step backward walk process. Chakraborty et al. [52] generated query variations automatically based on an external knowledge base with a prior term distribution or by building a relevance model in an iterative manner. Our work differs from previous work on automatic query variation generation in the following ways:

- Our methods do not require access to external corpora, a relevance model, or a query-URL click graph.
- We are not concerned with generating queries with the sole purpose of improving effectiveness, but with generating queries that are likely to occur in practice.
- Each of our generator methods follows a category of our taxonomy of query variations which allows us to *diagnose* ranking models' effectiveness by analyzing what types of variations are more detrimental to what ranking models.

6.2.2 Model Understanding

The success of pre-trained transformer-based language models such as BERT [80] and T5 [274] on several IR benchmarks—a comprehensive account of the effectiveness gains can be found in [193]—has lead to research on understanding their behaviour and the reasons behind their significant gains in ranking effectiveness [49, 209, 243, 265, 393].

Câmara and Hauff [49] showed that BERT does not adhere to IR axioms, i.e., heuristics that a reasonable IR model should fulfill, through the use of diagnostic datasets. MacAvaney et al. [209] expanded on the axiomatic diagnostic datasets [283] with ABNIRML, a framework to understand the behaviour of neural ranking models using three different strategies: measure and match (controlling certain measurements such as relevance or term frequency and changing another), manipulation of the documents' text (for example by shuffling words or replacing it with the query) and through the transfer of Natural Language Processing (NLP) datasets (for example comparing documents that are more/-less fluent or formal with inferred queries). We expand on MacAvaney et al. [209]'s work by proposing textual manipulations—unlike previous methods, we are inspired by *user-created* variations—to the queries instead of the documents and examine the robustness in terms of the effectiveness of neural ranking models to such manipulations.

A different direction of research in NLP has challenged how well current evaluation schemes through the use of held-out test sets are actually evaluating the desired capabilities of the models [34, 38, 189]. For example, Gardner et al. [108] proposed the manual creation of contrast sets—small perturbations that preserve artifacts but change the true label—in order to evaluate the models' decision boundaries for different NLP tasks. They showed that the model effectiveness on such contrast sets can be up to 25% lower than on the original test sets. Inspired by behavioral testing, i.e. validating input-output behaviour without knowledge about the internal structure, from software engineering tests, Ribeiro et al. [285] proposed to test NLP models with three different types of tests: minimum functionality tests (simple examples where the model should not fail), label (e.g. positive, negative and neutral in sentiment analysis) invariant changes to the input, and modifications to the input with known outcomes. With such tests, they were able to find actionable failures in different commercial models that had already been extensively tested.

It has also been shown that neural models developed for different NLP tasks can be tricked by adversarial examples [11, 102, 109], i.e. examples with perturbations indiscernible by humans which get misclassified by the model. In terms of query modifications, [366, 405] found typos to be detrimental to the effectiveness of neural rankers. Wu et al. [366] analyzed the robustness of neural rankers with respect to three dimensions: difficult queries from similar distribution, out-of-domain cases, and defense against adversarial operations. Our work differs from the adversarial line of research by evaluating the robustness of models to query modifications that could be generated by humans, i.e. transformations that naturally occur, and not modifications optimized to trick neural models.

6.3 Automatic Query Variations

We now first describe in Section 6.3.1 how we arrive at our query variation categories in a data-driven manner by annotating a large set of user-created query variations from `UQV100`. We end up with six categories: four that change the syntax (but not the semantics) and two that change the semantics. **In our work, we focus on the four syntax-changing categories.** In Section 6.3.2 we subsequently describe our methods to automatically generate query variations categories that do not change the query semantics.

6.3.1 UQV Taxonomy

In order to better understand how queries differ when we compare different query variations for the same information need, we resort to analyzing variations from the `UQV100` dataset. `UQV100` contains query variations for 100 (sub)-topics from the TREC 2013 and 2014 web tracks, written by crowd workers who received a "backstory" for each topic as a starting point. On average, `UQV100` contains 57.7 spelling corrected (corrected by the

Table 6.2: Taxonomy of query variations derived from a sample of the UQV100 dataset. Last column is the count of each query variation found on UQV100 based on manual annotation of tuples of queries for the same information need. Categories in grey change the semantics. * typos were already fixed for the UQV100 pairs.

Category	Definition	$\{q_i, q_j\}$ **from** UQV100			Count
Gen./specializ	Generalizes or specializes within the same information need.	american civil war	↔	number of battles in south carolina during civil war	172
Aspect change	Moves between related but different aspects within the same information need.	what types of spiders can bite you while gardening	↔	signs of spider bite	111
Misspelling	Adds or removes spelling errors.	raspberry pi	↔	raspeberry pi	*
Naturality	Moves between keyword queries and natural language queries.	how does zinc relate to wilson's disease	↔	zinc wilson's disease	118
Ordering	Changes the order of words	carotid cavernous fistula treatment.	↔	treatment carotid cavernous fistula	37
Paraphrasing	Rephrases the query by modifying one or more words.	cures for a bald spot	↔	cures for baldness	215

UQV100 authors using the spelling service of the Bing search engine) query variations per topic. We consider a query variation pair $\{q_i, q_j\}$ to be two queries q_i and q_j that were provided in UQV100 for the same backstory. In total, 365K such pairs exist; Table 6.2 (4th column) contains a number of $\{q_i, q_j\}$ examples. We sampled 100 pairs from the 365K available ones for manual annotation. The three authors of the paper that originated this chapter (the "annotators") performed an open card sort [365]. The annotators independently sorted the query variation pairs into different piles and named them, each representing a transformation T that can be applied to q_i and then leads to q_j, i.e. $T(q_i) = q_j$. Multiple transformations can be applied to q_i in order to yield q_j, e.g. $T_2(T_1(q_i)) = q_j$.

After the independent sorting step, the different piles were discussed and merged where necessary, which yielded five categories of transformations. Since the UQV100 data used had already been spelling-corrected by its authors, we added the category *misspellings*. The resulting taxonomy can be found in Table 6.2. It contains a concrete definition and examples for each of our—in total—six categories: (I) *generalization or specialization*, (II) *aspect change*, (III) *misspelling*, (IV) *naturality*, (V) *word ordering* and (VI) *paraphrasing*. We observed two broad types of transformations: transformations that change the semantics of the query and transformations that do not change the semantics. The *gen./specialization* and *aspect change* transformations fall into the former type, whereas all other categories fall into the latter. We highlight here that, unlike previous categorizations that describe how users revise queries in e-commerce [12, 134], how to generate better queries to substitute the original query [157], how users reformulate queries in a session [145], we study here how to categorize *query variations* for the same information need which is a related but different problem.

Having arrived at our six categories, our annotators then labeled an additional set of 550 $\{q_i, q_j\}$ randomly sampled pairs from UQV100 in order to determine the distribution of these categories in UQV100. Each $\{q_i, q_j\}$ was labeled as belonging to one (or more) of the five categories (with the exception of *misspelling* which, as already stated, had already been corrected by the UQV100 authors). In order to determine the inter-annotator agreement, 25 $\{q_i, q_j\}$ pairs were labeled by all three annotators, and 175 pairs were each labeled by a single annotator. The inter-annotator agreement [64] was moderate (Cohen's $\kappa = 0.42$); the disagreements were highest for the *naturality* and *paraphrasing* categories. We found that a total of 56 $\{q_i, q_j\}$ pairs had more than one category assigned to it[4]. The resulting distribution is shown in Table 6.2 (right-most column); the categories of query variations that change the query without changing its semantics account for 57% of all the transformations. In contrast, 43% of query variations are semantic changes. Among the syntax-changing categories, we found *naturality* to be the most common with 33% of all transformations falling into this category. Having observed that query variations change the syntax, but not the semantics for the majority of cases, **we focus in the remainder of our work on syntax-changing query variations**. We leave the exploration of query variation generators for *gen./specialization* and *aspect change* as future work.

Table 6.3: Example of applying each query generation method M for the query '*what is durable medical equipment consist of*' from TREC-DL-2019. Rightmost columns indicate the total percentage of valid queries by automatic query variation method based on manual annotation of queries from the test sets of TREC-DL-2019 and ANTIQUE.

C	**Method Name**	M(**'*what is durable medical equipment consist of*'**)	**TREC**	**ANT**
Misspelling	NeighbCharSwap	*what is durable* ***mdeical*** *equipment consist of*	100.00%	99.50%
	RandomCharSub	*what is durable* ***medycal*** *equipment consist of*	97.67%	91.00%
	QWERTYCharSub	*what is durable medical equipment* ***xonsist*** *of*	97.67%	98.50%
Naturality	RemoveStopWords	***~~what is~~*** *durable medical equipment consist* ***~~of~~***	86.05%	99.50%
	T5DescToTitle	***~~what is~~*** *durable medical equipment* ***~~consist of~~***	81.40%	68.00%
Ordering	RandomOrderSwap	***medical*** *is durable* ***what*** *equipment consist of*	100.00%	100.00%
Paraphrasing	BackTranslation	*what is* ***sustainable*** *medical equipment* *~~consist of~~*	53.49%	46.50%
	T5QQP	*what is durable medical equipment* ***~~consist of~~***	60.47%	52.50%
	WordEmbedSynSwap	*what is durable* ***medicinal*** *equipment consist of*	62.79%	62.00%
	WordNetSynSwap	*what is* ***long lasting*** *medical equipment consist of*	37.21%	35.50%

6.3.2 Query Generators

For each of the four syntax-changing categories, we explored different methods that generate query variations of the specified category. After an initial exploration of different query generator methods for each category, filtering approaches that did not generate valid variations for the category and approaches that have a high correlation with each other, we employed a total of ten different methods. These methods are listed in Table 6.3, each with an example transformation. We explain each one in more detail in this section. A method M_C receives as input a query q and outputs a query variation $\hat{q}$: $M_C(q) = \hat{q}$.

[4] For example, the pair {*"what is doctor zhivago all about"*, *"dr zhivago synopsis"*} had both *paraphrasing* and *naturality* labels, as it goes from a natural language question to a keyword-base question and also paraphrases *"doctor [...] all about"* to *"dr [...] synopsis"*

While most of the methods can generate multiple variations for a single input query (for example by replacing different words of the same query by synonyms or by including several spelling mistakes), for the experiments in the paper we resort to using a single query variation per method which already yields enough data for analysis (see § 2.7.1). Inspired by adversarial examples, we aim to make minimal perturbations to the input text when possible, e.g. replace only one word by a synonym, increasing the chances of obtaining valid variations.

Misspelling

The three methods in this category add one spelling error to the query; the query term an error is introduced in is chosen uniformly at random.

- `NeighbCharSwap`: Swaps two neighboring characters from a random query term (excluding stopwords[5]).
- `RandomCharSub`: Replaces a random character from a random query term (excluding stopwords) with a randomly chosen new ASCII character.
- `QWERTYCharSub`: Replaces a random character of a random query term (excluding stopwords) with another character from the keyboard such that only characters in close proximity are chosen, replicating errors that come from typing quickly.

Naturality

The two methods in this category transform natural language queries into keyword queries.

- `RemoveStopWords`: Removes all stopwords from the query.
- `T5DescToTitle`: Applies an encoder-decoder transformer model (here we employ T5 [274]) that we fine-tuned on the task of generating the title of a TREC topic title based on the TREC topic description. For example, a title and description tuple from *'trec-robust04'*: '*Evidence that rap music has a negative effect on young people.*' → '*Rap and Crime*'. We collect pairs of titles and descriptions from eleven datasets available through the IR datasets library [212]: `trec-robust04`, `trec-tb-2004`, `aquaint/trec-robust-2005`, `gov/trec-web-2002`, `ntcir-www-2`, `ntcir-www-3`, `trec-misinfo-2019`, `cord19/trec-covid`, `dd-trec-2015`, `dd-trec-2016` and `dd-trec-2017`. Overall, we fine-tuned our model on 1322 description/title tuples.

Ordering

In this category, we employ only one basic method to shuffle words as done by previous research on the order of words [209, 258].

- `RandomOrderSwap`: Randomly swap two words of the query.

[5] We use the NLTK english stopwords list for all the methods; it is available at `https://www.nltk.org/`.

Paraphrasing

The four methods in this category change query terms in the process of paraphrasing.

- `BackTranslation`: Applies a translation method to the query to a pivot language, i.e. an auxiliary language, and from the pivot language back to the original language of the query, i.e. English. In our experiments, we employ the M2M100 [90] model, a multilingual model that can translate between any pair of 100 languages, and we use *'German'* as the pivot language, which yielded better results—shown by manual inspection of the generated variations—than the other two languages for which the model had the most data for training (*'Spanish'* and *'French'*). This technique has been used before as a way to generate paraphrases [91, 217].
- `T5QQP`: Applies an encoder-decoder transformer model (here we employ T5 [274]) that was fine-tuned on the task of generating a paraphrase question from the original question[6]. The model employs the Quora Question Pairs[7] dataset for fine-tuning, which has 400k pairs of questions like the following: '*How do you start a bakery?*' → '*How can one start a bakery business?*'. We also tested T5 models fine-tuned for PAWS [378] and the combination of PAWS and Quora Question Pairs, but the manual inspection of the generated queries revealed that T5 fine-tuned for Quora Question Pairs generated a higher number of valid variations.
- `WordEmbedSynSwap`: Replaces a non-stop word with a synonym as defined by the nearest neighbor word in the embedding space according to a counter fitted-Glove embedding which yields better synonyms than standard Glove embeddings [227].
- `WordNetSynSwap`: Replaces a non-stop word by a the first synonym found on WordNet [8]. If there are no words with valid synonyms it will not output a valid variation.

6

6.4 Experimental Setup

In this section, we describe our experimental setup aimed to answer the question: *are retrieval pipelines robust to different variations in queries that do not change its semantics?*

6.4.1 Datasets

We consider the following datasets in our experiments: `TREC-DL-2019` [68] for the passage retrieval task and `ANTIQUE` [128] for non-factoid question answering task, containing 43 and 200 queries respectively in their test sets. For each of the test set queries, we generate one query variation for each of the proposed methods, and we use the manual annotation described in this section (§6.4.4) to take into account only the valid generated query variations in our experiments. The statistics of the datasets can be found in Table 6.4.

6.4.2 Ranking Models

We use different ranking models that cover from lexical traditional models (Trad) such as `BM25`, to neural ranking models (NN) such as `KNRM` and neural ranking models that employ

[6] As available here `https://huggingface.co/ramsrigouthamg/t5_paraphraser`

[7] `https://www.kaggle.com/c/quora-question-pairs`

[8] `https://wordnet.princeton.edu/`

Table 6.4: Statistics of the TREC-DL-2019 and ANTIQUE datasets used to evaluate the robustness of query variations.

	TREC-DL-2019	ANTIQUE
#Q train	367013	2426
#Q valid	5193	-
#Q test	43	200
# terms/ Q test	5.51	10.51
# valid query variations	334	1706

transformer-based language models (TNN) such as BERT. For all of our experiments, we apply BM25 as a first-stage retriever and re-rank the top 100 results with the neural ranking models, which is an established and efficient approach [193].

For BM25 [290] and RM3 [1] we resort to the default hyperparameters and implementation provided by the PyTerrier toolkit [213]. We trained the kernel-based ranking models KNRM [369] and CKNRM [73] on the training sets of TREC-DL-2019 and ANTIQUE using default settings from the OpenNIR [208] implementation. For the BERT-based methods EPIC [210], an efficiency-focused model that encodes query and documents separately, and BERT [236], also known as monoBERT, which concatenates query and the document and makes predictions based on the [CLS] token representation, we fine-tune the bert-base-uncased model for the train datasets. For T5 [274] we use the monoT5 [237] implementation from PyTerrier T5 plugin [9] which has the pre-trained weights for MSMarco [231] by the original authors of monoT5.

6.4.3 Query Generators Implementation

As for our methods of generating query variations, for T5DescToTitle and T5QQP we rely on pre-trained T5 models (t5-base) and we fine-tune them using the Huggingface transformers library [364]. For BackTranslation we use the facebook/m2m100_418M pre-trained model from the transformers library[10]. For all other methods, we use the implementations from the TextAttack [226] library.

6.4.4 Quality of Query Generators

Given the automatic nature of the methods we introduced, we need to evaluate their quality: how good are these methods at generating query variations users would also generate?

To this end, we consider two properties of the generated queries: (I) $\hat{q}$ maintains the same semantics as q, and (II) the syntax difference between q and $\hat{q}$ can be attributed to the category C. All pairs of q and $\hat{q} = M(q)$ from the test sets of TREC-DL-2019 (43 queries) and ANTIQUE (200 queries) for each of the 10 automatic variation methods went to the following process. First, we automatically set the variations from *misspelling*[11] and *ordering* as valid since they are rule-based transformations to the input.

[9] https://github.com/terrierteam/pyterrier_t5

[10] https://huggingface.co/facebook/m2m100_418M

[11] *misspelling* methods can generate invalid queries when all words of the query are stop-words (e.g. '*how is it being you*' from ANTIQUE would generate the same query as output since there are no non-stop-words to modify)

Then all transformations that generate a variation that is identical to the input query ($\hat{q} = M(q) = q$) were automatically set to invalid. The annotators (the authors) then annotated independently the remaining 1371 pairs of $\{q, \hat{q}\}$ for the two mentioned properties (binary labels). The percentage of queries that are valid (both desired properties) are displayed at the right-most columns of Table 6.3 for the 10 automatic variation methods used in the paper and all combinations of $\{q, \hat{q}\}$ (2430).

We find the methods in the *paraphrasing* category to yield the largest percentage of invalid query variations: fewer than 38% of query variations generated via `WordNetSynSwap` are valid. A manual inspection of the invalid queries reveals the following insights:

- `T5DescToTitle` at times removes query terms that are important and thus change its semantics (e.g. '*if i had a bad breath what should i do*' ⟶ '*if i had a*').
- `BackTranslation` and `T5QQP` methods can generate an identical copy of the input query which was automatically labelled as invalid (e.g. '*what is dark energy*' ⟶ '*what is dark energy*')
- Transformations that replace words by their presumed synonyms (`WordEmbedSynSwap` and `WordNetSynSwap`) at times adds words that are not in fact synonymous in the query context (e.g. '*what is dark energy*' ⟶ '*what is blackness energy*' and '*what is a active margin*' ⟶ '*what is a active border*').

To evaluate the robustness of the ranking models, we resort to using only the valid queries as defined by the manual annotations. We have thus 2,040 valid queries for datasets `TREC-DL-2019` and `ANTIQUE` that we employ in the experiments that follow. Since some methods generate more valid variations than others, it is possible that we get better approximations of their impact on the effectiveness of retrieval pipelines.

6.5 Results

In this section we first describe our main results on the robustness of models to query variations, analyzing them by category of variation and by category of ranking model. We then move on to discussing the fusion of the ranking list obtained by the query variations.

6.5.1 Robustness to Query Variations

In order to explore the robustness of our three types of ranking models (traditional, neural, and transformer-based), we compare the effectiveness of our models when we replace the original query with the respective query variation. The results of this experiment are displayed in Table 6.5 for both the `TREC-DL-2019` and `ANTIQUE` datasets. Each row shows the effectiveness of the ranking models (columns) when using the queries obtained from each automatic query variation method. The last column (#Q) displays the number of valid queries generated by each query variation method; the invalid queries are replaced with the original ones[12].

The results show that for most of the query variations and ranker combinations, we observe a statistically significant effectiveness drop (49 out of 70 times for `TREC-DL-2019`

[12] While rows are directly comparable, methods with fewer valid queries are a lower bound of the potential decreases in effectiveness.

Table 6.5: Effectiveness (nDCG@10) of different methods for TREC-DL-2019 and ANTIQUE when faced with different query variations. Bold indicates the highest values observed for each model and ↓/↑ subscripts indicate statistically significant losses/improvements, using two-sided paired Student's T-Test at 95% confidence interval with Bonferroni correction when compared against the model with original queries. #Q is the number of valid query variations (invalid query variations are replaced by the original query).

		TREC-DL-2019							
Category	**Variation**	BM25	RM3	KNRM	CKNRM	EPIC	BERT	T5	**#Q**
-	original query	**0.480**	**0.516**	**0.502**	**0.493**	**0.624**	**0.645**	0.700	43
Misspelling	NeighbCharSwap	$0.275^{\downarrow}$	$0.275^{\downarrow}$	$0.316^{\downarrow}$	$0.309^{\downarrow}$	$0.389^{\downarrow}$	$0.416^{\downarrow}$	$0.495^{\downarrow}$	43
	RandomCharSub	$0.231^{\downarrow}$	$0.233^{\downarrow}$	$0.236^{\downarrow}$	$0.226^{\downarrow}$	$0.295^{\downarrow}$	$0.328^{\downarrow}$	$0.396^{\downarrow}$	42
	QWERTYCharSub	$0.244^{\downarrow}$	$0.250^{\downarrow}$	$0.267^{\downarrow}$	$0.297^{\downarrow}$	$0.351^{\downarrow}$	$0.387^{\downarrow}$	$0.446^{\downarrow}$	42
Naturality	RemoveStopWords	0.478	0.511	0.484	0.476	0.621	0.639	0.687	37
	T5DescToTitle	0.421	$0.434^{\downarrow}$	0.392	0.393	$0.506^{\downarrow}$	$0.536^{\downarrow}$	$0.571^{\downarrow}$	35
Ordering	RandomOrderSwap	0.480	0.516	0.502	0.471	0.623	0.635	0.697	43
Paraphrasing	BackTranslation	0.396	$0.420^{\downarrow}$	0.393	$0.361^{\downarrow}$	0.530	$0.547^{\downarrow}$	0.606	23
	T5QQP	0.472	0.504	0.454	0.461	0.605	0.640	**0.705**	26
	WordEmbedSynSwap	$0.353^{\downarrow}$	$0.354^{\downarrow}$	$0.382^{\downarrow}$	$0.368^{\downarrow}$	$0.475^{\downarrow}$	$0.472^{\downarrow}$	$0.560^{\downarrow}$	27
	WordNetSynSwap	$0.349^{\downarrow}$	$0.365^{\downarrow}$	$0.381^{\downarrow}$	$0.361^{\downarrow}$	$0.449^{\downarrow}$	$0.447^{\downarrow}$	$0.545^{\downarrow}$	16

		ANTIQUE							
Category	**Variation**	BM25	RM3	KNRM	CKNRM	EPIC	BERT	T5	**#Q**
-	original query	**0.229**	**0.217**	0.218	0.207	0.266	**0.421**	**0.334**	200
Misspelling	NeighbCharSwap	$0.156^{\downarrow}$	$0.148^{\downarrow}$	$0.159^{\downarrow}$	$0.145^{\downarrow}$	$0.184^{\downarrow}$	$0.287^{\downarrow}$	$0.251^{\downarrow}$	199
	RandomCharSub	$0.162^{\downarrow}$	$0.159^{\downarrow}$	$0.156^{\downarrow}$	$0.148^{\downarrow}$	$0.189^{\downarrow}$	$0.280^{\downarrow}$	$0.249^{\downarrow}$	182
	QWERTYCharSub	$0.161^{\downarrow}$	$0.153^{\downarrow}$	$0.160^{\downarrow}$	$0.155^{\downarrow}$	$0.192^{\downarrow}$	$0.299^{\downarrow}$	$0.266^{\downarrow}$	197
Naturality	RemoveStopWords	0.227	0.216	**0.222**	**0.215**	**0.269**	$0.383^{\downarrow}$	0.320	199
	T5DescToTitle	$0.167^{\downarrow}$	$0.165^{\downarrow}$	$0.160^{\downarrow}$	$0.167^{\downarrow}$	$0.200^{\downarrow}$	$0.270^{\downarrow}$	$0.240^{\downarrow}$	136
Ordering	RandomOrderSwap	0.229	0.217	0.218	0.198	0.267	$0.413^{\downarrow}$	$0.325^{\downarrow}$	200
Paraphrasing	BackTranslation	$0.162^{\downarrow}$	$0.155^{\downarrow}$	$0.160^{\downarrow}$	$0.144^{\downarrow}$	$0.204^{\downarrow}$	$0.305^{\downarrow}$	$0.258^{\downarrow}$	93
	T5QQP	0.220	0.207	0.210	0.196	0.261	$0.393^{\downarrow}$	0.321	105
	WordEmbedSynSwap	$0.176^{\downarrow}$	$0.172^{\downarrow}$	$0.190^{\downarrow}$	$0.169^{\downarrow}$	$0.214^{\downarrow}$	$0.325^{\downarrow}$	$0.283^{\downarrow}$	124
	WordNetSynSwap	$0.179^{\downarrow}$	$0.175^{\downarrow}$	$0.196^{\downarrow}$	$0.177^{\downarrow}$	$0.212^{\downarrow}$	$0.324^{\downarrow}$	$0.273^{\downarrow}$	71

and 54 out of 70 times for ANTIQUE), and that no set of query variations improves statistically over using the original query. If we look into the percentage of overall effectiveness decreases considering only the valid queries, we see on average that the models become 20.62% and 19.21% less effective for TREC-DL-2019 and ANTIQUE respectively. **This answers our main research question indicating that retrieval pipelines are not robust to query variations**. This confirms previous empirical evidence that query variations induce a big variability effect on different IR systems [22, 410]. We show that even with newer large-scale collections such as TREC-DL-2019, pipelines with neural ranking models are not robust to such variations.

There are several potential explanations for this drop in effectiveness besides the lack of robustness of neural rankers. The first-stage ranker may be the point of failure, being unable to retrieve sufficiently many relevant documents for the neural rankers to re-rank. It is also possible that the query variations lead to unjudged documents being ranked highly by the retrieval pipelines, which in the standard retrieval evaluation setup are considered non-relevant. We now present two experiments to show that these alternative explanations are not the cause of the drop in retrieval effectiveness.

Let's focus first on the first-stage ranker. Figure 6.1 shows the effect of increasing the re-ranking threshold on the distribution of nDCG@10 Δ when using BERT, revealing that although the number of relevant documents on the re-ranking set increases (e.g. BM25 has Recalls @10, @100 and @1000 on average of 0.06, 0.25 and 0.48 for *misspelling* query variations), BERT still struggles (negative Δ) with query variations[13]. This indicates that even if we increase the number of relevant documents in the list to be re-ranked, the re-rankers still fail when facing query variations.

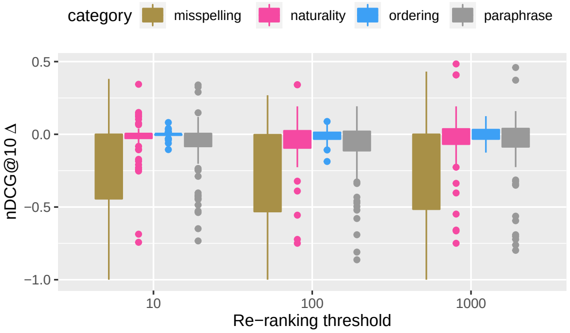


Figure 6.1: Distribution of nDCG@10 Δ for different re-ranking thresholds when using BERT as a re-ranker.

To further isolate the effect of the first-stage retrieval module, we analyzed whether the effectiveness of the pipelines would not degrade in case the first-stage retrieval was performed on the original query. In this experiment, only the re-ranker models use the query variations and we check whether the effectiveness drops persist. The results reveal that there are still statistically significant effectiveness drops when only the re-ranker models use the query variations, although in a smaller magnitude. While the drops in the effectiveness of the pipelines when using query variations for the entire pipeline are on average of 20% in nDCG@10, when using the query variations only for re-ranking they are of 9%. **This indicates that not only the first stage retrieval module is not robust to query variations, but also the neural re-rankers**.

Let's now focus on the matter of unjudged documents. It is possible that we are underestimating the effectiveness of the retrieval pipelines when facing query variations if (I) the number of unjudged documents in the top-10 ranked lists increases and (II) they turn out to be relevant. When counting the amount of judged documents in the top-10 ranked lists of the retrieval pipelines, we find that on average the number actually increases (4.30% for TREC-DL-2019 and 0.36% for ANTIQUE), **meaning that the performance drops of the retrieval pipelines can not be attributed to unjudged documents being brought up in the ranking by the query variations**.

Robustness by Query Variation Category

In order to study the effect of each query variation category, Figure 6.2 displays the nDCG@10 Δ (difference in effectiveness when replacing the original query by its varia-

[13] Similar results are obtained for other neural rankers.

tion) distribution per category and model. Although some query variations have a positive effect (points with positive Δ), the distributions are mostly skewed towards effectiveness decreases (negative Δ).

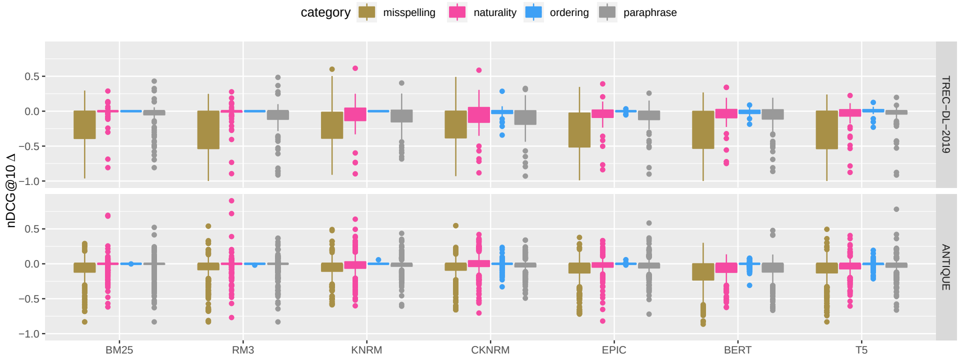


Figure 6.2: Distribution of nDCG@10 Δ when replacing the original query by the methods of each category.

First, we see that on average the decreases are higher for the *misspelling* category: -0.25 and -0.08 of nDCG@10 Δ for TREC-DL-2019 and ANTIQUE respectively. We hypothesize that the effect is higher on TREC-DL-2019 due to it having shorter queries than TREC-DL-2019 (see the average number of terms per query in Table 6.4).

The second highest effect on both datasets is the query variations from the *paraphrasing* category (-0.08 and -0.03 of nDCG@10 Δ) followed by *naturality* (-0.05 and -0.03). Compared to the *misspelling* variations which in most cases degrade the effectiveness of our models, *paraphrasing* and *naturality* have more queries for which the effect is positive, rendering the overall nDCG@10 Δ smaller.

Queries from the *ordering* category have the least effect (less than 0.01). Since traditional methods are in fact bag-of-words models, changing the word order will not have any effect on them, which makes the average of all models' nDCG@10 Δ closer to zero. In the following section, we take a further look at how each type of ranking model is affected by each query variation method.

Robustness by Model Category

When we consider how different models are affected by the query variations, we see from Figure 6.2 that with the exception of *ordering*, which has no effect on BM25, RM3 and KNRM, other transformations have a similar overall distribution of nDCG@10 Δ amongst different models. In order to understand if models (and category of models) make mistakes on the same queries, we label the models as follows: BM25 and RM3 are labeled as Trad (lexical matching), KNRM and CKNRM (neural network based) are labeled as NN and EPIC, BERT, T5 are labeled as TNN (transformer language model based). We then represent each model with the nDCG@10 Δ values obtained for each query and variation method resulting in a total of $\#Q \times \#M$ features per model. In order to visualize them we reduce this representation to 2 factors with tSNE[14] [340], as shown in Figure 6.3.

We observe that even though models have similar magnitudes and directions of nDCG@10 Δs, classes of models as indicated by color are clustered indicating that the query variations have similar effects for each type of model.

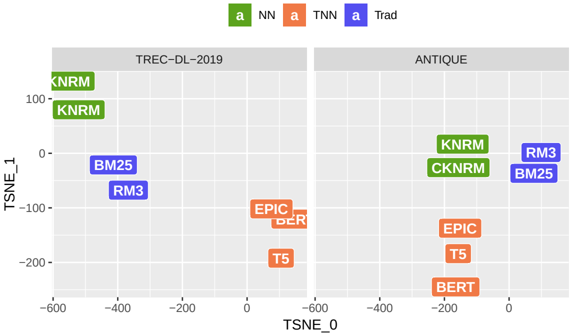


Figure 6.3: tSNE dimensionality reduction where each model is represented by the nDCG@10 Δ values obtained for each query and variation method ($\#Q \times \#M$).

While Trad models have decreases of -0.03 (TREC-DL-2019) and -0.01 (ANTIQUE) for *naturality* query variations, the effect is higher on TNN: -0.05 and -0.04 respectively. This is evidence that neural ranking models based on heavily pre-trained language models have a slight preference for natural language queries as opposed to keyword queries, which is a finding aligned with previous work [72]. Another interesting finding is that the word order does not have a great effect on TNN models (decreases smaller than 0.01). This is in line with recent research that indicates that the word order might not be as important as initially thought for transformer models [258, 306].

6.5.2 Fusing Query Variations

Although on average query variations make models less effective, there are cases when there are effectiveness gains (as shown with the positive nDCG@10 Δ in Figure 6.2). This motivates the combination of different query variations to obtain better ranking effectiveness. In order to understand whether we can improve the effectiveness of models by combining different query variations, we compare different methods for combining queries, as displayed in Table 6.6. $\texttt{RRF}_C$ indicates that we fuse the results obtained from the query variations obtained after applying M_C methods using the Reciprocal Rank Fusion (RRF) method [67], and $\texttt{RRF}_{All}$ fuses the results obtained by all query variation methods[15].

First, we see that there is potential to have significant effectiveness gains, as shown by the last line (*best query*) where we always use the query with the highest retrieval effectiveness amongst query variations and the original query. The results show that combining

[14] tSNE first calculates a probability distribution of pairs of objects in a way that similar ones (locally) have higher probability compared to dissimilar points in the high-dimensional space, then it defines a probability over the points in the low-dimensional space, minimizing the Kullback-Leibler divergence between the two distributions with respect to the locations of the points.

[15] *ordering* was not included in the experiments as a separated row since it only has one method, but it is included in the $\texttt{RRF}_{All}$ method.

Table 6.6: Effectiveness (nDCG@10) of different methods when employing rank fusion (RRF) of the rankings obtained by using different sets of queries, e.g. $\texttt{RRF}_{misspelling}$ fuses queries generated by *misspelling* methods. Bold indicates the highest values observed for each model and ↓/↑ subscripts indicate statistically significant losses/improvements, using t-test when compared against the same model with the original queries.

	TREC-DL-2019						
	BM25	RM3	KNRM	CKNRM	EPIC	BERT	T5
original query	0.479	0.515	0.501	0.493	0.624	0.644	0.699
$\texttt{RRF}_{Misspelling}$	$0.303^{\downarrow}$	$0.309^{\downarrow}$	$0.323^{\downarrow}$	$0.317^{\downarrow}$	$0.383^{\downarrow}$	$0.416^{\downarrow}$	$0.465^{\downarrow}$
$\texttt{RRF}_{Naturality}$	0.475	0.497	0.485	0.463	0.590	0.616	0.662
$\texttt{RRF}_{Paraphrasing}$	0.474	0.486	0.480	$0.433^{\downarrow}$	0.5847	0.612	0.662
$\texttt{RRF}_{All}$	0.474	0.497	0.502	0.495	$0.590^{\downarrow}$	$0.603^{\downarrow}$	$0.645^{\downarrow}$
best query	**$0.540^{\uparrow}$**	**$0.577^{\uparrow}$**	**$0.605^{\uparrow}$**	**$0.612^{\uparrow}$**	**$0.699^{\uparrow}$**	**$0.719^{\uparrow}$**	**$0.759^{\uparrow}$**
	ANTIQUE						
	BM25	RM3	KNRM	CKNRM	EPIC	BERT	T5
original query	0.228	0.216	0.217	0.206	0.266	0.421	0.333
$\texttt{RRF}_{Misspelling}$	$0.171^{\downarrow}$	$0.166^{\downarrow}$	$0.175^{\downarrow}$	$0.165^{\downarrow}$	$0.206^{\downarrow}$	$0.275^{\downarrow}$	$0.243^{\downarrow}$
$\texttt{RRF}_{Naturality}$	$0.184^{\downarrow}$	$0.186^{\downarrow}$	0.202	0.203	$0.240^{\downarrow}$	$0.317^{\downarrow}$	$0.270^{\downarrow}$
$\texttt{RRF}_{Paraphrasing}$	$0.190^{\downarrow}$	$0.184^{\downarrow}$	$0.191^{\downarrow}$	$0.176^{\downarrow}$	$0.238^{\downarrow}$	$0.339^{\downarrow}$	$0.288^{\downarrow}$
$\texttt{RRF}_{All}$	$0.199^{\downarrow}$	$0.197^{\downarrow}$	0.215	0.203	$0.243^{\downarrow}$	$0.317^{\downarrow}$	$0.272^{\downarrow}$
best query	**$0.271^{\uparrow}$**	**$0.268^{\uparrow}$**	**$0.298^{\uparrow}$**	**$0.284^{\uparrow}$**	**$0.337^{\uparrow}$**	**$0.448^{\uparrow}$**	**$0.392^{\uparrow}$**

query variations with RRF is better than using query variations individually (Table 6.5), and sometimes it is even the same as using the original query (no statistical difference). **Our results indicate that while rank fusion mitigates the decreases in effectiveness of different query variations ($\texttt{RRF}_{All}$ decreases are of 3% and 10% nDCG@10 for TREC-DL-2019 and ANTIQUE respectively when compared to the original query), it does not improve the effectiveness over using the original query**.

When are query variations better?

To better understand when models benefit from different query variations, we plot the distribution of query variations that improve over the original query by ranking model and query variation category in Figure 6.4.

We see that overall the queries obtained through the *naturality* and *paraphrasing* methods are the ones that improve over the original queries the most. Intuitively, *paraphrasing* query variations can potentially rewrite the query with better terms (e.g. '*why do criminals practice crime*' ⟶ '*why do criminals practice misdemeanour*' +0.13 nDCG@10 for BERT using WordEmbedSynSwap), make queries grammatically correct (e.g. '*how sun rises*' ⟶ '*how does the sun rise*' +0.03 nDCG@10 for BERT using T5QQP) and also corrects spelling mistakes (e.g. '*what is sosiology*' ⟶ '*what is sociology*' +0.47 nDCG@10 for BERT using BackTranslation). *naturality* methods make the queries shorter (e.g. '*who is robert gray*' ⟶ '*robert gray*' +0.34 nDCG@10 for BERT using RemoveStopWords), removing unnecessary information from the original query on certain cases.

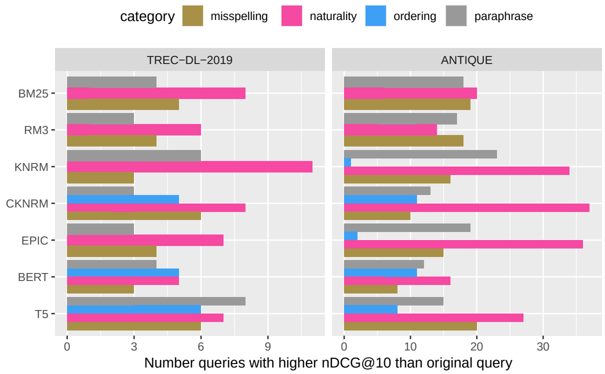


Figure 6.4: Distribution of query variations that are better than the original query.

6.6 Limitations

A limitation of the proposed methods to generate variations is that there is no guarantee that the outputs do not shift the original query in a way that modifies also the underlying information need. While we solved this problem in our study by manually going through the generated queries and checking that, this is not a scalable solution. A second point we would like to mention is that there are categories of query variations, specifically *misspelling* and *naturality*, that have a direction. For example, the transformation "add spelling errors" is different than "remove spelling errors". Removing spelling errors can be thought of as an auto-correct function that is present in most commercial search engines. The same is true when we use the *naturality* transformation to go from a natural language question to a query. Our work is limited as it did not consider two different models, one for each direction of the transformation.

A related but not covered aspect of the query variations is the auto-complete feature that most commercial search engines have. In our work, we do not consider the categories *Gen./specialization* and *Aspect change* (see Table 6.2) which are modifications to the query that are particularly interesting as auto-complete options.

Another aspect that we do not cover is language variations when dealing with full-blown conversations as opposed to initial information-seeking requests represented by the queries. Initial work has looked into query paraphrases for conversational passage retrieval [6], however, it is unknown the different types of language variations and how they occur when the interaction is a dialogue as opposed to single queries.

Finally, the set of valid query variations for some categories is small. Also, while the query variations are valid for the taxonomy proposed here it does not mean they are representative of how users actually generate such variations.

6.7 Conclusions

In this work, we studied the robustness of ranking models when faced with query variations. We first described a taxonomy of transformations between two queries for the same information need that characterizes how exactly a query is modified to arrive at one of its variants. We found six different types of transformations, and we focused our experiments on the ones that do not change the query semantics: *misspelling*, *naturality*, *ordering*, and *paraphrasing*. They account for 57% of observed variations in the UQV100 dataset.

For each of these four categories, we proposed different methods to automatically generate a query variation based on an input query. We studied the quality of the generated query variations, and based only on the valid ones we analyzed how robust retrieval pipelines are to them. Our experimental results on two different datasets quantify how much each model is affected by each type of query variation, demonstrating large effectiveness drops of 20% on average when compared to the original queries from the test sets. We found rank fusion techniques to somewhat mitigate the drops in effectiveness. Our work highlights the need of creating test collections that include query variations to better understand model effectiveness.

This chapter provides initial evidence for the third main research question of the thesis (M-RQ3). We show that language variations of users when engaging with information retrieval systems lead to degradation in the effectiveness of retrieval pipelines, both for retrieval and re-ranking. This indicates that our multi-stage retrieval pipeline for conversational search studied in the first part of this thesis needs to be improved in terms of robustness to language variations.

Considering that transformer-based language models are used throughout the entire multi-stage pipeline, we evaluate next which conversational search and recommendation capabilities they have, in order to provide further evidence regarding M-RQ3.

7

Evaluating Transformers with Conversational Recommendation Tasks

In this chapter, we continue to explore the limitations of multi-stage retrieval pipelines. Given that pre-trained transformer models are ubiquitous in such pipelines, from retrieval to re-ranking, we explore here their limitations for conversational recommendation tasks. Given that such models implicitly store factual knowledge in their parameters after pre-training, understanding this step is crucial for using and improving them for conversational recommendation models. We study how much off-the-shelf pre-trained BERT "knows" about recommendation items such as books, movies, and music. In order to analyze the knowledge stored in BERT's parameters, we use different probes (i.e., tasks to examine a trained model regarding certain properties) that require different types of knowledge to solve, namely content-based and collaborative-based. Content-based knowledge is the one that requires the model to match the titles of items with their content information, such as descriptions and genres. In contrast, collaborative-based knowledge requires the model to match items with similar ones, according to interactions such as ratings. We resort to BERT's Masked Language Modelling (MLM) head to probe it about the genre of items, with cloze style prompts. In addition, we employ BERT's Next Sentence Prediction (NSP) head and representations' similarity (SIM) to compare relevant and non-relevant search and recommendation query-document inputs to explore whether it can, without any fine-tuning, rank relevant items first. Finally, we study how BERT performs in a conversational recommendation downstream task. To this end, we fine-tune BERT to act as a retrieval-based CRS. The code required to reproduce this chapter is available at https://github.com/Guzpenha/ConvRecProbingBERT.

This chapter is based on the following paper:

Gustavo Penha and Claudia Hauff. 2020. What Does BERT Know About Books, Movies and Music? Probing BERT for Conversational Recommendation. RecSys. 388–397 [250].

7.1 Introduction

One important breakthrough in Natural Language Processing (NLP) is the use of heavily pre-trained transformers for language modeling, such as BERT [80] or T5 [274]. These pre-trained Language Models (LMs) are extremely powerful for many downstream tasks in NLP as well as IR, Recommender Systems, Dialogue Systems, and other fields—and have thus become an essential part of our machine learning pipelines. One advantage of these models is their capability to perform well on specific tasks and domains (that were *not* part of their training regime) via fine-tuning, i.e. the retraining of a pre-trained model with just a few thousand labeled task- and/or domain-specific examples. Besides the power of such models to model human language, they have also been shown to store factual knowledge in their parameters [257, 287]. For instance, we can extract the fact that the famous Dutch painter Rembrandt Harmenszoon van Rijn died in Amsterdam by feeding the prompt sentence *"Rembrandt died in the city of ____"* to a pre-trained LM[1], and use the token with the highest prediction score as the chosen answer.

Given the prevalence of such heavily pre-trained LMs for transfer learning in NLP tasks [267, 324], it is important to understand what the pre-training objectives are able to learn, and also what they fail to learn. Understanding the representations learned by such models has been an active research field, where the goal is to try and understand what aspects of language such models capture. Examples include analyzing the attention heads [61, 222], or using probing tasks [146, 324] that show which linguistic information is encoded. Such LMs have been successfully applied to different IR tasks [270, 297, 376, 377], but it is still unknown what exactly makes them so powerful in IR [49]. Unlike previous studies, we diagnose LMs here from the perspective of conversational recommendations. We focus on BERT [80] as its publicly released pre-trained models have been shown to be effective in a wide variety of NLP and IR tasks.

Thus, our first research question (**RQ1**) is: *How much knowledge do off-the-shelf BERT models store in their parameters about items to recommend?* We look specifically at movies, books, and music due to their popularity, since many users frequently engage with recommenders in such domains. Indeed, some of the largest existent commercial recommender systems such as Netflix, Spotify, and Amazon focus on the aforementioned domains.

In order to provide a better intuition of our work, consider the examples in Table 7.1. Shown are examples (for the movie domain) of inputs and outputs for the different tasks considered in our work. In conversational recommendation, users engage in a conversation with the system to obtain recommendations that satisfy their current information needs. This is the downstream task we focus on in this chapter. The users often describe items that they have interacted with and enjoyed (*"Power Rangers in 1995 and then Turbo in 1997"*), and give textual descriptions of what they are looking for regarding the recommendation (*"film with great soundtrack"* and *"dramas, thrillers"*). Such interactions can be categorized as having the intent of providing preferences [144]. We consider the knowledge of which items are often consumed together to be *collaborative-based knowledge*, and we examine models for this through a recommendation probing task: *given an item, find similar ones* (according to the community interaction data such as ratings from ML25M [127]), e.g. users who like *"Power Rangers"* also like *"Pulp Fiction"*. We consider

[1] This specific example works with both `bert-large-cased` and `roberta-large` in the fill-mask pipeline from the transformers library `https://huggingface.co/transformers/pretrained_models.html`.

the descriptions about the content of the items to be *content-based knowledge*, and we examine models for this using a search probing task for which a review of the item has to be matched with the title of the item, and a genre probing task for which the genres of the movie have to be matched with the movie title.

Table 7.1: Input and output examples for the probing and downstream tasks considered in the movie domain. For the first task, **recommendation**, the user input is the history of seen movies, and the output is the recommendation for what to watch next. This task requires a model to match movies that are often seen together by different users—and thus are similar in a collaborative sense. We refer to this as collaborative-based knowledge. The second task, **search**, requires that a model matches descriptions of the item (item review) with the title. Similarly, the **genre** requires the model to match the genres of the items with their titles. We refer to this type of knowledge described in the second column as content-based. In **conversational recommendation** (the downstream task we focus on here), we see that knowing that “*Pulp Fiction*” is a movie often seen by people who saw “*Power Rangers*” (recommendation probe), that it has a good soundtrack (search probe), and that it is from the genres “*drama*” and “*thriller*” (genre probe) are helpful information to give a credible and accurate response.

	Recommendation	**Search and Genre**	**Conversational Recommendation**
User input	Critters (1986) → NevErEnding Story, The (1984) → Power Rangers (1995) → Turbo: A Power Rangers Movie (1997)→	**search** ”[...] and there’s the music in the movie: the songs Tarantino chose for his masterpiece fit their respective scenes so perfectly that most of those pieces of music.” **genre** ”drama, thriller”	”90’s film with great soundtrack.[...] I thought Power Rangers in 1995 and then Turbo in 1997 were masterpieces of cinema, mind you [..] I’m looking for movies from that era with great music. Dramas, thrillers, road movies, adventure... Any genre (except too much romantic) will do .”
System output	Pulp Fiction (1994)	Pulp Fiction (1994)	You should see Pulp Fiction, Rock Star, [...]
Task type	probing	probing	downstream
Knowledge	collaborative	content	content and collaborative

To answer **RQ1**, we probe BERT models on *content-based knowledge*, by using the predictions of BERT’s Masked Language Modelling (MLM) head. We use knowledge sources to extract the information of the genre of the items, and generate prompt sentences such as *”Pulp Fiction is a movie of the* ____ *genre.”* similar to prior works [257], for which the tokens *drama, thriller* should have high prediction scores in case the BERT model stores this information. In order to probe BERT models for the search and recommendation probing tasks, we introduce two techniques that do not require fine-tuning and are able to estimate the match between two sentences. One technique is based on BERT’s Sentence Representation Similarity (SIM), while the other is based on BERT’s Next Sentence Prediction (NSP) head. We generate the relevant recommendation prompt sentences with items that are frequently consumed together and use both techniques to compare them against the non-relevant ones with items that are rarely consumed together. For example, the prompt *“If you liked Pulp Fiction* `[SEP]` *you will also like Reservoir Dogs“*[2] should have a higher next sentence prediction score than the input *“If you liked Pulp Fiction* `[SEP]` *you will also like*

[2] Note that the `[SEP]` token is used by BERT as sentence separator, and we, therefore, use the next sentence predictor head as a next *subsentence* predictor head.

To All the Boys I've Loved Before", since the first two movies co-occur more often than the second pair based on rating data such as MovieLens [127]. For the search prompt, we generate relevant sentences by matching the title of the items with their respective reviews, a common approach to simulate product search [122, 386].

Our experimental results for **RQ1** reveal the following:

- BERT has both collaborative-based and content-based knowledge stored in its parameters; correct genres are within the top-5 predicted tokens in 30% to 50% of the cases depending on the domain; reviews are matched to correct items 80% of the times in the book domain when having two candidates; correct recommendation sentences are selected around 60% of the time when having two candidates.
- BERT is more effective at storing content-based knowledge than collaborative-based knowledge as shown by our probing experiments.
- The NSP is an important pre-training objective for the search and recommendation probing tasks, improving the effectiveness over not using it up to 58%.
- BERT's effectiveness for search and recommendation probes drops considerably when increasing the number of candidates in the probes, especially for collaborative-based knowledge (i.e., a 35% decrease in the recall at the first position).

Based on these findings, we next study how to use BERT for conversational recommendation and, more importantly, manners to infuse collaborative-based knowledge and content-based knowledge into BERT models as a step towards better CRS. We hypothesize that a model which is able to perform well at search and recommendation probing tasks is better for conversational recommendation. And thus, our second research question (**RQ2**) is: *What is an effective manner to infuse additional knowledge for conversational recommendation into BERT?* Our experimental results show the following.

- Our fine-tuned BERT is highly effective in distinguishing relevant responses and nonrelevant responses, yielding significant improvements when compared to a competitive baseline for the downstream task.
- When faced with adversarially generated negative candidates with random items, BERT's effectiveness degrades significantly (from 0.78 to 0.07 MRR).
- Infusing content-based and collaborative-based knowledge via multi-task learning during the fine-tuning procedure improves conversational recommendation.

7.2 Related Work

The extensive success of pre-trained transformer-based language models such as BERT [80], RoBERTa [201][3], and T5 [274] can be attributed to the transformers' computational efficiency, the amount of pre-training data, the large amount of computations used to train such models[4] and the ease of adapting them to downstream tasks via fine-tuning. Given

[3] RoBERTa is similar to BERT but it is trained for longer on more data, and without the NSP pre-training task.

[4] For instance, the RoBERTa model [201] was trained on 160GB of text using 1,024 32GB *NVIDIA V100* GPUs

the remarkable success of such LMs, pioneered by BERT, researchers have focused on understanding what exactly such LMs learn during pre-training. For instance, by analyzing the attention heads [61, 222], by using probing tasks [146, 324] that examine BERT's representation to understand which linguistic information is encoded at which layer and by using diagnostic datasets [49].

BERT and RoBERTa failed completely on 4 out of the 8 probing tasks that require reasoning skills in experiments conducted by Talmor et al. [315]. The "Always-Never" probing task is an example of such a failure. Here, prompt sentences look like *"rhinoceros [MASK] have fur"*, with candidate answers for this task being *"never"* or *"always"*. Petroni et al. [257] showed that BERT can be used as a competitive model for extracting factual knowledge, by feeding cloze-style prompts to the model and extracting predictions for its vocabulary. Jiang et al. [151] extended this work, demonstrating that using better prompt sentences through paraphrases and mined templates led to better extraction of knowledge from LMs. Roberts et al. [287] showed that off-the-shelf (i.e., pre-trained LMs without fine-tuning) T5 outperformed competitive baselines for open-domain question answering. More recently, with the uptake in model size and pre-training time, we see improvements across multiple different tasks, and the effectiveness of such models when using zero-shot prompts is getting closer to the effectiveness of fine-tuned models [25, 172, 266][5].

Another line of work has focused on infusing different information in LM parameters to perform better at downstream tasks. One approach to do so is by having intermediary tasks before the fine-tuning on the downstream task [259]. The intuition here is that other tasks that are similar to the downstream task could improve the LM's effectiveness. It is still unknown why a combination of intermediate and downstream tasks is effective [263]. A similar approach is to continue the pre-training of the language model with domain-specific text corpora [123]. Wang et al. [351] proposed a different approach inspired by multi-task learning [397] that grouped similar NLP tasks together. When infusing different types of knowledge into LMs, it is possible for some of the knowledge that was stored in its parameters to be erased, otherwise known as catastrophic forgetting [169]. Thompson et al. [329] proposed a technique that regularizes the model when doing adaptation so that the weights are close to the pre-trained model. Wang et al. [354] tackled this problem by proposing adapters, i.e., auxiliary neural modules that have different sets of weights, instead of sharing weights in a multi-task manner—and are effective when infusing different types of knowledge into LMs (such as factual and linguistic).

Instead of probing LMs for linguistic properties or general facts, we examine LMs in our work through the lens of conversational recommendation. Specifically, we look into recommendation, search, and genre probes that require collaborative and content knowledge regarding items to be recommended. We then examine the effectiveness of the LMs for conversational recommendation—before and after infusing additional knowledge via multi-task learning.

[5] Given the closed nature of commercial models, such as ChatGPT and PaLM, it is difficult to evaluate what the model has seen and what the model has not seen during pre-training, and thus what is in fact zero-shot.

Table 7.2: Examples of the probes used in this paper. We use off-the-shelf BERT's Masked Language Modelling (MLM) head for predicting tokens, BERT's Next Sentence Prediction (NSP) head for predicting if the underlined sentence is the most likely continuation of the sentence, and BERT's last layer hidden representations (CLS pooled and MEAN pooled) for calculating the similarity between two texts (SIM). All probes require no fine-tuning, and thus indicate what BERT learns through its pre-training objectives. The knowledge source for recommendation prompts are interaction datasets, such as users' movie ratings. For search prompts, we use items' review data. No underline indicates sentences that are treated as the query, and underline indicates sentences that are treated as the document. Relevant documents for a query have label 1, e.g. document *you will also like Lord of the Rings* for the query *If you liked The Hobbit*, while non-relevant have label 0, e.g. document *you will also like Twilight* for the query *If you liked The Hobbit.*

Type	**Prediction**	**Task**	**Prompt Examples**	**Labels**
MLM	**Token**	**Genre**	TP-NoTitle: *"It is a movie of the [MASK] genre."* TP-Title: *"Pulp Fiction is a [MASK] movie."* TP-TitleGenre: *"Pulp Fiction is a movie of the [MASK] genre ."*	crime crime crime
			TP-NoTitle: *"It is a book of the [MASK] genre."* TP-Title: *"Palestine by Joe Sacco is a [MASK] book."* TP-TitleGenre: *"Palestine by Joe Sacco is a book of the [MASK] genre."*	comic comic comic
SIM	**IsSimilar**	**Recommendation**	*{("The Hobbit", "Lord of the Rings"),* *("The Hobbit", "Twilight")}*	{1, 0}
		Search	*{("The book is not about the murder [...]", "The Brothers Karamazov"),* *("It gives a brilliant picture of three bright young people [...]", "The Brothers Karamazov.") }*	{1, 0}
NSP	**IsNext**	**Recommendation**	*{"If you liked The Hobbit,* `[SEP]` *you will also like Lord of the Rings",* *"If You liked The Hobbit,* `[SEP]` *you will also like Twilight"}*	{1, 0}
		Search	*{"The book is not about the murder [...]* `[SEP]` *The Brothers Karamazov. ",* *" It gives a brilliant picture of three bright young people [...]* `[SEP]` *The Brothers Karamazov. "}*	{1, 0}

7.3 Method

In this section, we introduce our three types of probing tasks (genre, search, and recommendation). We then turn to our downstream task—conversational recommendation.

7.3.1 Genre Probes

We resort to genre (i.e. a style or category of the item such as *comedy*) probes to extract and quantify the knowledge stored in language models about the recommended items. Using knowledge sources that contain an item's title and its respective genres, e.g. "*Los miserables by Victor Hugo*" ⟶ *"romance, fiction, history"*[6], we create prompt sentences for each item with the genre as the masked token. Since we use the MLM head to make predictions, we refer to this probing as `MLM`. We use three manually defined prompt sentence templates (cf. Table 7.2, first row, for examples of each template type) inspired by [151] for the `MLM` probe to investigate what BERT can do with different templates:

- **TP-NoTitle**: we do not provide the item, only the domain of the item.
- **TP-Title**: we use both the title of the item and its domain.
- **TP-TitleGenre**: we provide the item title, domain, and additional phrase *"of the genre"* indicating that we are looking specifically for the genre of the item.

The underlying assumption of this probing technique is that if the correct tokens are ranked higher by the language model, it has this knowledge stored in its parameters about

[6] We can extract this information from user-generated tags to books for example.

the item. We evaluate the amount of knowledge stored in the model by counting the number of correctly ranked labels as the most probable in the first and first 5 positions, i.e. recall@1 and recall@5. Since the template sentences are not exhaustive, our manually selected templates offer only a lower bound on the amount of knowledge stored in the language model.

7.3.2 Recommendation and Search Probes

In order to probe a LM's capacity to rank relevant items in recommendation and search scenarios we now introduce two probing techniques (SIM and NSP). Like the genre probe, these two techniques do not require any fine-tuning to quantify the LM's ranking effectiveness. We were inspired by methods to calculate the matching degree between two sentences, in a non-supervised way [395]. While SIM uses the representations directly to calculate the matching degree, NSP relies on the fact that this pre-training BERT head was designed to understand the relationship between the two sentences, something not directly captured by the MLM training [80].

Using both techniques, we compare prompt sentences (the template and prompt generation are explained shortly) that represent either a 'query' or a 'document'. The query sentences take input from the user side (for search this is the item description, and for recommendation this is the history of rated items), and the document sentences contain a possible answer from the system to this input (the item to be recommended). We refer to relevant document sentences as the ones that are relevant items for the query sentence. Non-relevant document sentences are randomly sampled.

Probe Based on Similarity (SIM)

SIM ranks document sentences for a query sentence based on the representations learned by the LM: we calculate the dot similarity between the query sentence and document sentences using either the [CLS] token representation (SIMCLS), or the average pooling of all tokens (SIMMEAN). More formally:

$$SIM_{CLS} = BERT_{CLS}(query\_sentence) \cdot BERT_{CLS}(document\_sentence)$$

where $BERT_{CLS}(s)$ means the representation of the CLS token in the last layer, and

$$SIM_{MEAN} = BERT_{MEAN}(query\_sentence) \cdot BERT_{MEAN}(document\_sentence)$$

where $BERT_{MEAN}(s)$ means extracting the representations of each token in the last layer by taking the average.

Probe Based on Next Sentence Prediction Head (NSP)

NSP ranks document sentences for a query sentence based on the likelihood of the document sentence being the next sentence for the query sentence. Stated formally:

$$NSP = BERT_{NSP}(query\_sentence \mid \texttt{[SEP]} \mid document\_sentence)$$

where | indicates the string concatenation operator. This probe technique also does not require any fine-tuning of BERT.

Templates and Prompt Generation

Having defined our probing techniques, we now discuss how to generate the prompts for the recommendation and search probes, along with the templates. Based on the knowledge extracted from rating and review datasets, we create *prompt sentences* in a similar manner to how previous work extracted knowledge from other data sources [256, 257].

For the recommendation probe, the query sentence is built using an item that was rated by a user u, and the relevant document sentence is another item rated by u as well. The non-relevant document sentences are items that were not rated by u, and are sampled based on the item's popularity. Since we do not have access to negative feedback on items, we use a common assumption in the offline evaluation of recommender systems that a randomly sampled item is not relevant [28]. The assumption for the recommendation & search probes is that a model that has higher similarity between the query sentence and the relevant document sentence has knowledge regarding which items are consumed together. For instance, see the `SIM` recommendation example in Table 7.2—a successful collaborative-filtering recommender system would display a higher similarity between *"The Hobbit"* and *"Lord of the Rings"* (items extracted from the user ratings' history) than the similarity between *"The Hobbit"* and *"Twilight"* (an item not relevant to the given user). Conversely, for the `NSP` probes, we expect the next sentence prediction from the relevant document sentence to be higher than the non-relevant ones. Using the same user as an example, the next sentence prediction score for the relevant query-document sentence *"If you liked The Hobbit* `[SEP]`*, you will also like Lord of the Rings"* should be higher than the non-relevant sentence *"If You liked The Hobbit* `[SEP]`*, you will also like Twilight"*.

For the search probe, the query sentence is built using entire reviews from the items, whereas the relevant document sentence is the title of the item for which the review was written[7]. The non-relevant document sentences are reviews of randomly sampled items. We use review data to simulate product search inspired by previous works [5, 122, 341, 386]. For instance, we expect that the `SIM` and `NSP` scores between the item *"The Brothers Karamazov"* and its review text *"The book is not about the murder [...]"* to be higher than the scores between the item and a randomly sampled review.

7.3.3 Infusing Knowledge into LMs for Conversational Recommendation

Finally, we discuss our downstream task, i.e. the task we aim to solve better with knowledge gained from our probes. Let us first define how to use BERT as an end-to-end retrieval-based conversational recommender system by formally defining the problem, before discussing the infusion of knowledge into a pre-trained language model.

Conversational Recommendation

We treat the conversational recommendation task as finding the best response in a set of candidates as defined in Section 2.6.1. This formulation abstracts away the task of finding the specific item to be recommended and considers that the existing responses contain the relevant items to be recommended as part of their text. Another option for approaching conversational recommendation is to find the specific item IDs as a response to the

[7] We remove the titles of the items from the reviews to make the task more challenging.

dialogue context, which requires entities to be identified and linked in the training and evaluation datasets [51, 183].

To fine-tune BERT for the task we follow the procedure described in 2.7.2 and predict relevance as follows:

$$f(\mathcal{U}_i, r) = FFN(BERT_{CLS}(u^1 \mid \texttt{[SEP]} \mid u^2 \mid ... \mid u^\tau \mid \texttt{[SEP]} \mid r)),$$

where | indicates the concatenation operation[8] and FFN is a linear layer. We train it for binary classification using cross entropy as the loss function.

Infusing Knowledge into LMs

In order to infuse content-based and collaborative-based knowledge into BERT, we resort to multi-task learning [397]. In addition to fine-tuning BERT for the conversational data, we also consider interleaving batches of different tasks. $BERT_{rec}$ interleaves training instances of the conversational recommendation task, with the recommendation NSP probing task. Analogously $BERT_{search}$ interleaves the downstream task with search NSP.

Multi-task learning is challenging as the order of the tasks [255] and the weighting [163] for each task have a large impact on the model's quality; we leave such analyses as future work and resort to equal weights and interleaved batches.

This way, half of the time the inputs to BERT are

$$\{u^1 \mid \texttt{[SEP]} \mid u^2 \mid ... \mid u^\tau \mid \texttt{[SEP]} \mid r\},$$

the concatenation of the dialogue context and the candidate response, and the other half of the inputs to BERT are

$$\{query\_sentence \mid \texttt{[SEP]} \mid document\_sentence\},$$

the concatenation of the query sentences and the candidate document sentences from the search and recommendation probes as defined in Section 7.3.2. The labels are 1 when the input is a relevant query and document pair and 0 otherwise.

7.4 Experimental Setup

We first discuss our data sources and then point out important implementation details.

7.4.1 Data Sources

We use English language data[9] from three different domains in order to generate the templates for our probes:

- Books: we use the publicly available GoodReads[10] [348] data with over 200M interactions from the GoodReads community. We extract ratings, reviews, and genres.

[8] An example of input sequence for BERT is: *"Good morning, I am looking for a good classic today.* [SEP] *What type of movie are you looking for today?* [SEP] *I enjoyed Annie (1982)* [SEP] *okay no problem. If you enjoyed Annie then you will love You've Got Mail (1998)"*

[9] The data we created for this work, as well as all our code, are publicly available at https://github.com/Guzpenha/ConvRecProbingBERT.

[10] https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home

- `Movies`: we use the publicly available ML25M[11] [127] dataset that contains 25M interactions from the MovieLens community. We extract ratings and genres. Since ML25M does not have any review data, we crawled reviews for movies that were rated in ML25M from IMDB. We collected a maximum of 20 reviews for each movie from the ML25M data. This resulted in a total of 860k reviews (av. length of 84.22 words) and an average of 13.87 reviews per movie.

- `Music`: We use the *"CDs and Vinyl"* subset of the publicly available Amazon reviews[12] [232] dataset which contains 2.3m interactions. We extract ratings, reviews, and genres for music albums.

For all the probes in this paper (genre, search, and recommendation) we generate 100k instances, with the exception of movies in the genre probing task for which we have access to only approximately 60k movies (the number of movies in the ML25M dataset). For the genre probing task, we have on average 3.6, 1.8, and 1.4 genres for the books, movies, and music domains and a total of 16, 20, and 284 distinct genres respectively.

Inspired by previous work that uses online forums as a source of rich conversational data [246, 268], we extract conversational recommendation data for the three domains from reddit forums: */r/booksuggestions*, */r/moviesuggestions* and */r/musicsuggestions*[13] on March 17, 2020. They include multi-turn conversations where an information-seeker is looking for recommendations, and an information provider gives suggestions through natural language conversations.[14]

Additionally, we use the ReDial dataset [186] which was collected using crowd workers and includes dialogues of users seeking and providing recommendations in the movies domain. We use this dataset due to the annotated movie identifiers that are mentioned in each utterance, which is not available for the Reddit data. This allows us to create adversarial examples (see Table 7.9 for a concrete example) that require the model to reason about different items to be recommended, while the rest of the response remains the same. The statistics of the data used for conversational recommendation are shown in Table 7.3. For the music domain, there is a limited number of conversations available (the *musicsuggestions* subreddit has only 10k users, compared to the 292k users of the *booksuggestions* subreddit). ReDial has relatively few words in the responses.

For all dialogue datasets, we generate 50 candidate responses for every context by querying all available responses using BM25 [288] using the context as a query. This is the re-ranking setup described in Section 2.6.1, with conversational recommendation datasets instead of general information-seeking dialogues.

7.4.2 Implementation Details

We use the BERT and RoBERTa PyTorch transformers implementations[15]. When fine-tuning BERT for conversational recommendation, we employ a balanced number of relevant and non-relevant context and response pairs. We resort to BERT's default hyperpa-

[11] `https://grouplens.org/datasets/movielens/25m/`

[12] `https://nijianmo.github.io/amazon/index.html`

[13] `https://www.reddit.com/r/booksuggestions/`, `https://www.reddit.com/r/moviesuggestions/` and `https://www.reddit.com/r/musicsuggestions/`

[14] See the conversational recommendation example from Table 7.1 which comes from this dataset.

[15] `https://github.com/huggingface/transformers`

rameters, and use the *large* cased models; we fine-tune them with the Adam optimizer [168] with a learning rate of $5e-6$ and $\epsilon = 1e-8$. We employ early stopping using the validation nDCG. For the conversational recommendation task, we also employ as baselines traditional IR methods: QL[16] [261], and QL with RM3 [180]. We use the pyserini[17] implementation of QL and RM3 and use the context as query and candidate responses as candidate documents. In addition, we compare BERT against strong neural baselines for the task: DAM [402][18], and MSN [384][19], which are interaction-based methods that learn interactions between the utterances in the context and the response with attention and multi-hop selectors, respectively. We fine-tune the hyperparameters for the baseline models (QL, RM3, DAM, and MSN) using the validation set.

Table 7.3: Statistics of the conversational recommendation datasets. We use dialogues extracted from three subreddits: */r/booksuggestions*; */r/moviesuggestions*; and */r/musicsuggestions*. We also experiment with ReDial [186] due to its exact matches with movies.

	Books	Movies	Music	ReDial
# $\mathcal{U}-r$ **pairs**	157k	173k	2k	61k
# candidates per $\mathcal{U}$	50	50	50	50
Avg # turns	1.11	1.08	1.11	3.54
Avg # words per u	103.37	124.93	74.17	71.11
Avg # words per r	40.10	23.39	38.84	12.58

7.5 Results

In this section, we first discuss the results of the probes for genre, followed by the probes for search and recommendation. We then analyze how BERT performs in our downstream task of conversational recommendation.

7.5.1 Probing BERT

In this subsection, we first analyze the results of the genre probes, followed by the search and recommendation probes.

Genres

The results for probing BERT for each item's genre (100k books and music albums and 62k movies) are displayed in Table 7.4. We show the recall at positions 1 and 5 (number of relevant tokens in the first and first 5 predictions divided by the total number of relevant genres). To provide the reader with intuition, we provide example prompts and predictions in Table 7.5. First, we note that by just using the domain of the item, and not an item's title (TP-NoTitle templates), BERT can already retrieve a reasonable amount of tokens related to the genre in the first five positions (from 25% to 41% depending on the domain) which is high given that the vocabulary contains 29k tokens. We see examples of this in Table 7.5,

[16] We experimented with BM25 as well and kept QL due to it achieving better results.

[17] https://github.com/castorini/pyserini/

[18] https://github.com/baidu/Dialogue/tree/master/DAM

[19] https://github.com/chunyuanY/Dialogue

Table 7.4: Results for *BERT* genre MLM probe. Bold indicates a statistically significant difference over all other sentence types using a paired t-test with a confidence level of 0.95 and Bonferroni correction.

	Genre probes					
	Books		Movies		Music	
Template	**R@1**	**R@5**	**R@1**	**R@5**	**R@1**	**R@5**
TP-NoTitle	0.067	0.259	0.067	0.395	0.074	0.412
TP-Title	0.031	0.119	0.058	0.258	0.139	0.346
TP-TitleGenre	**0.109**	**0.296**	**0.179**	**0.505**	**0.156**	0.412

where for instance BERT predicts *fantasy* if you ask for a book genre and *pop* if you ask for an album genre. This result shows that the pre-trained model indeed contains information regarding which genres are specific to each domain.

When we consider the template types where we inform BERT about the item's title (TP-Title and TP-TitleGenre), we see that there is knowledge about specific items stored in BERT's parameters, as the results of TP-TitleGenre are better than TP-NoTitle, with improvements from 0.067 to 0.179 R@1. **We can thus answer RQ1 partially: BERT has content knowledge about items stored in its parameter, specifically regarding their genres.** From a total of 29k tokens it can find the correct genre token up to 50% of the times in the first 5 positions using TP-TitleGenre.

We also note that a prompt with more specific information leads to better results (from TP-Title to TP-TitleGenre for instance), and this is only a lower bound for the knowledge stored since some information might be stored in BERT that we could have retrieved with a different prompt template sentence. For example, if we do not indicate in the prompt that we are looking for the genres of the items (TP-Title), we get tokens that can describe the item but are not genres. For example, for the prompt *"The Wind-Up Bird Chronicle by Haruki Murakami is a _____ book."* we get the token *japanese*, (cf. Table 7.5), which is valid since the author is Japanese, but it is not the correct answer for the genre probe task. In some cases TP-Title retrieves the publication year of the item, e.g. *"1990 book"*.

Search and Recommendation

The results of the recommendation and search probes are shown in Tables 7.6 and 7.7 respectively. We show the recall at 1 with 2 and 5 candidates $R_2@1$ & $R_5@1$ (we resort to using different numbers of candidates here, due to the candidates being sentences and not tokens like the genre probing task). We see that using both SIM and NSP techniques BERT retrieves better than the random baseline (being equal to the random baseline would mean that there is no such information stored in BERT's parameters). **This answers RQ1: BERT has content-knowledge and collaborative-knowledge about items stored in its parameter.** Using the NSP technique BERT matches items with their respective reviews 82%, 67% and 75% of the times for the books, movies, and music domains when choosing between two options. Also, BERT selects the most appropriate item to match a user history (recommendation probe) 65% of the time when choosing between two options.

Regarding the technique to probe BERT with, NSP is the most effective, showing that this pre-training objective is indeed crucial for tasks that require relationships between

Table 7.5: Examples of *BERT* predictions for each of the domains when probing it with the MLM head for item genres. Bold indicates a correct prediction. BERT is able to match domains with common genres (TP-NoTitle template), e.g. books with fantasy and music with rock. Prompt sentences that indicates to BERT it is looking for the genre of items (TP-TitleGenre as opposed to TP-Title) yields better predictions as they avoid general descriptions, e.g. "*television, 2003, japanese*".

	Sentence	Genre Prompt	Predicted (top 2)
Books	TP-NoTitle	It is a book of the genre _____.	**fantasy** [0.18], romance [0.13]
	TP-Title	The Wind-Up Bird Chronicle is a _____ book.	comic [0.07], japanese [0.04]
	TP-TitleGenre	The Wind-Up Bird Chronicle is a book of the genre _____.	**fantasy** [0.60], horror [0.04]
Movies	TP-NoTitle	It is a movie of the genre _____.	horror [0.08], **action** [0.05]
	TP-Title	I, Robot (2004) is a _____ movie.	tv [0.16], television [0.16]
	TP-TitleGenre	I, Robot (2004) is a movie of the genre _____.	robot [0.54], horror [0.08]
Music	TP-NoTitle	It is a music album of the genre _____.	pop [0.09], **rock** [0.07]
	TP-Title	Tom Petty: Greatest Hits is a _____ music album.	country [0.09], 2003 [0.08]
	TP-TitleGenre	Tom Petty: Greatest Hits is a music album of the genre _____.	**rock** [0.73], country [0.10]

Table 7.6: Results for the recommendation probes using SIM-based and NSP-based approaches. Bold means statistical significance compared to baselines (paired t-tests with Bonferroni correction and confidence level of 0.95). NSP-based probes are the most effective for all three datasets.

		Recommendation probes					
		Books		Movies		Music	
Technique	**Model**	**R_2@1**	**R_5@1**	**R_2@1**	**R_5@1**	**R_2@1**	**R_5@1**
-	Random	0.500	0.200	0.500	0.200	0.500	0.200
SIM_{CLS}	BERT	0.538	0.252	0.525	0.230	0.537	0.254
SIM_{CLS}	RoBERTa	0.574	0.291	0.509	0.219	0.550	0.267
SIM_{MEAN}	BERT	0.601	0.331	0.525	0.232	0.583	0.295
SIM_{MEAN}	RoBERTa	0.518	0.230	0.497	0.205	0.534	0.243
NSP	BERT	**0.651**	**0.402**	**0.653**	**0.367**	**0.610**	**0.333**

sentences. Although RoBERTa uses a similar framework to BERT, it has more parameters (340M ⟶ 355M), and it is trained on more data (16GB ⟶ 160GB of text) for longer (100K ⟶ 500K steps). BERT is still more effective than RoBERTa, when we employ the NSP head. We note that during the training phase of RoBERTa the NSP pre-training objective was not employed as for NLP downstream tasks no gains were observed [201].

We see that BERT has about 17% more content-based knowledge than collaborative-based knowledge considering the results from our probes. We hypothesize that this is due to textual descriptions of items with content information (useful for search) being more common than comparative sentences between different items (useful for recommendation) in the data used for BERT's pre-training. We also note in Figure 7.1 that when increasing the number of candidates (x-axis), the effectiveness of the recommendation probe degrades more than for the search probes. This means that for a downstream task, BERT would have to be employed as a re-ranker for only a few candidates.

Table 7.7: Results for the search probes using SIM-based and NSP-based approaches. Bold indicates statistical significance compared to all baselines (paired t-tests with Bonferroni correction and confidence level of 0.95). BERT stores more content-based knowledge (search, this table) than collaborative-based knowledge (recommendation, Table 7.6). NSP-based probes are the most effective for all three datasets.

		Search probes					
		Books		Movies		Music	
Technique	**Model**	**R_2@1**	**R_5@1**	**R_2@1**	**R_5@1**	**R_2@1**	**R_5@1**
-	Random	0.500	0.200	0.500	0.200	0.500	0.200
SIM_{CLS}	BERT	0.495	0.198	0.387	0.123	0.498	0.200
SIM_{CLS}	RoBERTa	0.578	0.255	0.516	0.229	0.527	0.215
SIM_{MEAN}	BERT	0.612	0.338	0.523	0.235	0.579	0.314
SIM_{MEAN}	RoBERTa	0.548	0.225	0.476	0.208	0.492	0.192
NSP	BERT	**0.825**	**0.636**	**0.670**	**0.420**	**0.755**	**0.537**

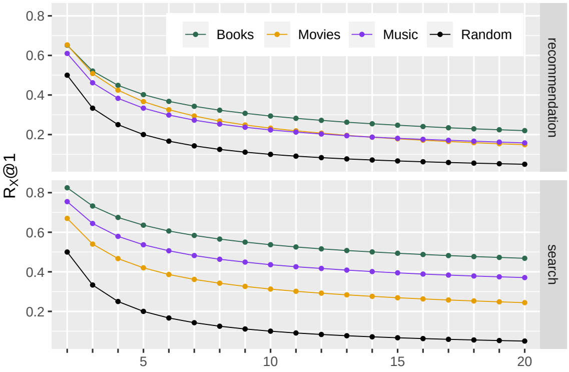


Figure 7.1: BERT effectiveness (R_x@1) for NSP probes when increasing the number of candidates to rank x.

When comparing different domains, the highest observed effectiveness when probing BERT for search is for books. We hypothesize this to be due to one of BERT's pre-training data being the BookCorpus [404]. Since the review data used for the search probe often include mentions of book content, the overlap between both data sources is probably high. We are unable to verify this directly because the BookCorpus dataset is not publicly available anymore.

7.5.2 Infusing Knowledge for Conversational Recommendation

Table 7.8 shows the results of fine-tuning BERT for the conversational recommendation task on the three domains using our Reddit forum data. Standard IR baselines, QL, and QL with RM3 performed poorly on this task (≈0.05 MRR). We hypothesize this happens due to the recommendation nature of the underlying task in the conversation. For example, a user that describes its previously liked items does not want to receive answers with the same items being recommended in it (which are highly ranked by QL) but new item titles that have semantic similarity with the conversational context. The deep models (DMN and MSN) that learn semantic interactions between utterances and responses on the other hand perform better than traditional IR methods (up to 0.79 MRR), MSN being the best non-BERT approach. BERT is powerful at this task (up to 0.93 MRR), with statistically significant improvements for books, movies, and music when compared to MSN.

Table 7.8: Results for the conversational recommendation task. We provide the MRR, with the respective standard deviation (for 5 runs). Bold indicates statistical significance compared to all baselines (paired t-tests with Bonferroni correction and confidence level of 0.95). Fine-tuned BERT is remarkably effective for retrieving relevant answers in conversations containing recommendations when sampling 50 negative candidates with BM25.

	/r/booksuggestions	/r/moviessuggestions	/r/musicsuggestions
QL	0.055 (.00)	0.048 (.00)	0.061 (.00)
RM3	0.051 (.00)	0.046 (.00)	0.049 (.00)
DAM	0.610 (.02)	0.662 (.02)	0.208 (.04)
MSN	0.707 (.01)	0.788 (.02)	0.535 (.06)
BERT	**0.886** (.01)	**0.929** (.00)	**0.620** (.03)

To investigate why BERT is so successful at this task, we resort to the ReDial dataset. Specifically, we create adversarial response candidates for the responses that included a recommendation. This is possible because, unlike our Reddit-based corpus, ReDial has additional labels indicating which item from ML25M was recommended at each utterance. For every relevant response containing a recommendation, we generate adversarial candidates by changing only the relevant item with randomly selected items, see Table 7.9 for some examples. This way, we can evaluate whether BERT is only picking up linguistic cues of what makes a natural response to a dialogue context or if it is using collaborative knowledge to retrieve relevant items to recommend.

The results for the adversarial dataset are displayed in Table 7.10. BERT's effectiveness drops significantly (from 0.78 to 0.07 MRR) when we test using the adversarial version of ReDial. Previous works have also been able to generate adversarial examples that fool BERT on different NLP tasks [152, 314].

Failing on the adversarial data shows that BERT is not able to successfully distinguish relevant items from non-relevant items, and is only using linguistic cues to find relevant answers. This motivates infusing additional knowledge into BERT, besides fine-tuning it for the conversational recommendation task. In order to do so, we do multi-task learning for the probe tasks in order to infuse additional content-based ($BERT_{search}$) and collaborative-based ($BERT_{rec}$) knowledge into BERT using only probes for items that are mentioned in the training conversations.

Our results in Table 7.10 show that the recommendation probe improves BERT by 9% for the adversarial dataset $ReDial_{Adv}$, while the search probe improves effectiveness on $ReDial_{BM25}$ by 1%. This indicates that the adversarial dataset indeed requires more collaborative-based knowledge. The approach of multi-task learning for infusing knowledge into BERT was not successful for our Reddit-based forum data. We hypothesize that this happened because, unlike ReDial, we have no additional labels indicating which items were mentioned in the reddit conversations. This forces us to train on probes for items

Table 7.9: Examples of the ReDial dataset for conversational recommendation using either BM25 to sample negative candidates ($ReDial_{BM25}$) or the adversarial generation that replaces the movies from the relevant response with random movies ($ReDial_{Adv}$) but keeps the context. The adversarial candidates requires BERT to be able to chose between different movies, while for the BM25 candidates BERT can use language cues to select the correct response—likely text given the context.

Context	**Relevant response**	**Negative BM25 candidate** ($ReDial_{BM25}$)	**Negative adversarial candidate** ($ReDial_{Adv}$)
Good morning, I am looking for a good classic today. [SEP] What type of movie are you looking for today? [SEP] I enjoyed Annie (1982)	okay no problem.. If you enjoyed Annie then you will love You've Got Mail (1998) !	I am great! What type of movie are you looking for today?	okay no problem.. If you enjoyed Annie then you will love The Best Years of Our Lives (1946) !
HI! [SEP] Hi what type of movie would you like? [SEP] I am looking for something like Star Wars (1977) but not Star Trek	Have you seen Avatar (2009)	I love Star Trek Generations (1994) the best!	Have you seen Wishmaster (1997),

Table 7.10: Fine-tuned BERT results (MRR) for conversational recommendation for the dataset when using different procedures to sample negative candidates. Bold indicates statistical significance compared to other approaches (paired t-tests with Bonferroni correction and confidence level of 0.95). *BERT* is the model fine-tuned on ReDial, $BERT_{rec}$ multi-tasks between fine-tuning for ReDial and for the recommendation probes and $BERT_{rec}$ multi-tasks between fine-tuning for ReDial and for the search probes.

	$ReDial_{BM25}$	$ReDial_{Adv}$
BERT	0.778 (.01)	0.069 (.02)
BERT_{rec}	0.780 (.00)	**0.073 (.01)**
BERT_{search}	**0.791 (.01)**	0.072 (.02)

that are likely not going to be useful. We leave the study of automatically identifying mentions of items in conversations as future work.

Answering our second research question **(RQ2), we demonstrate that infusing knowledge from the probing tasks into BERT, via multi-task learning during the fine-tuning procedure is an effective technique**.

7.6 Limitations

Given the fast pace of research in language models, our findings might not hold for recent models with significantly higher amounts of parameters. While the biggest BERT model has 340M parameters, GPT-3 has 175B, and PaLM [60] has 540B. Whether simply scaling up such language models solves the limitations we found is still an open question.

The language models tested in this chapter are not yet applicable in realistic conversational recommendation scenarios. In the first research question of this chapter, we analyze what was already stored in the weights of language models during pre-training through simple probes, which do not necessarily translate to realistic scenarios of using such models for delivering recommendations. In the second section, we analyze how to infuse such knowledge and see that in more complicated scenarios (e.g. adversarial recommendations) language models still have a long way to go. Our study is also limited in the experimental setup, as it does not put such language models in contact with dynamic dialogues with real users, and is limited to English corpora.

7.7 Conclusions

Given the potential that heavily pre-trained language models offer for conversational recommender systems, we examine how much knowledge is stored in BERT's parameters regarding books, movies, and music. We resort to different probes in order to answer this question. We find that we can use BERT to extract the genre for 30-50% of the items on the top 5 predictions, depending on the domain; and that BERT has about 17% more content-based knowledge (search) than collaborative-based knowledge (recommendation).

Based on the findings of our probing task we investigate a retrieval-based approach based on BERT for conversational recommendation, and how to infuse knowledge into its parameters. We show that BERT is powerful for distinguishing relevant from non-relevant responses. By using adversarial data, we demonstrate that BERT is less effective when it has to distinguish candidate responses that are reasonable responses but include randomly selected item recommendations. This motivates infusing collaborative-based and content-based knowledge into BERT, which we do via multi-task learning during the fine-tuning step, obtaining effectiveness improvements of up to 9%.

This chapter provides further evidence for the third main research question of the thesis (M-RQ3). We show that transformer-based language models have only a limited knowledge about entities such as movies, books and fail in recommendation problems that require collaborative-filtering. This indicates that retrieval and re-ranking systems for conversational search require better ways to infuse or combine relationships between entities that is commonly used for recommendation.

V

Conclusions

8 Conclusions

In this chapter, we summarize the thesis by revisiting the main research questions that were introduced and our main findings. Then we discuss its limitations, followed by a discussion on ethical concerns and wider implications of conversational search systems, and finish with a discussion on future directions in the field of conversational search.

8.1 Summary

In this section, we revisit the main findings of this thesis guided by our main research questions stated in the introduction. They are:

> **M-RQ1**: What is a strong baseline for the retrieval, i.e. first-stage, of responses for conversational search? Do the findings of passage and document retrieval tasks translate to the retrieval of responses for dialogues?
>
> **M-RQ2**: Do different notions of difficulty improve the re-ranking, i.e. second-stage, of responses for conversational search?
>
> **M-RQ3**: What are the limitations of transformer-based models for conversational search and recommendation?

8.1.1 First-stage Retrieval

In order to answer **M-RQ1**, in Chapter 3 we compare major techniques from four different categories of models that are capable of performing first-stage retrieval: unsupervised and supervised sparse retrieval, zero-shot and fine-tuned dense retrieval. While such models were initially proposed for document and passage retrieval tasks, we show that most of the findings hold when we go to the conversational search domain. Specifically, we show that a pipeline to obtain a dense retriever composed of (1) self-supervised pre-training, (2) intermediate representation learning, and (3) a final fine-tuning on target conversational data with hard negative samples is the best-performing approach. Such a dense model is able to significantly outperform a supervised sparse baseline based on document augmentation. Our results indicate that findings from other tasks such as passage retrieval do generalize to retrieval for conversational search. Our study also reveals that there is room to improve such models when adapting them to deal with conversations. For example, we show that a better way to adapt doc2query [238] is to predict only the last utterance of the dialogue context (the query in our domain), as they are typically longer than queries in the document retrieval task.

8.1.2 Difficulty Notions for Re-ranking

In order to answer **M-RQ2**, we analyze two different ways difficulty notions can be implemented in order to improve conversational search systems, in Chapters 4 and 5.

We first rely on a machine learning technique known as curriculum learning. This technique expects a difficulty estimation for each training instance from the dataset. Based on this estimate, first, the model will receive easy instances, and the hard ones will be seen in the later stages of the training procedure. In Chapter 4 we compare a number of ways to estimate the difficulty of training instances of conversational search models. We find that the better-performing difficulty estimate is to use the difference between the predictions for the relevant response for the dialogue context and the negative response for the dialogue context. Applying curriculum learning to the training of neural re-rankers for conversational search is an effective way to consider the difficulty of instances during training. Subsequent work[1] in IR [211] shows that curriculum learning is also effective for passage retrieval tasks.

Similar ideas to estimate the difficulty of instances have been proposed after the publication of Chapter 4 to improve ranking models with different techniques other than curriculum learning, such as the distillation of scores with a margin MSE loss [135], residual-based margin [107], hard negative sampling [392]. This is further evidence that indeed using notions of difficulty can be used to improve ranking models during training time.

As for dealing with difficult instances at test time, we provide in Chapter 5 ways to estimate the uncertainty of the predictions of neural ranking models, and how to take them into account to obtain better models. We do so with stochastic rankers. Instead of predicting a single value that tells if a response is relevant for a dialogue context, stochastic rankers output a relevance distribution. With such distribution, we can measure how spread the predictions are, i.e. their variance, indicating the level of uncertainty for the dialogue context and response at hand. In order to improve the effectiveness of conversational search systems, we use such estimates to take into account how risky, i.e. level of uncertainty, the response is. A risk-aware ranker takes into account both the relevance prediction as well as the uncertainty related to it. We show in Chapter 5 that a risk-aware re-ranker is particularly effective when dealing with test conditions that have distribution shifts compared to the train conditions. Subsequent work[2] in IR [63] showed that a risk-aware re-ranker is also effective for passage retrieval tasks. This is further evidence that using notions of difficulty can also be used to improve ranking models during test time.

8.1.3 Retrievers and Rankers Limitations and Behavior

Finally, we provide two studies to better understand the limitations of conversational search and recommendation systems for answering **M-RQ3**, in Chapters 6 and 7. In the

[1] The paper that originated Chapter 4 was available online in December 2019.

[2] The paper that originated Chapter 5 was available online in January 2021.

first study, we analyze the impact query variations have on retrieval pipelines. A query variation of a query is another way to express the same underlying information need. The assumption we base this study on is that if a query and a query variation express the same information need, the retrieval pipeline should also behave the same. In order to study the effect of different types of variations when expressing an information need, we first define a taxonomy of query variations. In Chapter 6 we propose different techniques to generate a query variation for a given query and chosen category of variation. We then quantify the effect of each category of query variation on the effectiveness of retrieval pipelines. We find that retrieval pipelines are not robust to such query variations, with significant drops in effectiveness.

Our results relate to a broader research direction which shows that neural networks struggle with inputs that are somehow different from the training due to distribution shifts. In comparison with adversarial examples that are created with the goal of tricking the model [111], query variations are phenomena observed by users that interact with the system. Thus dealing with them is a crucial problem for existing pipelines, and can be an obstacle to the implementation of conversational search systems.

In Chapter 7 we test what type of knowledge is stored in the weights of pre-trained language models. A better understanding of the pre-training procedure is important for being able to take advantage of them and improving them for conversational search and recommendation. We focus on entities from the music, book, and movie domains. Through different probes, i.e. tasks to examine a trained model for certain properties, we evaluate search, genre, and conversational recommendation capabilities. Our findings indicate that models such as BERT are able to answer questions about the content of entities (such as finding the correct genre of a book) to a certain extent. However, they have little collaborative filtering capacities, e.g. knowing which movies are typically watched together.

In conversational recommendation, where there are well-defined entities and attributes, language models still lack important capabilities. Sileo et al. [305] extended our analysis and showed that GPT-2 models perform better than BERT for recommendation tasks based solely on the knowledge stored in their pre-training weights. However, they are still outperformed by simple recommendation baselines when there is enough training data.

8.2 Limitations

We would like to acknowledge a number of limitations that runs through the entire thesis. First, all experiments were performed using corpora in a high-resource language (English), and thus the findings might not generalize to other languages. Additionally, although we have datasets containing dialogues from multiple domains, they are a finite set. Thus the findings of our thesis and the effectiveness of the models in truly open-domain scenarios might differ. The main information-seeking dialogue datasets used in this thesis were extracted from online forums, which is a specific and non-comprehensive way people interact with other users to find information—long descriptive initial utterances, and asynchronous dialogue. However, this might not be how people will interact with conversational search agents.

There are other limitations of the experimental setup employed as described in Section 2.6.1 that challenge how realistic offline tasks used to evaluate retrieval-based chatbots are. A few of them we do not look into in this thesis, namely the creation and maintenance of a pool of responses[3], the fact that test instances from the same dialogue are considered independent and that there is only one adequate answer. The implications are that an effective model for such offline tasks might not perform well when users interact with them. Online experiments, albeit expensive, offer a solution to some of the limitations we just described.

8.3 Ethical Concerns and Wider Implications

The adoption of conversational systems for search may have major implications for how we deal with the information overload problem. Users already act as if search engines provide testimony, acquiring and altering beliefs on the basis of results the model has ranked on top [229]. Direct answers in search engines can further reduce the cognitive load required to go through documents as "*the answer given, and not others, is the one to be taken seriously*" [262], even in unwarranted cases such as complex and controversial topics. Conversational interfaces can further increase the trust of users in information systems, by using anthropomorphic design cues that lead to the appearance of human-likeness [14]. This raises a significant question: How to ensure that conversational search systems are not harmful to the information literacy of users? Users should be able to understand that there are several sources of information being retrieved and that the agent is less of a domain expert and more of a librarian. A conversational search system should not "*deposit knowledge*" in the user but engage in a truly interactive dialogue.

A significant concern is that research with increasingly larger language models requires significant financial investments, incurring environmental costs due to the costs of the training procedures [30]. This is also effectively a barrier to where such research is done and who is able to do so. While techniques to compress [84], prune [355], and distill knowledge [187] allow the usage of smaller models by a larger number of researchers, they are not a substitute for training large models from scratch. A number of capabilities seem to emerge only when training models with a large number of weights [356], and that pruning from the beginning does not lead to the same accuracy[4].

Another concern is that the gap between users that create pieces of trustworthy information and the user accessing this information can increase. This might demotivate content creation, as there would be no credit or economic value given to the original content creator. The capabilities of current large language models of creating plausible content might further aggravate this problem as anyone can create and disseminate disinformation through social media and messaging apps, for example, to influence elections [284].

Finally, a critical challenge is to guarantee that conversational search models are not harmful, for example by propagating bias against marginalized groups. The large datasets used to pre-train language models contain a number of problematic patterns such as abusive language, hate speech, gender and race bias, dehumanization and etc [188, 359]. Without mechanisms to detect and control them, the systems will propagate them.

[3] We would like to highlight again that generative approaches could be used to create such a pool of responses.

[4] Frankle and Carbin [95] introduced the lottery ticket hypothesis: randomly initialized neural networks have *winning tickets*, i.e. sub-networks with a particular set of initial weights that reach the same accuracy of the neural network when trained in isolation. Understanding the circumstances when one can effectively train sparse networks is an active area of research that could lead to the democratization of the field.

8.4 Future Directions

Based on the discussion of the findings of this thesis we provide a number of areas for future work that follow directly from the research questions of this thesis: first-stage retrieval, improving difficulty estimations and their applications, and understanding ranking models. Then we discuss broader directions for future work.

8.4.1 Directions Related to the Main Research Questions

Improving First-stage Retrieval

The quality of a pipeline for conversational search is heavily dependent on the first-stage retrieval step. If there are no relevant responses in the responses retrieved, later re-ranking stages will not be effective. Also, there is a chance there is no relevant response in the entire pool of responses, and thus a challenging problem is detecting such cases. Handling such failures is an area that requires further investigation: a system should not give a response that seems plausible when there is no valid answer, and users might not be satisfied with uninformative *"I am unable to answer this request."*. How to maneuver conversations in a direction the system is able to answer is an open area of research.

In the realm of dense retrieval, approaches that take into account multiple dense vectors have shown to be more effective than single vector approaches [165, 185, 264]. This has great potential to improve conversational search, as the input has structure, i.e. different utterances, and more than one speaker (the seeker and the provider)[5], which could be directly used for different representation spaces.

Multi-vector approaches however suffer from the problem of increased computational cost and increased space to store indexes. Another concern with dense methods for first stage-retrieval is that in practice the pool of responses is constantly changing, with novel responses for example. Also, model updates can be expensive, as a naive approach requires calculating the embeddings of the whole collection again—intelligently updating embeddings is a useful direction of research.

The field of supervised sparse retrieval is also quickly developing [71, 94, 105, 191, 216]. Although we provided initial evidence in this thesis on their usage for retrieval, the effectiveness of more complex sparse methods, that perform weighting and expansion for both the dialogue context and the responses, is still unknown.

Another possible venue for research is to combine retrieval with generation methods. While generative approaches to conversational search have limitations we laid out in the introduction, they can be used in conjunction with retrieval. How to combine generated answers in a retrieval pipeline without getting for example hallucinations and incorrect responses is still an open research direction. Such a combination could be done for instance when there is no direct answer in the corpus for the request; to generate a direct answer from a document that was retrieved; to combine multiple responses into one.

There are also other practical problems that prevent the implementation of conversational search systems. Transformer models have high complexity regarding the size of the input sentence—$O(N^2)$ where N is the number of input tokens. Since they are the backbone of many techniques for retrieval and ranking, it is a challenge how to adapt them to

[5] The development of conversational search systems that handle more than two speakers is also a developing direction known as conversational collaborative search [17]. This field is the intersection of conversational search and collaborative search [225, 300].

deal with long conversations—for example, BERT accepts a maximum of 512 input tokens. How to model the context of the conversation without using the whole dialogue as input? Transformers that deal with long sequences [29, 170, 321, 385] are incipient and have shown limited success. Approaches that model the entire history of the dialogue will be required to guarantee that the agent is able to remember past utterances in the dialogue.

Additionally, given that collections of responses will be constantly growing, e.g. Stack-Exchange receives over 400 new questions and answers per hour on average[6], learned models for sparse and dense retrieval will require smart ways to perform continual learning and also update their indexes when new responses arrive.

Estimation of Difficulty and its Applications for Ranking

There are a number of ways to take advantage of difficulty estimates for improving ranking models. With an accurate prediction of the difficulty of a dialogue, a conversational search pipeline can, for example, decide to ask a clarification question[7] or to present the results [9]. It can also be used as a feature to classify if there is no valid answer in the pool of responses [92, 251]—a very difficult dialogue might indicate there is no answer.

We explored two techniques in this thesis that consider difficulty estimations, curriculum learning, and risk-aware ranking. Negative sampling is another technique that can benefit from difficulty estimates. Negative candidate responses found using random sampling lead to easy and uninformative training instances. Harder negatives have been shown to improve the effectiveness of ranking models in a number of domains [184, 207, 291]. Approaches to finding negative samples typically deal with model-based difficulty estimates, for example using the ranking model itself [370]. This is likely due to what is difficult depending on what the model has learned at some point in training. Thus, we believe more sophisticated sampling procedures to find negative samples, that considers the model is not static and thus the notion of difficulty evolves during training, are a promising direction for research.

Another related research direction is to use the prior knowledge of an instance difficulty in the loss function. For example, we showed [252] that using a curriculum learning approach for introducing label smoothing in the loss function improved the effectiveness of ranking models. Hofstätter et al. [136] considered a notion of difficulty (the margin of a teacher model) to balance easy and difficult instances of training batches and also used it as the supervision signal in a knowledge distillation setup. The most advantageous way to use difficulty notions as inductive biases for ranking models remains an open question—for example as part of the loss function, as part of the order of the training instances, as part of the negative sampling procedure, or as part of a risk-aware re-ranking strategy.

Better Understanding of Ranking Models

An active area of research in natural language processing is devoted to finding the limitations and understanding and explaining black box neural models [76, 182, 408]. One of the concerns which might prevent the adoption of conversational systems is the lack

[6] Statistics of new questions and answers obtained from `https://sostats.github.io/` on 22-12-2022.

[7] According to Braslavski et al. [39] clarification questions can be used to ask for more information, check a fact, try to reason about the request, ask for more general details, filter and narrow down a specific aspect, and ask for past experience details. In a different taxonomy focused on search systems, Zamani et al. [388] categorized clarification questions into disambiguation, preference elicitation, topic narrowing, and comparison-based.

of robustness and understanding of when and why they fail. This is thus a crucial and nascent field for conversational search systems. We provided evidence that rankers are not robust to query variations. We hypothesize that a model which is able to align the representations of equivalent queries and equivalent documents will improve their robustness. In data augmentation, the model is simply given additional training instances to learn such equivalences, e.g. different but equivalent queries are matched against the same document. While more complex approaches have been proposed to align equivalent queries in the embedding space [57, 407], further work is necessary as identifying which instances are equivalent for the different types of query variations is a hard and unsolved problem. Additionally, creating datasets with query variations remains an open challenge, as models that do so automatically are prone to shifts in the information need and noise.

A particularly promising way to reason about representations is through the lens of disentangled learning [58, 132, 142]. With disentangled representations, the underlying assumption is that the model would benefit from separating (disentangling) the underlying structure of the input into disjoint parts of its representation. Such representation would allow us to model transformations such as query variations, which have an effect on form factors of the representation but that do not affect the factor representing the underlying information need. Another benefit of such representations is that they are interpretable. With a disentangled representation we could calculate a similarity score between a query and a document in terms of different aspects, going beyond a single number to describe their similarity. Initial work [196] has applied this idea to recommender systems, in order to provide relevant items with respect to different aspects.

Besides robustness, another important future work direction is to understand ranking model behavior, its potential biases, and weaknesses. Understanding ranking models' behavior is still an incipient field of research [49, 209, 265, 283]. In the particular domain of conversational search, this is a crucial and under-explored task, due to potential risks of employing language models [30, 299]. This research direction is closely related to the field of explainability. Some open research questions are: How to explain a response from a conversational search system? Why has the model ranked/generated such a response, where did this information come from? How to increase trust and other potential objectives [330], such as persuasiveness and scrutability, an explanation can have?

8.4.2 Broader Directions

Challenges in Generative Approaches

At the time of writing of this conclusion[8], OpenAI released a new language model for dialogue: ChatGPT[9]. It is a sibling model to InstructGPT [241], which improves over GPT-3 by taking into account human feedback to generate outputs to prompts and also to rank different outputs from the model. Another key difference between ChatGPT and GPT-3 is that it is able to generate answers in a dialogue, as opposed to one-shot answers given to prompts. ChatGPT reached one million users in five days. Users have already found it useful as a tool for learning to code—a case where you can check the correctness of the answers it provides—and as a way to surf through reading material while asking

[8] This chapter was written in December 2022.

[9] `https://openai.com/blog/chatgpt/`

questions[10]. Other users claim it to be helpful in scenarios that require creativity such as brainstorming ideas and presenting information.

Enthusiasts claim that ChatGPT will replace Google search entirely[11] because it answers questions more directly and clearly than search engines. However, a careful analysis of its answers reveals that it is often incorrect, making plenty of mistakes[12], all while sounding like reasonable and plausible answers[13]. This has led for example StackOverflow to ban ChatGPT answers[14], *"because the average rate of getting correct answers from ChatGPT is too low, the posting of answers created by ChatGPT is substantially harmful to the site and to users who are asking or looking for correct answers".*

The debate around fully generative models replacing search is not new [262, 299]. Language models certainly will be a crucial component of conversational search systems. However, we believe that using them as fully generative models will not be possible to solve the entire conversational search problem, and that retrieval and ranking components have to be part of the equation. As discussed in the introduction, generative models are capable of generating convincing responses with untrue facts—in Figure 1.5 we show how it recommends software that is not available for the operating system that the user asked for. This can be harmful in a number of domains, e.g. in health-related searches. Also, there are a number of ethical concerns which are exacerbated by generative models (see Section 8.3).

Evaluation Challenges

The evaluation of conversational search is a complex problem, where applying traditional search evaluation paradigms, i.e. the Cranfield paradigm [62, 345], is not straightforward [96, 202]. In a conversation, there are an exponential number of paths that a dialogue can evolve to, depending on which utterances are chosen by the information seeker and the system as shown in Figure 8.1. An observed dialogue, as highlighted in pastel yellow, from human-to-human or human-to-machine interaction provides us with only a single path among all possible options. When using this dialogue as ground truth to compare models repeatedly, we miss "*conterfactual*" [154] paths—what would have happened if the left response was given instead of the right response at the initial turn of the dialogue in Figure 8.1? One direction to improve offline evaluation of conversational search and offering a remedy for such problem is through user simulation [23, 197, 298]. A challenge is to obtain a realistic model that correlates with human interactions while being efficient to run repeatedly.

[10] The tweet accessible at `https://twitter.com/yacineMTB/status/1599618855273664515` contains the following: "*The new way to learn: Wikipedia on the left, chatGPT on the right. You can surf through material so ridiculously fast while relating it to things you already know. It's actually speedrunning knowledge uptake. And this is only a non-special purpose v1. Incredible!*"

[11] The tweet accessible at `https://twitter.com/jdjkelly/status/1598021488795586561` contains a thread with following initial tweet: "*Google is done. Compare the quality of these responses*" followed by a number of screenshots of the Google search engine response and the ChatGPT response to the same requests.

[12] In the following blog post `https://vitalik.eth.limo/general/2022/12/06/gpt3.html` the author shows how many mistakes ChatGPT makes when helping him solve a coding problem.

[13] In the following blog post `https://aisnakeoil.substack.com/p/chatgpt-is-a-bullshit-generator-but` the author argues that ChatGPT output texts that are intended to persuade without regard for the truth.

[14] `https://meta.stackoverflow.com/questions/421831/temporary-policy-chatgpt-is-banned`

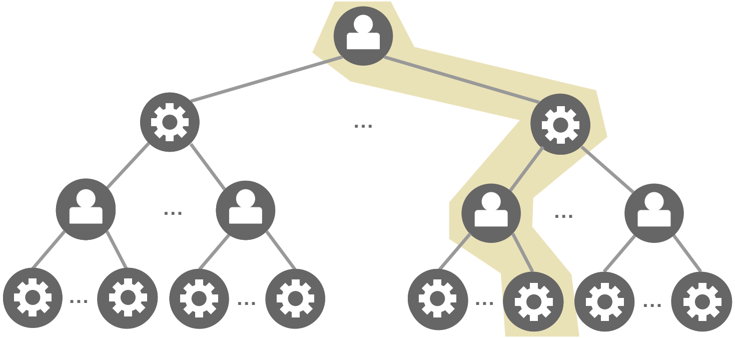


Figure 8.1: Different paths a dialogue might take, depending on which answer is given by the information-seeker and the system. In pastel yellow, we see a single dialogue that we might observe and is typically how systems are evaluated. This overlooks all other paths the dialogue could have taken.

Additionally, the focus of the NLP field and also IR has been on English-speaking users. We lack multi-lingual datasets and also datasets for specific domains, e.g. scholar searches, medical searches, etc. Large-scale human conversation data is expensive to create, and mapping out different paths of dialogue increases this cost exponentially. While public benchmarks are helpful in advancing the field, they overlook the fact that their collections are static. In reality, the pool of responses will evolve, new content will be added to it, and we need resources to be able to evaluate the effect of content evolution, e.g. Is there a point where answers become outdated and should not be retrieved?

A more complete offline evaluation of a conversational search system would also test different dimensions of the user experience [13, 161], e.g. trust, cognitive load, effort, utility, etc, and would not treat each utterance in a single observed dialogue independently [89]. Given that the majority of research in the field evaluates only small modules or tasks, improving existing evaluation schemes is a key factor to develop better conversational search systems.

In fact, a significant step that needs to be taken for conversational search adoption is to move from purely offline evaluation to online evaluation. User studies are really scarce in this domain and mostly use the wizard-of-oz setup[15], partially due to the difficulty in creating practical end-to-end conversational search systems for testing. Given the intrinsic interactive nature of conversational information access, we claim user studies will be essential for the adoption and development of the field.

Interaction Challenges

Radlinski and Craswell [273] argued that a conversational search system should display five properties when interacting with users (see Section 2.2): user revealment, system revealment, mixed-initiative, memory, and set retrieval. Six years have passed since the publication of his article, which has proven to be influential as researchers have indeed explored such objectives. For example, in order to achieve user revealment, i.e. the capac-

[15] A wizard-of-oz experiment is when subjects interact with a system that they believe to be fully automated, but there is actually a human behind it.

ity of helping the user express and discover their true information need, researchers have focused on the problem of asking follow-up and clarification questions [8, 9, 298], that are capable of eliciting users information needs. In conversational recommender systems, this elicitation process has also received attention [141, 272].

Clarification questions are also helpful to achieve mixed-initiative interactions, as the initiative that is typically dictated by the user is taken by the system. There are still open questions that need progress in this area: when to ask for clarification, how to model the ambiguity of the user request and the uncertainty of the system, how to generate/rank clarification questions, and what objective the question has. Another under-explored aspect of mixed-initiative is the system starting a conversation instead of the user. The conversational agent might recommend an item to the user based on contextual information, such as an online event based on the time and date. Important questions that still need to be addressed are how to detect when, why, and how to start such conversations [347]. Despite recent developments [7, 221, 298, 338], enabling truly mixed-initiative conversations is still an open challenge in the field.

Another aspect of the interaction that has received little attention is how to reveal to the user what the system is able to achieve, the reach of its corpus, setting expectations, and thus having the capability of performing system revealment. The capacity of exploring and understanding the corpus is related to the field of exploratory search [244, 362]. Exploration and investigation could potentially occur through multiple conversations with the system, which could help the user in finding and analyzing what is available.

Memory is another crucial aspect of conversational search systems that is still unsolved. An agent should be able to relate to the history of interactions when considering a single conversation and across different conversations the user had with the system. This includes for example creating a long-term profile of preferences, understanding the level of expertise of the user for a certain topic, references to past statements made, and so on. As previously pointed out (see Section 8.4.1), a simple approach to concatenate previous interactions with the system is not viable for transformer-based architectures, which constitute the backbone of solutions to many different tasks related to conversational search. How to model long conversations and previous interactions is still an open question.

Finally, there is still work required to better understand how to present information to users in conversational search. The modality (voice, text, image), the device, and the way to present information in such settings are important factors to be considered when delivering responses. Some open questions are: How to transition between devices? What scenario invites which modality? What is the best length of the response when a certain modality is used?

Design Challenges

In Chapter 2 we lay out a number of tasks from different research fields that can contribute to the implementation of a conversational search system. Such components still need to be put together to build a functional system, which raises questions such as how to go from the evaluation of certain tasks to the entire system in real-world settings, and how to better integrate different components in a functional system.

A conversational search system that works in practice needs to be constantly evolving. New conversations and documents should be used to update existing models continually.

Besides the costs attached to updating embeddings (see Section 8.4.1), a naive approach that continues the training procedure can lead to problems such as forgetting previous knowledge that was already learned, i.e. catastrophic forgetting [203]. Additionally, since approaches use a static dataset to train and evaluate models, there is a risk of shifts in the distribution when the system is engaging with real users that have dynamic tasks and settings. Conversations are not predominant in large natural language datasets, which exacerbates such out-of-domain scenarios. Since there is a scarcity of dialogue data compared to unlabeled natural language datasets[16] used to train large language models, an important direction of research is to reduce the dependency on supervised data.

[16] For example, `https://huggingface.co/datasets` has 297 datasets for language-modeling, including C4, a 305GB dataset which is based on a web crawl. Whereas it contains only one conversational dataset. Accessed on 23-12-2022

Bibliography

References

[1] Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. *Computer Science Department Faculty Publication Series* (2004).

[2] Taiga Abe, Estefany Kelly Buchanan, Geoff Pleiss, Richard Zemel, and John P Cunningham. 2022. Deep Ensembles Work, But Are They Necessary? *NeurIPS* (2022), 33646–33660.

[3] Charu C Aggarwal et al. 2016. *Recommender Systems*. Springer.

[4] Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive Multi-task Representations with Pre-finetuning. *arXiv preprint arXiv:2101.11038* (2021).

[5] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a Hierarchical Embedding Model for Personalized Product Search. In *SIGIR*. 645–654.

[6] Haya Al-Thani, Bernard J Jansen, and Tamer Elsayed. 2023. ECAsT: A Large Dataset for Conversational Search and an Evaluation of Metric Robustness. *PeerJ Computer Science* (2023).

[7] Mohammad Aliannejadi, Leif Azzopardi, Hamed Zamani, Evangelos Kanoulas, Paul Thomas, and Nick Craswell. 2021. Analysing Mixed Initiatives and Search Strategies During Conversational Search. In *CIKM*. 16–26.

[8] Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2020. ConvAI3: Generating Clarifying Questions for Open-domain Dialogue Systems (ClariQ). *arXiv preprint arXiv:2009.11352* (2020).

[9] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-seeking Conversations. In *SIGIR*. 475–484.

[10] James F Allen, Donna K Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. 2001. Toward Conversational Human-computer Interaction. *AI magazine* (2001), 27–27.

[11] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating Natural Language Adversarial Examples. *arXiv preprint arXiv:1804.07998* (2018).

[12] Yuki Amemiya, Tomohiro Manabe, Sumio Fujita, and Tetsuya Sakai. 2021. How Do Users Revise Zero-Hit Product Search Queries?. In *Advances in Information Retrieval*. 185–192.

[13] Avishek Anand, Lawrence Cavedon, Hideo Joho, Mark Sanderson, and Benno Stein. 2020. Conversational Search (Dagstuhl Seminar 19461). In *Dagstuhl Reports*.

[14] Theo Araujo. 2018. Living Up to the Chatbot Hype: The Influence of Anthropomorphic Design Cues and Communicative Agency Framing on Conversational Agent and Company Perceptions. *Computers in Human Behavior* (2018), 183–189.

[15] Nima Asadi and Jimmy Lin. 2013. Effectiveness/Efficiency Tradeoffs for Candidate Generation in Multi-Stage Retrieval Architectures. In *SIGIR*. 997–1000.

[16] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A Corpus for Adding Memory to Goal-oriented Dialogue Systems. In *SIGDIAL*. 207–219.

[17] Sandeep Avula, Bogeum Choi, and Jaime Arguello. 2022. The Effects of System Initiative during Conversational Collaborative Search. *HCI* (2022), 1–30.

[18] Leif Azzopardi, Mateusz Dubiel, Martin Halvey, and Jeffery Dalton. 2018. Conceptualizing Agent-human Interactions During the Conversational Search Process. In *CAIR*.

[19] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern Information Retrieval*.

[20] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2015. User Variability and IR System Evaluation. In *SIGIR*. 625–634.

[21] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *SIGIR*. 725–728.

[22] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2017. Retrieval Consistency in the Presence of Query Variations. In *SIGIR*. 395–404.

[23] Krisztian Balog. 2021. Conversational AI From an Information Retrieval Perspective: Remaining Challenges and a Case for User Simulation. In *DESIRES*.

[24] Krisztian Balog, Lucie Flekova, Matthias Hagen, Rosie Jones, Martin Potthast, Filip Radlinski, Mark Sanderson, Svitlana Vakulenko, and Hamed Zamani. 2020. Common Conversational Community Prototype: Scholarly Conversational Assistant. *arXiv preprint arXiv:2001.06910* (2020).

[25] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A Multitask, Multilingual, Multimodal Evaluation of Chatgpt on Reasoning, Hallucination, and Interactivity. *arXiv preprint arXiv:2302.04023* (2023).

[26] Nicholas J Belkin, Colleen Cool, W Bruce Croft, and James P Callan. 1993. The Effect Multiple Query Representations on Information Retrieval System Performance. In *SIGIR*. 339–346.

[27] Nicholas J. Belkin, Paul Kantor, Edward A. Fox, and Joseph A Shaw. 1995. Combining the Evidence of Multiple Query Representations for Information Retrieval. *Information Processing & Management* (1995), 431–448.

[28] Alejandro Bellogin, Pablo Castells, and Ivan Cantador. 2011. Precision-Oriented Evaluation of Recommender Systems: An Algorithmic Comparison. In *RecSys*. 333–336.

[29] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The Long-document Transformer. *arXiv preprint arXiv:2004.05150* (2020).

[30] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *FAT*. 610–623.

[31] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *ICML*. 41–48.

[32] Rodger Benham, J Shane Culpepper, Luke Gallagher, Xiaolu Lu, and Joel Mackenzie. 2018. Towards Efficient and Effective Query Variant Generation.. In *DESIRES*. 62–67.

[33] Rodger Benham, Joel Mackenzie, Alistair Moffat, and J Shane Culpepper. 2019. Boosting Search Performance Using Query Variations. *TOIS* (2019), 1–25.

[34] Kathrin Blagec, Jakob Kraiger, Wolfgang Frühwirt, and Matthias Samwald. 2022. Benchmark Datasets Driving Artificial Intelligence Development Fail to Capture the Needs of Medical Professionals. *Journal of Biomedical Informatics* (2022).

[35] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in Neural Networks. In *ICML*. 1613–1622.

[36] Daniel G Bobrow, Ronald M Kaplan, Martin Kay, Donald A Norman, Henry Thompson, and Terry Winograd. 1977. GUS, a Frame-Driven Dialog System. *Artificial Intelligence* (1977), 155–173.

[37] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational advertising. *JMLR* (2013), 3207–3260.

[38] Samuel R Bowman and George E Dahl. 2021. What Will it Take to Fix Benchmarking in Natural Language Understanding? *arXiv preprint arXiv:2104.02145* (2021).

[39] Pavel Braslavski, Denis Savenkov, Eugene Agichtein, and Alina Dubatovka. 2017. What Do You Mean Exactly? Analyzing Clarification Questions in CQA. In *CHIIR*. 345–348.

[40] Leo Breiman. 1996. Bagging Predictors. *Machine learning* (1996), 123–140.

[41] Leo Breiman. 1996. Stacked Regressions. *Machine learning* (1996), 49–64.

[42] Leo Breiman. 1998. Arcing Classifier. *The annals of statistics* (1998), 801–849.

[43] Stephanie Willen Brown. 2008. The Reference Interview: Theories and Practice. (2008).

[44] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-shot Learners. *NeurIPS* (2020), 1877–1901.

[45] Chris Buckley and Janet Walz. 1999. The TREC-8 Query Track. In *TREC*.

[46] Paweł et al. Budzianowski. 2018. MultiWOZ-A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *EMNLP*. 5016–5026.

[47] Christopher JC Burges. 2010. From Ranknet to Lambdarank to Lambdamart: An Overview. *Learning* 23-581 (2010).

[48] Katriina Byström and Preben Hansen. 2005. Conceptual framework for tasks in information studies. *Journal of the American Society for Information science and Technology* (2005), 1050–1061.

[49] Arthur Câmara and Claudia Hauff. 2020. Diagnosing BERT with Retrieval Heuristics. In *ECIR*. Springer, 605–618.

[50] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-aware Query Suggestion by Mining Click-through and Session Data. In *SIGKDD*. ACM, 875–883.

[51] Arun Tejasvi Chaganty, Megan Leszczynski, Shu Zhang, Ravi Ganti, Krisztian Balog, and Filip Radlinski. 2023. Beyond Single Items: Exploring User Preferences in Item Sets with the Conversational Playlist Curation Dataset. *arXiv preprint arXiv:2303.06791* (2023).

[52] Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. 2020. Retrievability based Document Selection for Relevance Feedback with Automatically Generated Query Variants. In *CIKM*. 125–134.

[53] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. 2017. Active Bias: Training More Accurate Neural Networks by Emphasizing High Variance Samples. In *NeurIPS*. 1002–1012.

[54] Jonathan P Chang, Caleb Chiam, Liye Fu, Andrew Z Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. 2020. Convokit: A Toolkit for the Analysis of Conversations. *arXiv preprint arXiv:2005.04246* (2020).

[55] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *SIGKDD Explorations Newsletter* (2017), 25–35.

[56] Xinlei Chen and Abhinav Gupta. 2015. Webly Supervised Learning of Convolutional Networks. In *ICCV*. 1431–1439.

[57] Xuanang Chen, Jian Luo, Ben He, Le Sun, and Yingfei Sun. 2022. Towards Robust Dense Retrieval via Local Ranking Alignment. In *IJCAI*. 1980–1986.

[58] Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving Disentangled Text Representation Learning with Information-Theoretic Guidance. *arXiv preprint arXiv:2006.00693* (2020).

[59] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *EMNLP*. 2174–2184.

[60] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311* (2022).

[61] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *ACL Workshop BlackboxNLP*. 276–286.

[62] Cyril W Cleverdon. 1960. The aslib Cranfield Research Project on the Comparative Efficiency of Indexing Systems. In *Aslib Proceedings*.

[63] Daniel Cohen, Bhaskar Mitra, Oleg Lesota, Navid Rekabsaz, and Carsten Eickhoff. 2021. Not all Relevance Scores Are Equal: Efficient Uncertainty and Calibration Modeling for Deep Retrieval Models. In *SIGIR*. 654–664.

[64] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* (1960), 37–46.

[65] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active Learning with Statistical Models. *Journal of artificial intelligence research* (1996), 129–145.

[66] Kenneth Mark Colby, Sylvia Weber, and Franklin Dennis Hilf. 1971. Artificial Paranoia. *Artificial Intelligence* (1971), 1–25.

[67] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *SIGIR*. 758–759.

[68] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 Deep Learning Track. *arXiv preprint arXiv:2003.07820* (2020).

[69] J Shane Culpepper, Charles LA Clarke, and Jimmy Lin. 2016. Dynamic Cutoff Prediction in Multi-stage Retrieval Systems. In *ADCS*. 17–24.

[70] J Shane Culpepper, 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 *SIGIR*. 34–90.

[71] Zhuyun Dai and Jamie Callan. 2019. Context-aware Sentence/Passage Term Importance Estimation for First Stage Retrieval. *arXiv preprint arXiv:1910.10687* (2019).

[72] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *SIGIR*. 985–988.

[73] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching n-grams in Ad-hoc Search. In *WSDM*. 126–134.

[74] Jeffrey Dalton, Sophie Fischer, Paul Owoicho, Filip Radlinski, Federico Rossetto, Johanne R Trippas, and Hamed Zamani. 2022. Conversational Information Seeking: Theory and Application. In *SIGIR*. 3455–3458.

[75] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2019. Cast 2019: The Conversational Assistance Track Overview. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC*. 13–15.

[76] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A Survey of the State of Explainable AI for Natural Language Processing. *arXiv preprint arXiv:2010.00711* (2020).

[77] Morris H DeGroot and Stephen E Fienberg. 1983. The Comparison and Evaluation of Forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)* 1-2 (1983), 12–22.

[78] John S. Denker, Daniel B. Schwartz, Ben S. Wittner, Sara A. Solla, Richard E. Howard, Lawrence D. Jackel, and John J. Hopfield. 1987. Large Automatic Learning, Rule Extraction, and Generalization. *Complex Systems* (1987).

[79] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on Evaluation Methods for Dialogue Systems. *Artificial Intelligence Review* (2021), 755–810.

[80] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.

[81] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. *ICLR* (2019).

[82] Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2022. A Survey of Natural Language Generation. *Comput. Surveys* (2022), 1–38.

[83] Jianxiong Dong and Jim Huang. 2018. Enhance Word Representation for Out-of-Vocabulary on Ubuntu Dialogue Corpus. *arXiv preprint arXiv:1802.02614* (2018).

[84] Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan Awadallah. 2021. What do Compressed Large Language Models Forget? Robustness Challenges in Model Compression. *arXiv preprint arXiv:2110.08419* (2021).

[85] Andrzej Duda and Mark A Sheldon. 1994. Content Routing in a Network of WAIS Servers. In *ICDS*. 124–132.

[86] Joan C Durrance. 1989. Reference Success: Does the 55 Percent Rule Tell the Whole Story? *Library Journal* (1989), 31–36.

[87] Jeffrey L Elman. 1993. Learning and Development in Neural Networks: The Importance of Starting Small. *Cognition* (1993), 71–99.

[88] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-Value Retrieval Networks for Task-Oriented Dialogue. In *SIGDIAL*. 37–49.

[89] Guglielmo Faggioli, Marco Ferrante, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2021. Hierarchical Dependence-Aware Evaluation Measures for Conversational Search. In *SIGIR*. 1935–1939.

[90] Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2020. Beyond English-centric Multilingual Machine Translation. *arXiv preprint arXiv:2010.11125* (2020).

[91] Christian Federmann, Oussama Elachqar, and Chris Quirk. 2019. Multilingual Whispers: Generating Paraphrases with Translation. In *Workshop on Noisy User-generated Text*. 17–26.

[92] Yulan Feng, Shikib Mehri, Maxine Eskenazi, and Tiancheng Zhao. 2020. "None of the Above": Measure Uncertainty in Dialog Response Retrieval. In *ACL*.

[93] Nicola Ferro, Claudio Lucchese, Maria Maistro, and Raffaele Perego. 2018. Continuation Methods and Curriculum Learning for Learning to Rank. In *CIKM*. 1523–1526.

[94] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. *arXiv preprint arXiv:2109.10086* (2021).

[95] Jonathan Frankle and Michael Carbin. 2018. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *arXiv preprint arXiv:1803.03635* (2018).

[96] Xiao Fu, Emine Yilmaz, and Aldo Lipani. 2022. Evaluating the Cranfield Paradigm for Conversational Search Systems. In *ICTIR*. 275–280.

[97] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-Again Neural Networks. In *ICML*. 1602–1611.

[98] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. 1987. The Vocabulary Problem in Human-System Communication. *Commun. ACM* (1987), 964–971.

[99] Yarin Gal. 2016. Uncertainty in Deep Learning. *University of Cambridge* (2016).

[100] Y Gal and Z Ghahramani. 2016. Dropout as a Bayesian Approximation. In *ICML*. 1661–1680.

[101] Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural Approaches to Conversational AI. In *SIGIR*. 1371–1374.

[102] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. In *SPW*. 50–56.

[103] Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2023. *Neural Approaches to Conversational Information Retrieval*. Springer Nature.

[104] Luyu Gao and Jamie Callan. 2021. Unsupervised Corpus Aware Language Model Pre-Training for Dense Passage Retrieval. *arXiv preprint arXiv:2108.05540* (2021).

[105] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. *arXiv preprint arXiv:2104.07186* (2021).

[106] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink Training of BERT Rerankers in Multi-stage Retrieval Pipeline. In *ECIR*. Springer, 280–286.

[107] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complement Lexical Retrieval Model with Semantic Residual Embeddings. In *ECIR*. Springer, 146–160.

[108] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *EMNLP*. 1307–1323.

[109] Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based Adversarial Examples for Text Classification. *arXiv preprint arXiv:2004.01970* (2020).

[110] Chen Gong, Dacheng Tao, Stephen J Maybank, Wei Liu, Guoliang Kang, and Jie Yang. 2016. Multi-Modal Curriculum Learning for Semi-Supervised Image Classification. *IEEE Transactions on Image Processing* (2016), 3249–3260.

[111] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572* (2014).

[112] Michael D Gordon and Peter Lenk. 1991. A Utility Theoretic Examination of the Probability Ranking Principle in Information Retrieval. *Journal of the American Society for Information Science* (1991), 703–714.

[113] Jia-Chen Gu, Tianda Li, Quan Liu, Xiaodan Zhu, Zhen-Hua Ling, Zhiming Su, and Si Wei. 2020. Speaker-Aware BERT for Multi-Turn Response Selection in Retrieval-Based Chatbots. *arXiv preprint arXiv:2004.03588* (2020).

[114] Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2019. Interactive Matching Network for Multi-turn Response Selection in Retrieval-Based Chatbots. In *CIKM*. 2321–2324.

[115] Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2019. Utterance-to-Utterance Interactive Matching Network for Multi-Turn Response Selection in Retrieval-Based Chatbots. *TASLP* (2019), 369–379.

[116] Liangke Gui, Tadas Baltrušaitis, and Louis-Philippe Morency. 2017. Curriculum Learning for Facial Expression Recognition. In *FG*. 505–511.

[117] Ishaan Gulrajani and David Lopez-Paz. 2020. In Search of Lost Domain Generalization. *arXiv preprint arXiv:2007.01434* (2020).

[118] Chulaka Gunasekara, Jonathan K Kummerfeld, Lazaros Polymenakos, and Walter Lasecki. 2019. Dstc7 Task 1: Noetic end-to-end Response Selection. In *Workshop on NLP for Conversational AI*. 60–67.

[119] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On Calibration of Modern Neural Networks. *arXiv preprint arXiv:1706.04599* (2017).

[120] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM*. 55–64.

[121] Qi Guo and Eugene Agichtein. 2012. Beyond Dwell Time: Estimating Document Relevance from Cursor Movements and Other Post-C lick Searcher Behavior. In *WWW*. ACM, 569–578.

[122] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2019. Attentive Long Short-term Preference Modeling for Personalized Product Search. *TOIS* (2019), 1–27.

[123] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*. 8342–8360.

[124] Guy Hacohen and Daphna Weinshall. 2019. On the Power of Curriculum Learning in Training Deep Networks. *arXiv preprint arXiv:1904.03626* (2019).

[125] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*. 1735–1742.

[126] Janghoon Han, Taesuk Hong, Byoungjae Kim, Youngjoong Ko, and Jungyun Seo. 2021. Fine-grained post-training for improving retrieval-based dialogue systems. In *NAACL*. 1549–1558.

[127] F Maxwell Harper and Joseph A Konstan. 2015. The Movielens Datasets: History and Context. *TiiS* (2015), 1–19.

[128] Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft. 2020. ANTIQUE: A Non-factoid Question Answering Benchmark. In *ECIR*. Springer, 166–173.

[129] Matthew Henderson, Paweł Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, et al. 2019. A Repository of Conversational Datasets. *arXiv preprint arXiv:1904.06472* (2019).

[130] Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Ivan Vulić, et al. 2019. ConveRT: Efficient and Accurate Conversational Representations from Transformers. *arXiv preprint arXiv:1911.03688* (2019).

[131] Peter Hernon and Charles R McClure. 1986. Unobtrusive Reference Testing: The 55 Percent Rule. *Library Journal* 111 (1986), 37–41.

[132] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. 2018. Towards a Definition of Disentangled Representations. *arXiv preprint arXiv:1812.02230* (2018).

[133] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* (2015).

[134] Sharon Hirsch, Ido Guy, Alexander Nus, Arnon Dagan, and Oren Kurland. 2020. Query Reformulation in E-commerce Search. In *SIGIR*. 1319–1328.

[135] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. *arXiv preprint arXiv:2010.02666* (2020).

[136] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retrierver with Balanced Topic Aware Sampling. In *SIGIR*. 113–122.

[137] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *NeurIPS*. 2042–2050.

[138] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *CIKM*. 2333–2338.

[139] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Polyencoders: Transformer Architectures and Pre-Training Strategies for Fast and Accurate Multi-sentence Scoring. *arXiv preprint arXiv:1905.01969* (2019).

[140] Clayton J Hutto and Eric Gilbert. 2014. Vader: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In *ICWSM*. 216–225.

[141] Andrea Iovine, Pasquale Lops, Fedelucio Narducci, Marco de Gemmis, and Giovanni Semeraro. 2022. An Empirical Evaluation of Active Learning Strategies for Profile Elicitation in a Conversational Recommender System. *Journal of Intelligent Information Systems* (2022), 337–362.

[142] Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain J Marshall, and Byron C Wallace. 2018. Learning Disentangled Representations of Texts with Application to Biomedical Abstracts. In *EMNLP*. 4683.

[143] Dietmar Jannach and Li Chen. 2022. Conversational Recommendation: A Grand AI Challenge. *arXiv preprint arXiv:2203.09126* (2022).

[144] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A Survey on Conversational Recommender Systems. *ACM Computing Surveys (CSUR)* (2021), 1–36.

[145] Bernard J Jansen, Danielle L Booth, and Amanda Spink. 2009. Patterns of Query Reformulation During Web Searching. *Journal of the American Society for Information Science and Technology* (2009), 1358–1371.

[146] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn About the Structure of Language?. In *ACL*.

[147] Ziwei Ji, 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. *Comput. Surveys* (2023), 1–38.

[148] Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *EMNLP*. 2021–2031.

[149] Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer With Dynamic Negative Sampling for High-Performance Extreme Multi-Label Text Classification. *arXiv preprint arXiv:2101.03305* (2021).

[150] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. 2012. Calibrating Predictive Model Estimates to Support Personalized Medicine. *Journal of the American Medical Informatics Association* (2012), 263–274.

[151] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *TACL* (2020), 423–438.

[152] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In *AAAI*. 8018–8025.

[153] Thorsten Joachims, Laura A Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately Interpreting Click-Through Data as Implicit Feedback. In *SIGIR*. 154–161.

[154] Thorsten Joachims and Adith Swaminathan. 2016. Counterfactual Evaluation and Learning for Search, Recommendation and Ad Placement. In *SIGIR*. 1199–1201.

[155] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data* (2019), 535–547.

[156] Karen Sparck Jones. 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of documentation* (1972).

[157] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating Query Substitutions. In *WWW*. 387–396.

[158] Dan Jurafsky and James H Martin. 2019. Speech and Language Processing.

[159] Rudolf Kadlec, Martin Schmid, and Jan Kleindienst. 2015. Improved Deep Learning Baselines for Ubuntu Corpus Dialogs. *arXiv preprint arXiv:1510.03753* (2015).

[160] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint arXiv:2004.04906* (2020).

[161] Abhishek Kaushik and Gareth JF Jones. 2021. A Conceptual Framework for Implicit Evaluation of Conversational Search Interfaces. *arXiv preprint arXiv:2104.03940* (2021).

[162] Diane Kelly et al. 2009. Methods for Evaluating Interactive Information Retrieval Systems With Users. *Foundations and Trends® in Information Retrieval* (2009), 1–224.

[163] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *CVPR*. 7482–7491.

[164] Tom Kenter and Maarten de Rijke. 2017. Attentive Memory Networks: Efficient Machine Reading for Conversational Search. *arXiv preprint arXiv:1712.07229* (2017).

[165] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and Effective Passage Search via Contextualized Late Interaction Over BERT. In *SIGIR*. 39–48.

[166] Jack Kiefer and Jacob Wolfowitz. 1952. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics* (1952), 462–466.

[167] Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. 2019. The Eighth Dialog System Technology Challenge. *arXiv preprint arXiv:1911.06394* (2019).

[168] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[169] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the national academy of sciences* (2017), 3521–3526.

[170] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. *arXiv preprint arXiv:2001.04451* (2020).

[171] Tom Kocmi and Ondrej Bojar. 2017. Curriculum Learning and Minibatch Bucketing in Neural Machine Translation. *RANLP* (2017), 379–386.

[172] Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. ChatGPT: Jack of All Trades, Master of None. *arXiv preprint arXiv:2302.10724* (2023).

[173] Carol C Kuhlthau. 1991. Inside the Search Process: Information Seeking from the User's Perspective. *Journal of the American Society for Information Science* (1991), 361–371.

[174] M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced Learning for Latent Variable Models. In *NeurIPS*. 1189–1197.

[175] Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A Large-Scale Corpus for Conversation Disentanglement. *ACL* (2019). https://doi.org/10.18653/v1/p19-1374

[176] Gautam Kunapuli. 2023. *Ensemble Methods for Machine Learning.*

[177] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *NeurIPS*. 6402–6413.

[178] Tian Lan, Deng Cai, Yan Wang, Yixuan Su, Xian-Ling Mao, and Heyan Huang. 2021. Exploring Dense Retrieval for Dialogue Response Selection. *arXiv preprint arXiv:2110.06612* (2021).

[179] Stefan Larson and Kevin Leach. 2022. A Survey of Intent Classification and Slot-Filling Datasets for Task-Oriented Dialog. *arXiv preprint arXiv:2207.13211* (2022).

[180] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *SIGIR*. 120–127.

[181] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *NeurIPS*. 7167–7177.

[182] Piyawat Lertvittayakumjorn and Francesca Toni. 2021. Explanation-Based Human Debugging of NLP Models: A Survey. *Transactions of the Association for Computational Linguistics* (2021), 1508–1528.

[183] Megan Leszczynski, Ravi Ganti, Shu Zhang, Krisztian Balog, Filip Radlinski, Fernando Pereira, and Arun Tejasvi Chaganty. 2023. Generating Synthetic Data for Conversational Music Recommendation Using Random Walks and Language Models. *arXiv preprint arXiv:2301.11489* (2023).

[184] Jia Li, Chongyang Tao, Wei Wu, Yansong Feng, Dongyan Zhao, and Rui Yan. 2019. Sampling Matters! An Empirical Study of Negative Sampling Strategies for Learning of Matching Models in Retrieval-based Dialogue Systems. In *EMNLP-IJCNLP*. 1291–1296.

[185] Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2022. CITADEL: Conditional Token Interaction via Dynamic Lexical Routing for Efficient and Effective Multi-Vector Retrieval. *arXiv preprint arXiv:2211.10411* (2022).

[186] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *NeurIPS*. 9725–9735.

[187] Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2020. MixKD: Towards Efficient Distillation of Large-Scale Language Models. *arXiv preprint arXiv:2011.00593* (2020).

[188] Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards Understanding and Mitigating Social Biases in Language Models. In *ICML*. PMLR, 6565–6576.

[189] Thomas Liao, Rohan Taori, Inioluwa Deborah Raji, and Ludwig Schmidt. 2021. Are We Learning Yet? A Meta Review of Evaluation Failures Across Machine Learning. In *NeurIPS*.

[190] Jimmy Lin. 2022. A Proposed Conceptual Framework for a Representational Approach to Information Retrieval. In *ACM SIGIR Forum*. 1–29.

[191] Jimmy Lin and Xueguang Ma. 2021. A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. *arXiv preprint arXiv:2106.14807* (2021).

[192] Jimmy Lin, 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 *SIGIR*. 2356–2362.

[193] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. *Synthesis Lectures on Human Language Technologies* (2021), 1–325.

[194] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Query reformulation using query history for passage retrieval in conversational search. *arXiv preprint arXiv:2005.02230* (2020).

[195] Zibo Lin, Deng Cai, Yan Wang, Xiaojiang Liu, Hai-Tao Zheng, and Shuming Shi. 2020. The World is not Binary: Learning to Rank with Grayscale Data for Dialogue Response Selection. *arXiv preprint arXiv:2004.02421* (2020).

[196] Zihan Lin, Hui Wang, Jingshu Mao, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and Ji-Rong Wen. 2022. Feature-aware Diversified Re-ranking with Disentangled Representations for Relevant Recommendation. In *SIGKDD*. 3327–3335.

[197] Aldo Lipani, Ben Carterette, and Emine Yilmaz. 2021. How am I doing?: Evaluating Conversational Search Systems offline. *TOIS* (2021), 1–22.

[198] Bulou Liu, Yueyue Wu, Yiqun Liu, Fan Zhang, Yunqiu Shao, Chenliang Li, Min Zhang, and Shaoping Ma. 2021. Conversational vs Traditional: Comparing Search Behavior and Outcome in Legal Case Retrieval. In *SIGIR*. 1622–1626.

[199] Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum Learning for Natural Answer Generation. In *IJCAI*. 4223–4229.

[200] Tie-Yan Liu et al. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval* (2009), 225–331.

[201] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[202] Zeyang Liu, Ke Zhou, and Max L Wilson. 2021. Meta-Evaluation of Conversational Search Evaluation Metrics. *TOIS* (2021), 1–42.

[203] Jesús Lovón-Melgarejo, Laure Soulier, Karen Pinel-Sauvagnat, and Lynda Tamine. 2021. Studying Catastrophic Forgetting in Neural Ranking Models. In *ECIR*. Springer, 375–390.

[204] Ryan Lowe, Nissan Pow, Iulian V Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *SIGDIAL*. 285–294.

[205] Jing Lu, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2020. Neural Passage Retrieval with Improved Negative Contrast. *arXiv preprint arXiv:2010.12523* (2020).

[206] Xiaolu Lu, Oren Kurland, J Shane Culpepper, Nick Craswell, and Ofri Rom. 2019. Relevance Modeling with Multiple Query Variations. In *SIGIR*. 27–34.

[207] Claudio Lucchese, Franco Maria Nardini, Raffaele Perego, and Salvatore Trani. 2017. The Impact of Negative Samples on Learning to Rank.. In *LEARNER@ ICTIR*.

[208] Sean MacAvaney. 2020. OpenNIR: A Complete Neural Ad-Hoc Ranking Pipeline. In *WSDM*. 353–360.

[209] Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2020. ABNIRML: Analyzing the Behavior of Neural IR Models. *arXiv preprint arXiv:2011.00696* (2020).

[210] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via Prediction of Importance with Contextualization. In *SIGIR*. 1573–1576.

[211] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Training Curricula for Open Domain Answer Re-ranking. In *SIGIR*. 529–538.

[212] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified Data Wrangling with ir_datasets. In *SIGIR*. 2429–2436.

[213] Craig Macdonald and Nicola Tonellotto. 2020. Declarative Experimentation in Information Retrieval using PyTerrier. In *ICTIR*. 161–168.

[214] David JC MacKay. 1992. A Practical Bayesian Framework for Backpropagation Networks. *Neural computation* (1992), 448–472.

[215] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A Simple Baseline for Bayesian Uncertainty in Deep Learning. In *NeurIPS*. 13153–13164.

[216] Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning Passage Impacts for Inverted Indexes. In *SIGIR*. 1723–1727.

[217] Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing Revisited with Neural Machine Translation. In *EACL*. 881–893.

[218] Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022. Curriculum Contrastive Context Denoising for Few-shot Conversational Dense Retrieval. In *SIGIR*. 176–186.

[219] Gary Marchionini. 2006. Exploratory Search: From Finding to Understanding. *Commun. ACM* (2006), 41–46.

[220] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training Millions of Personalized Dialogue Agents. *arXiv preprint arXiv:1809.01984* (2018).

[221] Ida Mele, Cristina Ioana Muntean, Mohammad Aliannejadi, and Nikos Voskarides. 2021. MICROS: Mixed-Initiative ConveRsatiOnal Systems Workshop. In *ECIR*. Springer, 710–713.

[222] Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One?. In *NeurIPS*. 14014–14024.

[223] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781* (2013).

[224] Alistair Moffat, Falk Scholer, Paul Thomas, and Peter Bailey. 2015. Pooled Evaluation Over Query Variations: Users are as Diverse as Systems. In *CIKM*. 1759–1762.

[225] Felipe Moraes, Kilian Grashoff, and Claudia Hauff. 2019. On the Impact of Group Size on Collaborative Search Effectiveness. *Information Retrieval Journal* (2019), 476–498.

[226] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 119–126.

[227] Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-Fitting Word Vectors to Linguistic Constraints. *arXiv preprint arXiv:1603.00892* (2016).

[228] Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. 2015. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *AAAI*. 2901.

[229] Devesh Narayanan and David De Cremer. 2022. "Google Told Me So!" On the Bent Testimony of Search Engine Algorithms. *Philosophy & Technology* (2022), 1–19.

[230] Radford M Neal. 1995. *Bayesian Learning for Neural Networks*. Ph.D. Dissertation. University of Toronto.

[231] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. In *CoCo@ NIPS*.

[232] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP-IJCNLP*. 188–197.

[233] Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2022. Recent Advances in Deep Learning Based Dialogue Systems: A Systematic Survey. *Artificial Intelligence Review* (2022), 1–101.

[234] Raymond S Nickerson. 1976. On Conversational Interaction With Computers. In *ACM/SIGGRAPH Workshop on User-Oriented Design of Interactive Graphics Systems*. 101–113.

[235] Yifan Nie, Yanling Li, and Jian-Yun Nie. 2018. Empirical Study of Multi-level Convolution Models for IR Based on Representations and Interactions. In *SIGIR*. 59–66.

[236] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[237] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. *arXiv preprint arXiv:2003.06713* (2020).

[238] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint* 6 (2019).

[239] Robert N Oddy. 1977. Information Retrieval Through Man-Machine Dialogue. *Journal of documentation* (1977), 1–14.

[240] Barlas Oğuz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, et al. 2021. Domain-Matched Pre-Training Tasks for Dense Retrieval. *arXiv preprint arXiv:2107.13602* (2021).

[241] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instructions with Human Feedback. *arXiv preprint arXiv:2203.02155* (2022).

[242] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you Trust your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. In *NeurIPS*. 13991–14002.

[243] Harshith Padigela, Hamed Zamani, and W Bruce Croft. 2019. Investigating the Successes and Failures of BERT for Passage Re-Ranking. *arXiv preprint arXiv:1905.01758* (2019).

[244] Emilie Palagi, Fabien Gandon, Alain Giboin, and Raphaël Troncy. 2017. A Survey of Definitions and Models of Exploratory Search. In *Proceedings of the 2017 ACM workshop on Exploratory Search and Interactive Data Analytics*. 3–8.

[245] Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Kam-Fai Wong, and Shang-Yu Su. 2018. Deep dyna-q: Integrating Planning for Task-completion Dialogue Policy Learning. *arXiv preprint arXiv:1801.06176* (2018).

[246] Gustavo Penha, Alexandru Balan, and Claudia Hauff. 2019. Introducing MANtIS: A Novel Multi-Domain Information Seeking Dialogues Dataset. *arXiv preprint arXiv:1912.04639* (2019).

[247] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In *ECIR*. Springer, 397–412.

[248] Gustavo Penha and Claudia Hauff. 2020. Challenges in the Evaluation of Conversational Search Systems.. In *Converse@KDD*.

[249] Gustavo Penha and Claudia Hauff. 2020. Curriculum Learning Strategies for IR. In *ECIR*. Springer, 699–713.

[250] Gustavo Penha and Claudia Hauff. 2020. What Does BERT Know About Books, Movies and Music? Probing BERT for Conversational Recommendation. In *RecSys*. 388–397.

[251] Gustavo Penha and Claudia Hauff. 2021. On the Calibration and Uncertainty of Neural Learning to Rank Models for Conversational Search. In *EACL*. 160–170.

[252] Gustavo Penha and Claudia Hauff. 2021. Weakly Supervised Label Smoothing. In *ECIR*. Springer, 334–341.

[253] Gustavo Penha and Claudia Hauff. 2023. Do the Findings of Document and Passage Retrieval Generalize to the Retrieval of Responses for Dialogues?. In *ECIR*. Springer, 132–147.

[254] Gustavo Penha, Svitlana Vakulenko, Ondrej Dusek, Leigh Clark, Vaishali Pal, and Vaibhav Adlakha. 2022. The Seventh Workshop on Search-Oriented Conversational Artificial Intelligence (SCAI'22). In *SIGIR*. 3466–3469.

[255] Anastasia Pentina, Viktoriia Sharmanska, and Christoph H Lampert. 2015. Curriculum Learning of Multiple Tasks. In *CVPR*. 5492–5500.

[256] Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. *arXiv preprint arXiv:2005.04611* (2020).

[257] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *EMNLP-IJCNLP*. 2463–2473.

[258] Thang M Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of Order: How Important is the Sequential Order of Words in a Sentence in Natural Language Understanding Tasks? *arXiv preprint arXiv:2012.15180* (2020).

[259] Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence Encoders on Stilts: Supplementary Training on Intermediate Labeled-data Tasks. *arXiv:1811.01088* (2018).

[260] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based Curriculum Learning for Neural Machine Translation. In *NAACL*. 1162–1172.

[261] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *SIGIR (SIGIR '98)*. 275–281.

[262] Martin Potthast, Matthias Hagen, and Benno Stein. 2021. The Dilemma of the Direct Answer. In *ACM SIGIR Forum*. 1–12.

[263] Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-Task Transfer Learning with Pretrained Language Models: When and Why Does It Work?. In *ACL*. 5231–5247.

[264] Yujie Qian, Jinhyuk Lee, Sai Meher Karthik Duddu, Zhuyun Dai, Siddhartha Brahma, Iftekhar Naim, Tao Lei, and Vincent Y Zhao. 2022. Multi-Vector Retrieval as Sparse Alignment. *arXiv preprint arXiv:2211.01267* (2022).

[265] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. *arXiv preprint arXiv:1904.07531* (2019).

[266] Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is Chatgpt a General-Purpose Natural Language Processing Task Solver? *arXiv preprint arXiv:2302.06476* (2023).

[267] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained Models for Natural Language Processing: A Survey. *arXiv:2003.08271* (2020).

[268] Chen Qu, Liu Yang, W Bruce Croft, Johanne R Trippas, Yongfeng Zhang, and Minghui Qiu. 2018. Analyzing and characterizing user intent in information-seeking conversations. In *SIGIR*. 989–992.

[269] Chen Qu, Liu Yang, W Bruce Croft, Yongfeng Zhang, Johanne R Trippas, and Minghui Qiu. 2019. User intent prediction in information-seeking conversations. In *CHIIR*. 25–33.

[270] Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with history answer embedding for conversational question answering. In *SIGIR*. 1133–1136.

[271] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-domain Question Answering. *arXiv preprint arXiv:2010.08191* (2020).

[272] Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached Conversational Preference Elicitation: A Case Study in Understanding Movie Preferences. In *SIGDIAL*. 353–360.

[273] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *CHIIR*. 117–126.

[274] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv preprint arXiv:1910.10683* (2019).

[275] Shivesh Ranjan, John HL Hansen, Shivesh Ranjan, and John HL Hansen. 2018. Curriculum Learning Based Approaches for Noise Robust Speaker Recognition. *TASLP* (2018), 197–210.

[276] Jinfeng Rao, Linqing Liu, Yi Tay, Wei Yang, Peng Shi, and Jimmy Lin. 2019. Bridging the Gap Between Relevance Matching and Semantic Matching for Short Text Similarity Modeling. In *EMNLP-IJCNLP*. 5370–5381.

[277] Sudha Rao. 2017. Are You Asking the Right Questions? Teaching Machines to Ask Clarification Questions. In *ACL Student Research Workshop*. 30–35.

[278] Sudha Rao and Hal Daumé III. 2018. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information. *ACL* (2018), 2737–2746.

[279] Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* (2019), 249–266.

[280] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. In *EMNLP-IJCNLP*. 3973–3983.

[281] Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. PAIR: Leveraging Passage-Centric Similarity Relation for Improving Dense Passage Retrieval. *arXiv preprint arXiv:2108.06027* (2021).

[282] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qifei Wu, Yuchen Ding, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2022. A Thorough Examination on Zero-Shot Dense Retrieval. *arXiv preprint arXiv:2204.12755* (2022).

[283] Daniël Rennings, Felipe Moraes, and Claudia Hauff. 2019. An Axiomatic Approach to Diagnosing Neural IR Models. In *ECIR*. Springer, 489–503.

[284] Gustavo Resende, Philipe Melo, Hugo Sousa, Johnnatan Messias, Marisa Vasconcelos, Jussara Almeida, and Fabrício Benevenuto. 2019. (Mis) Information Dissemination in WhatsApp: Gathering, Analyzing and Countermeasures. In *WWW*. 818–828.

[285] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. *arXiv preprint arXiv:2005.04118* (2020).

[286] Alan Ritter, Colin Cherry, and Bill Dolan. 2011. Data-Driven Response Generation in Social Media. In *EMNLP*.

[287] Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? *arXiv:2002.08910* (2020).

[288] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* (2009), 333–389.

[289] Stephen E Robertson. 1977. The Probability Ranking Principle in IR. *Journal of documentation* (1977).

[290] Stephen E Robertson and Steve Walker. 1994. Some Simple Effective Approximations to the 2-poisson Model for Probabilistic weighted retrieval. In *SIGIR*. 232–241.

[291] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive Learning with Hard Negative Samples. *arXiv preprint arXiv:2010.04592* (2020).

[292] Joseph John Rocchio. 1971. Relevance Feedback in Information Retrieval. *The SMART retrieval system: experiments in automatic document processing* (1971), 313–323.

[293] Douglas LT Rohde and David C Plaut. 1999. Language Acquisition in the Absence of Explicit Negative Evidence: How Important is Starting Small? *Cognition* (1999), 67–109.

[294] Catherine Sheldrick Ross, Kirsti Nilsen, and Marie L Radford. 2019. *Conducting the Reference Interview*. American Library Association.

[295] Mrinmaya Sachan and Eric Xing. 2016. Easy Questions First? A Case Study on Curriculum Learning for Question Answering. In *ACL*. 453–463.

[296] Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of Natural Language Rules in Conversational Machine Reading. In *EMNLP*. 2087–2097.

[297] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. FAQ Retrieval using Query-Question Similarity and BERT-Based Query-Answer Relevance. *arXiv preprint arXiv:1905.02851* (2019).

[298] Ivan Sekulić, Mohammad Aliannejadi, and Fabio Crestani. 2022. Evaluating mixed-initiative Conversational Search Systems via user simulation. In *WSDM*. 888–896.

[299] Chirag Shah and Emily M Bender. 2022. Situating Search. In *SIGIR*. 221–232.

[300] Chirag Shah and Roberto González-Ibáñez. 2010. Exploring Information Seeking Processes in Collaborative Search Tasks. *Proceedings of the American Society for Information Science and Technology* (2010), 1–7.

[301] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *CIKM*. 101–110.

[302] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. 2016. Training Region-based Object Detectors with Online Hard Example Mining. In *CVPR*. 761–769.

[303] Anna Shtok, Oren Kurland, and David Carmel. 2009. Predicting Query Performance by Query-Drift Estimation. In *ICTIR*. 305–312.

[304] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. *arXiv preprint arXiv:2104.07567* (2021).

[305] Damien Sileo, Wout Vossen, and Robbe Raymaekers. 2022. Zero-Shot Recommendation as Language Modeling. In *ECIR*. Springer, 223–230.

[306] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little. *arXiv preprint arXiv:2104.06644* (2021).

[307] Donggil Song, Eun Young Oh, and Marilyn Rice. 2017. Interacting With a Conversational Agent System for Educational Purposes in Online Courses. In *HSI*. 78–82.

[308] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and Permuted Pre-training for Language Understanding. *NeurIPS* (2020), 16857–16867.

[309] Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan. 2018. An Ensemble of Retrieval-Based and Generation-Based Human-Computer Conversation Systems. In *IJCAI*. 4382–4388.

[310] Petru Soviany, Claudiu Ardei, Radu Tudor Ionescu, and Marius Leordeanu. 2019. Image Difficulty Curriculum for Generative Adversarial Networks (CuGAN). *arXiv preprint arXiv:1910.08967* (2019).

[311] Karen Spark-Jones. 1975. Report on the Need for and Provision of an 'Ideal' Information Retrieval Test Collection. *Computer Laboratory* (1975).

[312] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a Simple Way to Prevent Neural Networks from Overfitting. *The journal of machine learning research* (2014), 1929–1958.

[313] Sandeep Subramanian, Sai Rajeswar, Francis Dutil, Chris Pal, and Aaron Courville. 2017. Adversarial Generation of Natural Language. In *Rep4NLP*. 241–251.

[314] Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-BERT: BERT is not Robust on Misspellings! Generating Nature Adversarial Samples on BERT. arXiv:2003.04985

[315] Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. oLMpics–On what Language Model Pre-training Captures. *arXiv:1912.13283* (2019).

[316] Da Tang, Xiujun Li, Jianfeng Gao, Chong Wang, Lihong Li, and Tony Jebara. 2018. Subgoal Discovery for Hierarchical Dialogue Policy Learning. *arXiv preprint arXiv:1804.07855* (2018).

[317] Chongyang Tao, Jiazhan Feng, Chang Liu, Juntao Li, Xiubo Geng, and Daxin Jiang. 2021. Building an Efficient and Effective Retrieval-based Dialogue System via Mutual Learning. *arXiv preprint arXiv:2110.00159* (2021).

[318] Chongyang Tao, Jiazhan Feng, Rui Yan, Wei Wu, and Daxin Jiang. 2021. A Survey on Response Selection for Retrieval-based Dialogues. In *IJCAI*. 4619–4626.

[319] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-Representation Fusion Network for Multi-Turn Response Selection in Retrieval-Based Chatbots. In *WSDM*. 267–275.

[320] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. One Time of Interaction May Not Be Enough: Go Deep with an Interaction-over-Interaction Network for Response Selection in Dialogues. In *ACL*. 1–11.

[321] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long Range Arena: A Benchmark for Efficient Transformers. *arXiv preprint arXiv:2011.04006* (2020).

[322] Robert R Taylor. 1962. Question-Negotiation and Information Seeking in Libraries. *American Documentation* 391 (1962).

[323] Robert S Taylor. 1962. The Process of Asking Questions. *American documentation* (1962), 391–396.

[324] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *ACL*. 4593–4601.

[325] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data Augmentation Method for Improving Bi-encoders for Pairwise Sentence Scoring Tasks. *arXiv preprint arXiv:2010.08240* (2020).

[326] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-Shot Evaluation of Information Retrieval Models. *arXiv preprint arXiv:2104.08663* (2021).

[327] Jayaraman J Thiagarajan, Prasanna Sattigeri, Deepta Rajan, and Bindya Venkatesh. 2020. Calibrating Healthcare AI: Towards Reliable and Interpretable Deep Predictive Models. *arXiv preprint arXiv:2004.14480* (2020).

[328] Paul Thomas, Daniel McDuff, Mary Czerwinski, and Nick Craswell. 2017. MISC: A Data Set of Information-Seeking Conversations. In *CAIR Workshop*.

[329] Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming Catastrophic Forgetting During Domain Adaptation of Neural Machine Translation. In *NAACL*. 2062–2068.

[330] Nava Tintarev and Judith Masthoff. 2015. Explaining Recommendations: Design and Evaluation. In *Recommender Systems handbook*. 353–382.

[331] Simon Tong and Daphne Koller. 2001. Support Vector Machine Active Learning with Applications to Text Classification. *Journal of machine learning research* (2001), 45–66.

[332] Johanne R Trippas. 2021. Spoken Conversational Search: Audio-Only Interactive Information Retrieval. In *ACM SIGIR Forum*. 106–107.

[333] Johanne R Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the Design of Spoken Conversational Search: Perspective Paper. In *CHIIR*. 32–41.

[334] Johanne R Trippas, Damiano Spina, Lawrence Cavedon, and Mark Sanderson. 2017. How do People Interact in Conversational Speech-only Search Tasks: A Preliminary Analysis. In *CHIIR*. 325–328.

[335] Radu Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P Papadopoulos, and Vittorio Ferrari. 2016. How Hard Can it be? Estimating the Difficulty of Visual Search in an Image. In *CVPR*. 2157–2166.

[336] Andrew Turpin, Yohannes Tsegay, David Hawking, and Hugh E Williams. 2007. Fast Generation of Result Snippets in Web Search. In *SIGIR*. 127–134.

[337] Pertti Vakkari and Nanna Hakala. 2000. Changes in Relevance Criteria and Problem Stages in Task Performance. *Journal of Documentation* (2000).

[338] Svitlana Vakulenko, Evangelos Kanoulas, and Maarten De Rijke. 2021. A Large-Scale Analysis of Mixed Initiative in Information-Seeking Dialogues for Conversational Search. *TOIS* (2021), 1–32.

[339] Svitlana Vakulenko, Kate Revoredo, Claudio Di Ciccio, and Maarten de Rijke. 2019. QRFA: A Data-Driven Model of Information-Seeking Dialogues. In *ECIR*. 541–557.

[340] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using t-SNE. *Journal of machine learning research* (2008).

[341] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning Latent Vector Spaces for Product Search. In *CIKM*. 165–174.

[342] Hal R Varian. 1999. Economics and Search. In *ACM SIGIR Forum*. 1–5.

[343] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NeurIPS*. 5998–6008.

[344] Jesse Vig and Kalai Ramea. 2019. Comparison of Transfer-learning Approaches for Response Selection in Multi-turn Conversations. In *Workshop on DSTC7*.

[345] Ellen M Voorhees. 2019. The Evolution of Cranfield. In *Information Retrieval Evaluation in a changing world*. Springer, 45–69.

[346] Alexandra Vtyurina, Denis Savenkov, Eugene Agichtein, and Charles LA Clarke. 2017. Exploring Conversational Search with Humans, Assistants, and Wizards. In *CHI EA*. 2187–2193.

[347] Somin Wadhwa and Hamed Zamani. 2021. Towards System-Initiative Conversational Information Seeking.. In *DESIRES*. 102–116.

[348] Mengting Wan and Julian McAuley. 2018. Item Recommendation on Monotonic Behavior Chains. In *RecSys*. 86–94.

[349] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. In *AAAI*.

[350] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-SRNN: Modeling the Recursive Matching Structure with Spatial RNN. In *IJCAI*. AAAI Press, 2922–2928.

[351] Chengyu Wang, Minghui Qiu, Jun Huang, and Xiaofeng He. 2020. Meta Fine-Tuning Neural Language Models for Multi-Domain Text Mining. arXiv:2003.13003

[352] Jun Wang. 2009. Mean-Variance Analysis: A New Document Ranking Theory in Information Retrieval. In *ECIR*. 4–16.

[353] Jun Wang and Jianhan Zhu. 2009. Portfolio Theory of Information Retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in Information Retrieval*. 115–122.

[354] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge Into Pre-Trained Models with Adapters. *arXiv:2002.01808* (2020).

[355] Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2019. Structured Pruning of Large Language Models. *arXiv preprint arXiv:1910.04732* (2019).

[356] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent Abilities of Large Language Models. *arXiv preprint arXiv:2206.07682* (2022).

[357] Daphna Weinshall, Gad Cohen, and Dan Amir. 2018. Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks. In *ICML*. 5235–5243.

[358] Joseph Weizenbaum. 1966. ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine. *Commun. ACM* (1966), 36–45.

[359] Johannes Welbl, Amelia Glaese, Jonathan Uesato, Sumanth Dathathri, John Mellor, Lisa Anne Hendricks, Kirsty Anderson, Pushmeet Kohli, Ben Coppin, and Po-Sen Huang. 2021. Challenges in Detoxifying Language Models. *arXiv preprint arXiv:2109.07445* (2021).

[360] Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and HeuiSeok Lim. 2019. Domain Adaptive Training BERT for Response Selection. *arXiv preprint arXiv:1908.04812* (2019).

[361] Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. 2021. Do Response Selection Models Really Know What's Next? Utterance Manipulation Strategies for Multi-Turn Response Selection. In *AAAI*. 14041–14049.

[362] Ryen W White and Resa A Roth. 2009. Exploratory Search: Beyond the Query-Response Paradigm. *Synthesis lectures on information concepts, retrieval, and services* (2009), 1–98.

[363] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771* (2019).

[364] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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. 2020. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP: System Demonstrations*. 38–45.

[365] Jed R Wood and Larry E Wood. 2008. Card Sorting: Current Practices and Beyond. *Journal of Usability Studies* (2008), 1–6.

[366] Chen Wu, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2021. Are Neural Ranking Models Robust? *arXiv preprint arXiv:2108.05018* (2021).

[367] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots. In *ACL*. 496–505.

[368] Yijun Xiao and William Yang Wang. 2021. On Hallucination and Predictive Uncertainty in Conditional Language Generation. *arXiv preprint arXiv:2103.15025* (2021).

[369] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end Neural Ad-hoc Ranking with Kernel Pooling. In *SIGIR*. 55–64.

[370] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[371] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System. In *SIGIR*. 55–64.

[372] Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A Hybrid Retrieval-Generation Neural Conversation Model. In *CIKM*. 1341–1350.

[373] Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, and Haiqing Chen. 2020. IART: Intent-aware Response Ranking with Transformers in Information-seeking Conversation Systems. *arXiv preprint arXiv:2002.00571* (2020).

[374] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-Seeking Conversation Systems. In *SIGIR*. 245–254.

[375] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. 2019. Critically Examining the Neural Hype: Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In *SIGIR*. 1129–1132.

[376] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-End Open-Domain Question Answering with BERTserini. In *NACL*. 72–77.

[377] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. *arXiv preprint arXiv:1903.10972* (2019).

[378] Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *EMNLP*. 3687–3692.

[379] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237* (2019).

[380] Zhen Yang, Ming Ding, Chang Zhou, Hongxia Yang, Jingren Zhou, and Jie Tang. 2020. Understanding Negative Sampling in Graph Representation Learning. In *SIGKDD*. 1666–1676.

[381] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. EMNLP, 3481–3487.

[382] Koichiro Yoshino, Chiori Hori, Julien Perez, Luis Fernando D'Haro, Lazaros Polymenakos, Chulaka Gunasekara, Walter S Lasecki, Jonathan K Kummerfeld, Michel Galley, Chris Brockett, et al. 2019. Dialog System Technology Challenge 7. *arXiv preprint arXiv:1901.03461* (2019).

[383] Yang Yu, Wei Zhang, Kazi Hasan, Mo Yu, Bing Xiang, and Bowen Zhou. 2016. End-to-end Answer Chunk Extraction and Ranking for Reading Comprehension. *arXiv preprint arXiv:1610.09996* (2016).

[384] Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots. In *EMNLP*. 111–120.

[385] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big Bird: Transformers for Longer Sequences. *NeurIPS*, 17283–17297.

[386] Hamed Zamani and W Bruce Croft. 2020. Learning a Joint Search and Recommendation Model from User-Item Interactions. In *CIKM*. 717–725.

[387] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From Neural Re-ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. In *CIKM*. 497–506.

[388] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating Clarifying Questions for Information Retrieval. In *TheWebConf*. 418–428.

[389] Hamed Zamani, Johanne R Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational information seeking. *arXiv preprint arXiv:2201.08808* (2022).

[390] Oleg Zendel, Anna Shtok, Fiana Raiber, Oren Kurland, and J Shane Culpepper. 2019. Information Needs, Queries, and Query Performance Prediction. In *SIGIR*. 395–404.

[391] Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum Learning for Dense Retrieval Distillation. *arXiv preprint arXiv:2204.13679* (2022).

[392] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *SIGIR*. 1503–1512.

[393] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An Analysis of BERT in Document Ranking. In *SIGIR*. 1941–1944.

[394] Dakun Zhang, Jungi Kim, Josep Crego, and Jean Senellart. 2017. Boosting Neural Machine Translation. In *IJCNLP*. 271–276.

[395] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. In *ICLR*.

[396] Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An Empirical Exploration of Curriculum Learning for Neural Machine Translation. *arXiv preprint arXiv:1811.00739* (2018).

[397] Yu Zhang and Qiang Yang. 2017. A Survey on Multi-Task Learning. *arXiv:1707.08114* (2017).

[398] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling Multi-turn Conversation with Deep Utterance Aggregation. In *ACL*. 3740–3752.

[399] Zhuosheng Zhang and Hai Zhao. 2021. Advances in Multi-Turn Dialogue Comprehension: A Survey. *arXiv preprint arXiv:2110.04984* (2021).

[400] Kun Zhou, Yeyun Gong, Xiao Liu, Wayne Xin Zhao, Yelong Shen, Anlei Dong, Jingwen Lu, Rangan Majumder, Ji-Rong Wen, Nan Duan, et al. 2022. SimANS: Simple Ambiguous Negatives Sampling for Dense Text Retrieval. *arXiv preprint arXiv:2210.11773* (2022).

[401] Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A Dataset for Document Grounded Conversations. In *EMNLP*. 708–713.

[402] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn Response Selection for Chatbots with Deep Attention Matching Cetwork. In *ACL*. 1118–1127.

[403] Jianhan Zhu, Jun Wang, Ingemar J Cox, and Michael J Taylor. 2009. Risky Business: Modeling and Exploiting Uncertainty in Information Retrieval. In *SIGIR*. 99–106.

[404] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading books. In *ICCV*. 19–27.

[405] Shengyao Zhuang and Guido Zuccon. 2021. Dealing with Typos for BERT-based Passage Retrieval and Ranking. *arXiv preprint arXiv:2108.12139* (2021).

[406] Shengyao Zhuang and Guido Zuccon. 2021. Fast Passage Re-ranking with Contextualized Exact Term Matching and Efficient Passage Expansion. *arXiv preprint arXiv:2108.08513* (2021).

[407] Shengyao Zhuang and Guido Zuccon. 2022. CharacterBERT and Self-Teaching for Improving the Robustness of Dense Retrievers on Queries with Typos. *arXiv preprint arXiv:2204.00716* (2022).

[408] Julia El Zini and Mariette Awad. 2022. On the explainability of natural language processing deep models. *Comput. Surveys* (2022), 1–31.

[409] Guido Zuccon, Leif Azzopardi, and Keith van Rijsbergen. 2011. Back to the Roots: Mean-Variance Analysis of Relevance Estimations. In *ECIR*. Springer, 716–720.

[410] Guido Zuccon, Joao Palotti, and Allan Hanbury. 2016. Query Variations and Their Effect on Comparing Information Retrieval Systems. In *CIKM*. 691–700.

List of Figures

List of Tables

Curriculum Vitæ

Experience and Education

2023-1–now	Research scientist at Spotify.
2022-6–8	Research internship at Spotify.
2021-5–8	Research internship at Amazon.
2015–2018	Data scientist at Hekima.
2016–2018	M.Sc. in Computer Science, UFMG.
2014–2015	Undergraduate researcher at LBD, UFMG.
2012–2016	B.Sc. in Computer Science, UFMG.

Publications

TheWebConf 2023	Improving Content Retrievability in Search with Controllable Query Generation **Gustavo Penha**, Enrico Palumbo, Maryam Aziz, Alice Wang, Hughes Bouchard
ECIR 2023	📄 Do the Findings of Document and Passage Retrieval Generalize to the Retrieval of Responses for Dialogues? **Gustavo Penha**, Claudia Hauff
CHI 2022	Helping Voice Shoppers Make Purchase Decisions **Gustavo Penha**, Eyal Krikon, Vanessa Murdock, Sandeep Avula
CHIIR 2022	Pairwise review-based explanations for voice product search **Gustavo Penha**, Eyal Krikon, Vanessa Murdock
ECIR 2022	📄 🏆 Evaluating the robustness of retrieval pipelines with query variation generators **Gustavo Penha**, Arthur Câmara, Claudia Hauff
EACL 2021	📄 On the Calibration and Uncertainty of Neural Learning to Rank Models for Conversational Search **Gustavo Penha**, Claudia Hauff
ECIR 2021	Weakly Supervised Label Smoothing **Gustavo Penha**, Claudia Hauff

SCAI 2020	Slice-aware Neural Ranking **Gustavo Penha**, Claudia Hauff
RecSys 2020	🏆 Exploiting Performance Estimates for Augmenting Recommendation Ensembles **Gustavo Penha**, Rodrygo Santos
RecSys 2020	📄 What does BERT know about books, movies and music? Probing BERT for Conversational Recommendation **Gustavo Penha**, Claudia Hauff
ECIR 2020	📄 Curriculum Learning Strategies for IR **Gustavo Penha**, Claudia Hauff
CONVERSE 2020	📄 Challenges in The Evaluation of Conversational Search Systems **Gustavo Penha**, Claudia Hauff
CAIR 2020	Domain Adaptation for Conversation Response Ranking **Gustavo Penha**, Claudia Hauff
Arxiv 2019	📄 Introducing MANtIS: a novel multi-domain information seeking dialogues dataset **Gustavo Penha**, Alex Balan Claudia Hauff
ECIR 2019	Document Performance Prediction for Automatic Text Classification **Gustavo Penha**, Raphael Campos, Sérgio Canuto, Marcos Gonçalves, Rodrygo L. T. Santos

📄 Included in this thesis.
🏆 Won award.

SIKS Dissertation Series

Since 1998, all dissertations written by PhD. students who have conducted their research under auspices of a senior research fellow of the SIKS research school are published in the SIKS Dissertation Series.

==== **1998** ====

1998-1 Johan van den Akker (CWI) DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM) Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD) A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective

1998-4 Dennis Breuker (UM) Memory versus Search in Games

1998-5 E.W.Oskamp (RUL) Computerondersteuning bij Straftoemeting

==== **1999** ====

1999-1 Mark Sloof (VU) Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR) Classification using decision trees and neural nets

1999-3 Don Beal (UM) The Nature of Minimax Search

1999-4 Jacques Penders (UM) The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB) Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU) Re-design of compositional systems

1999-7 David Spelt (UT) Verification support for object database design

1999-8 Jacques H.J. Lenting (UM) Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.

==== **2000** ====

2000-1 Frank Niessink (VU) Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE) Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA) Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU) ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM) Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU) Programming Languages for Agent Communication

2000-7 Niels Peek (UU) Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coupé (EUR) Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI) Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI) Image Database Management System Design Considerations, Algorithms and Architecture

2000-11 Jonas Karlsson (CWI) Scalable Distributed Data Structures for Database Management

==== **2001** ====

2001-1 Silja Renooij (UU) Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU) Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA) Learning as problem solving

2001-4 Evgueni Smirnov (UM) Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU) Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU) Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU) Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU) A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL) Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA) Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design

2001-11 Tom M. van Engers (VUA) Knowledge Management: The Role of Mental Models in Business Systems Design

==== **2002** ====

2002-01 Nico Lassing (VU) Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT) Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT) Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU) The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU) The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL) Applied legal epistemology; Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI) Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU) Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB) Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM) Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU) Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva) Processing XML in Database Systems

2002-13 Hongjing Wu (TUE) A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU) Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT) Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU) The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA) Understanding, Modeling, and Improving Main-Memory Database Performance

==== **2003** ====

2003-01 Heiner Stuckenschmidt (VU) Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU) Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD) Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT) Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA) Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT) Development and specification of virtual environments

2003-07 Machiel Jansen (UvA) Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM) Repair Based Scheduling

2003-09 Rens Kortmann (UM) The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT) Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT) Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT) Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM) Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN) Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD) Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI) Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses

2003-17 David Jansen (UT) Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM) Learning Search Decisions

==== **2004** ====

2004-01 Virginia Dignum (UU) A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT) Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU) A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA) Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR) Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD) The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM) Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes

2004-08 Joop Verbeek(UM) Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU) For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA) Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU) Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT) Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT) Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU) Logic in Conflict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU) Multi-Relational Data Mining

2004-16 Federico Divina (VU) Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM) Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA) Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (UT) Using generative probabilistic models for multimedia retrieval

2004-20 Madelon Evers (Nyenrode) Learning from Design: facilitating multidisciplinary design teams

==== **2005** ====

2005-01 Floor Verdenius (UVA) Methodological Aspects of Designing Induction-Based Applications

2005-02 Erik van der Werf (UM)) AI techniques for the game of Go

2005-03 Franc Grootjen (RUN) A Pragmatic Approach to the Conceptualisation of Language

2005-04 Nirvana Meratnia (UT) Towards Database Support for Moving Object data

2005-05 Gabriel Infante-Lopez (UVA) Two-Level Probabilistic Grammars for Natural Language Parsing

2005-06 Pieter Spronck (UM) Adaptive Game AI

2005-07 Flavius Frasincar (TUE) Hypermedia Presentation Generation for Semantic Web Information Systems

2005-08 Richard Vdovjak (TUE) A Model-driven Approach for Building Distributed Ontology-based Web Applications

2005-09 Jeen Broekstra (VU) Storage, Querying and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UVA) Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments

2005-11 Elth Ogston (VU) Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR) Distributed Simulation in Industry

2005-13 Fred Hamburg (UL) Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU) Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics

2005-15 Tibor Bosse (VU) Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU) Usability of XML Query Languages

2005-17 Boris Shishkov (TUD) Software Specification Based on Re-usable Business Components

2005-18 Danielle Sent (UU) Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM) Situated Representation

2005-20 Cristina Coteanu (UL) Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT) Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

==== **2006** ====

2006-01 Samuil Angelov (TUE) Foundations of B2B Electronic Contracting

2006-02 Cristina Chisalita (VU) Contextual issues in the design and use of information technology in organizations

2006-03 Noor Christoph (UVA) The role of metacognitive skills in learning to solve problems

2006-04 Marta Sabou (VU) Building Web Service Ontologies

2006-05 Cees Pierik (UU) Validation Techniques for Object-Oriented Proof Outlines

2006-06 Ziv Baida (VU) Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling

2006-07 Marko Smiljanic (UT) XML schema matching – balancing efficiency and effectiveness by means of clustering

2006-08 Eelco Herder (UT) Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09 Mohamed Wahdan (UM) Automatic Formulation of the Auditor's Opinion

2006-10 Ronny Siebes (VU) Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT) Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU) Interactivation - Towards an e-cology of people, our technological environment, and the arts

2006-13 Henk-Jan Lebbink (UU) Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU) Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change

2006-15 Rainer Malik (UU) CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU) Approximation Methods for Efficient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU) User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UVA) Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU) Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT) Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN) Aptness on the Web

2006-22 Paul de Vrieze (RUN) Fundaments of Adaptive Personalisation

2006-23 Ion Juvina (UU) Development of Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU) Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU) Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlovic (UT) Score Region Algebra: A Flexible Framework for Structured Information Retrieval

2006-27 Stefano Bocconi (CWI) Vox Populi: generating video documentaries from semantically annotated media repositories

2006-28 Borkur Sigurbjornsson (UVA) Focused Information Access using XML Element Retrieval

==== **2007** ====

2007-01 Kees Leune (UvT) Access Control and Service-Oriented Architectures

2007-02 Wouter Teepe (RUG) Reconciling Information Exchange and Confidentiality: A Formal Approach

2007-03 Peter Mika (VU) Social Networks and the Semantic Web

2007-04 Jurriaan van Diggelen (UU) Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach

2007-05 Bart Schermer (UL) Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance

2007-06 Gilad Mishne (UVA) Applied Text Analytics for Blogs

2007-07 Natasa Jovanovic' (UT) To Whom It May Concern - Addressee Identification in Face-to-Face Meetings

2007-08 Mark Hoogendoorn (VU) Modeling of Change in Multi-Agent Organizations

2007-09 David Mobach (VU) Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU) Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols

2007-11 Natalia Stash (TUE) Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

2007-12 Marcel van Gerven (RUN) Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty

2007-13 Rutger Rienks (UT) Meetings in Smart Environments; Implications of Progressing Technology

2007-14 Niek Bergboer (UM) Context-Based Image Analysis

2007-15 Joyca Lacroix (UM) NIM: a Situated Computational Memory Model

2007-16 Davide Grossi (UU) Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems

2007-17 Theodore Charitos (UU) Reasoning with Dynamic Networks in Practice

2007-18 Bart Orriens (UvT) On the development an management of adaptive business collaborations

2007-19 David Levy (UM) Intimate relationships with artificial partners

2007-20 Slinger Jansen (UU) Customer Configuration Updating in a Software Supply Network

2007-21 Karianne Vermaas (UU) Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005

2007-22 Zlatko Zlatev (UT) Goal-oriented design of value and process models from patterns

2007-23 Peter Barna (TUE) Specification of Application Logic in Web Information Systems

2007-24 Georgina Ramírez Camps (CWI) Structural Features in XML Retrieval

2007-25 Joost Schalken (VU) Empirical Investigations in Software Process Improvement

==== **2008** ====

2008-01 Katalin Boer-Sorbán (EUR) Agent-Based Simulation of Financial Markets: A modular, continuous-time approach

2008-02 Alexei Sharpanskykh (VU) On Computer-Aided Methods for Modeling and Analysis of Organizations

2008-03 Vera Hollink (UVA) Optimizing hierarchical menus: a usage-based approach

2008-04 Ander de Keijzer (UT) Management of Uncertain Data - towards unattended integration

2008-05 Bela Mutschler (UT) Modeling and simulating causal dependencies on process-aware information systems from a cost perspective

2008-06 Arjen Hommersom (RUN) On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective

2008-07 Peter van Rosmalen (OU) Supporting the tutor in the design and support of adaptive e-learning

2008-08 Janneke Bolt (UU) Bayesian Networks: Aspects of Approximate Inference

2008-09 Christof van Nimwegen (UU) The paradox of the guided user: assistance can be counter-effective

2008-10 Wauter Bosma (UT) Discourse oriented summarization

2008-11 Vera Kartseva (VU) Designing Controls for Network Organizations: A Value-Based Approach

2008-12 Jozsef Farkas (RUN) A Semiotically Oriented Cognitive Model of Knowledge Representation

2008-13 Caterina Carraciolo (UVA) Topic Driven Access to Scientific Handbooks

2008-14 Arthur van Bunningen (UT) Context-Aware Querying; Better Answers with Less Effort

2008-15 Martijn van Otterlo (UT) The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.

2008-16 Henriette van Vugt (VU) Embodied agents from a user's perspective

2008-17 Martin Op 't Land (TUD) Applying Architecture and Ontology to the Splitting and Allying of Enterprises

2008-18 Guido de Croon (UM) Adaptive Active Vision

2008-19 Henning Rode (UT) From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search

2008-20 Rex Arendsen (UVA) Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven.

2008-21 Krisztian Balog (UVA) People Search in the Enterprise

2008-22 Henk Koning (UU) Communication of IT-Architecture

2008-23 Stefan Visscher (UU) Bayesian network models for the management of ventilator-associated pneumonia

2008-24 Zharko Aleksovski (VU) Using background knowledge in ontology matching

2008-25 Geert Jonker (UU) Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency

2008-26 Marijn Huijbregts (UT) Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled

2008-27 Hubert Vogten (OU) Design and Implementation Strategies for IMS Learning Design

2008-28 Ildiko Flesch (RUN) On the Use of Independence Relations in Bayesian Networks

2008-29 Dennis Reidsma (UT) Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans

2008-30 Wouter van Atteveldt (VU) Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content

2008-31 Loes Braun (UM) Pro-Active Medical Information Retrieval

2008-32 Trung H. Bui (UT) Toward Affective Dialogue Management using Partially Observable Markov Decision Processes

2008-33 Frank Terpstra (UVA) Scientific Workflow Design; theoretical and practical issues

2008-34 Jeroen de Knijf (UU) Studies in Frequent Tree Mining

2008-35 Ben Torben Nielsen (UvT) Dendritic morphologies: function shapes structure

==== **2009** ====

2009-01 Rasa Jurgelenaite (RUN) Symmetric Causal Independence Models

2009-02 Willem Robert van Hage (VU) Evaluating Ontology-Alignment Techniques

2009-03 Hans Stol (UvT) A Framework for Evidence-based Policy Making Using IT

2009-04 Josephine Nabukenya (RUN) Improving the Quality of Organisational Policy Making using Collaboration Engineering

2009-05 Sietse Overbeek (RUN) Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality

2009-06 Muhammad Subianto (UU) Understanding Classification

2009-07 Ronald Poppe (UT) Discriminative Vision-Based Recovery and Recognition of Human Motion

2009-08 Volker Nannen (VU) Evolutionary Agent-Based Policy Analysis in Dynamic Environments

2009-09 Benjamin Kanagwa (RUN) Design, Discovery and Construction of Service-oriented Systems

2009-10 Jan Wielemaker (UVA) Logic programming for knowledge-intensive interactive applications

2009-11 Alexander Boer (UVA) Legal Theory, Sources of Law & the Semantic Web

2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin) Operating Guidelines for Services

2009-13 Steven de Jong (UM) Fairness in Multi-Agent Systems

2009-14 Maksym Korotkiy (VU) From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)

2009-15 Rinke Hoekstra (UVA) Ontology Representation - Design Patterns and Ontologies that Make Sense

2009-16 Fritz Reul (UvT) New Architectures in Computer Chess

2009-17 Laurens van der Maaten (UvT) Feature Extraction from Visual Data

2009-18 Fabian Groffen (CWI) Armada, An Evolving Database System

2009-19 Valentin Robu (CWI) Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets

2009-20 Bob van der Vecht (UU) Adjustable Autonomy: Controling Influences on Decision Making

2009-21 Stijn Vanderlooy (UM) Ranking and Reliable Classification

2009-22 Pavel Serdyukov (UT) Search For Expertise: Going beyond direct evidence

2009-23 Peter Hofgesang (VU) Modelling Web Usage in a Changing Environment

2009-24 Annerieke Heuvelink (VUA) Cognitive Models for Training Simulations

2009-25 Alex van Ballegooij (CWI) "RAM: Array Database Management through Relational Mapping"

2009-26 Fernando Koch (UU) An Agent-Based Model for the Development of Intelligent Mobile Services

2009-27 Christian Glahn (OU) Contextual Support of social Engagement and Reflection on the Web

2009-28 Sander Evers (UT) Sensor Data Management with Probabilistic Models

2009-29 Stanislav Pokraev (UT) Model-Driven Semantic Integration of Service-Oriented Applications

2009-30 Marcin Zukowski (CWI) Balancing vectorized query execution with bandwidth-optimized storage

2009-31 Sofiya Katrenko (UVA) A Closer Look at Learning Relations from Text

2009-32 Rik Farenhorst (VU) and Remco de Boer (VU) Architectural Knowledge Management: Supporting Architects and Auditors

2009-33 Khiet Truong (UT) How Does Real Affect Affect Affect Recognition In Speech?

2009-34 Inge van de Weerd (UU) Advancing in Software Product Management: An Incremental Method Engineering Approach

2009-35 Wouter Koelewijn (UL) Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling

2009-36 Marco Kalz (OUN) Placement Support for Learners in Learning Networks

2009-37 Hendrik Drachsler (OUN) Navigation Support for Learners in Informal Learning Networks

2009-38 Riina Vuorikari (OU) Tags and self-organisation: a metadata ecology for learning resources in a multilingual context

2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin) Service Substitution – A Behavioral Approach Based on Petri Nets

2009-40 Stephan Raaijmakers (UvT) Multinomial Language Learning: Investigations into the Geometry of Language

2009-41 Igor Berezhnyy (UvT) Digital Analysis of Paintings

2009-42 Toine Bogers (UvT) Recommender Systems for Social Bookmarking

2009-43 Virginia Nunes Leal Franqueira (UT) Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients

2009-44 Roberto Santana Tapia (UT) Assessing Business-IT Alignment in Networked Organizations

2009-45 Jilles Vreeken (UU) Making Pattern Mining Useful

2009-46 Loredana Afanasiev (UvA) Querying XML: Benchmarks and Recursion

==== **2010** ====

2010-01 Matthijs van Leeuwen (UU) Patterns that Matter

2010-02 Ingo Wassink (UT) Work flows in Life Science

2010-03 Joost Geurts (CWI) A Document Engineering Model and Processing Framework for Multimedia documents

2010-04 Olga Kulyk (UT) Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments

2010-05 Claudia Hauff (UT) Predicting the Effectiveness of Queries and Retrieval Systems

2010-06 Sander Bakkes (UvT) Rapid Adaptation of Video Game AI

2010-07 Wim Fikkert (UT) Gesture interaction at a Distance

2010-08 Krzysztof Siewicz (UL) Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments

2010-09 Hugo Kielman (UL) A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging

2010-10 Rebecca Ong (UL) Mobile Communication and Protection of Children

2010-11 Adriaan Ter Mors (TUD) The world according to MARP: Multi-Agent Route Planning

2010-12 Susan van den Braak (UU) Sensemaking software for crime analysis

2010-13 Gianluigi Folino (RUN) High Performance Data Mining using Bio-inspired techniques

2010-14 Sander van Splunter (VU) Automated Web Service Reconfiguration

2010-15 Lianne Bodenstaff (UT) Managing Dependency Relations in Inter-Organizational Models

2010-16 Sicco Verwer (TUD) Efficient Identification of Timed Automata, theory and practice

2010-17 Spyros Kotoulas (VU) Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications

2010-18 Charlotte Gerritsen (VU) Caught in the Act: Investigating Crime by Agent-Based Simulation

2010-19 Henriette Cramer (UvA) People's Responses to Autonomous and Adaptive Systems

2010-20 Ivo Swartjes (UT) Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative

2010-21 Harold van Heerde (UT) Privacy-aware data management by means of data degradation

2010-22 Michiel Hildebrand (CWI) End-user Support for Access to
Heterogeneous Linked Data

2010-23 Bas Steunebrink (UU) The Logical Structure of Emotions

2010-24 Dmytro Tykhonov Designing Generic and Efficient Negotiation Strategies

2010-25 Zulfiqar Ali Memon (VU) Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective

2010-26 Ying Zhang (CWI) XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines

2010-27 Marten Voulon (UL) Automatisch contracteren

2010-28 Arne Koopman (UU) Characteristic Relational Patterns

2010-29 Stratos Idreos(CWI) Database Cracking: Towards Auto-tuning Database Kernels

2010-30 Marieke van Erp (UvT) Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval

2010-31 Victor de Boer (UVA) Ontology Enrichment from Heterogeneous Sources on the Web

2010-32 Marcel Hiel (UvT) An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems

2010-33 Robin Aly (UT) Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval

2010-34 Teduh Dirgahayu (UT) Interaction Design in Service Compositions

2010-35 Dolf Trieschnigg (UT) Proof of Concept: Concept-based Biomedical Information Retrieval

2010-36 Jose Janssen (OU) Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification

2010-37 Niels Lohmann (TUE) Correctness of services and their composition

2010-38 Dirk Fahland (TUE) From Scenarios to components

2010-39 Ghazanfar Farooq Siddiqui (VU) Integrative modeling of emotions in virtual agents

2010-40 Mark van Assem (VU) Converting and Integrating Vocabularies for the Semantic Web

2010-41 Guillaume Chaslot (UM) Monte-Carlo Tree Search

2010-42 Sybren de Kinderen (VU) Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach

2010-43 Peter van Kranenburg (UU) A Computational Approach to Content-Based Retrieval of Folk Song Melodies

2010-44 Pieter Bellekens (TUE) An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain

2010-45 Vasilios Andrikopoulos (UvT) A theory and model for the evolution of software services

2010-46 Vincent Pijpers (VU) e3alignment: Exploring Inter-Organizational Business-ICT Alignment

2010-47 Chen Li (UT) Mining Process Model Variants: Challenges, Techniques, Examples

2010-48 Withdrawn

2010-49 Jahn-Takeshi Saito (UM) Solving difficult game positions

2010-50 Bouke Huurnink (UVA) Search in Audiovisual Broadcast Archives

2010-51 Alia Khairia Amin (CWI) Understanding and supporting information seeking tasks in multiple sources

2010-52 Peter-Paul van Maanen (VU) Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention

2010-53 Edgar Meij (UVA) Combining Concepts and Language Models for Information Access

==== **2011** ====

2011-01 Botond Cseke (RUN) Variational Algorithms for Bayesian Inference in Latent Gaussian Models

2011-02 Nick Tinnemeier(UU) Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

2011-03 Jan Martijn van der Werf (TUE) Compositional Design and Verification of Component-Based Information Systems

2011-04 Hado van Hasselt (UU) Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms

2011-05 Base van der Raadt (VU) Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

2011-06 Yiwen Wang (TUE) Semantically-Enhanced Recommendations in Cultural Heritage

2011-07 Yujia Cao (UT) Multimodal Information Presentation for High Load Human Computer Interaction

2011-08 Nieske Vergunst (UU) BDI-based Generation of Robust Task-Oriented Dialogues

2011-09 Tim de Jong (OU) Contextualised Mobile Media for Learning

2011-10 Bart Bogaert (UvT) Cloud Content Contention

2011-11 Dhaval Vyas (UT) Designing for Awareness: An Experience-focused HCI Perspective

2011-12 Carmen Bratosin (TUE) Grid Architecture for Distributed Process Mining

2011-13 Xiaoyu Mao (UvT) Airport under Control. Multi-agent Scheduling for Airport Ground Handling

2011-14 Milan Lovric (EUR) Behavioral Finance and Agent-Based Artificial Markets

2011-15 Marijn Koolen (UvA) The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16 Maarten Schadd (UM) Selective Search in Games of Different Complexity

2011-17 Jiyin He (UVA) Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18 Mark Ponsen (UM) Strategic Decision-Making in complex games

2011-19 Ellen Rusman (OU) The Mind ' s Eye on Personal Profiles

2011-20 Qing Gu (VU) Guiding service-oriented software engineering - A view-based approach

2011-21 Linda Terlouw (TUD) Modularization and Specification of Service-Oriented Systems

2011-22 Junte Zhang (UVA) System Evaluation of Archival Description and Access

2011-23 Wouter Weerkamp (UVA) Finding People and their Utterances in Social Media

2011-24 Herwin van Welbergen (UT) Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25 Syed Waqar ul Qounain Jaffry (VU)) Analysis and Validation of Models for Trust Dynamics

2011-26 Matthijs Aart Pontier (VU) Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27 Aniel Bhulai (VU) Dynamic website optimization through autonomous management of design patterns

2011-28 Rianne Kaptein(UVA) Effective Focused Retrieval by Exploiting Query Context and Document Structure

2011-29 Faisal Kamiran (TUE) Discrimination-aware Classification

2011-30 Egon van den Broek (UT) Affective Signal Processing (ASP): Unraveling the mystery of emotions

2011-31 Ludo Waltman (EUR) Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

2011-32 Nees-Jan van Eck (EUR) Methodological Advances in Bibliometric Mapping of Science

2011-33 Tom van der Weide (UU) Arguing to Motivate Decisions

2011-34 Paolo Turrini (UU) Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

2011-35 Maaike Harbers (UU) Explaining Agent Behavior in Virtual Training

2011-36 Erik van der Spek (UU) Experiments in serious game design: a cognitive approach

2011-37 Adriana Burlutiu (RUN) Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

2011-38 Nyree Lemmens (UM) Bee-inspired Distributed Optimization

2011-39 Joost Westra (UU) Organizing Adaptation using Agents in Serious Games

2011-40 Viktor Clerc (VU) Architectural Knowledge Management in Global Software Development

2011-41 Luan Ibraimi (UT) Cryptographically Enforced Distributed Data Access Control

2011-42 Michal Sindlar (UU) Explaining Behavior through Mental State Attribution

2011-43 Henk van der Schuur (UU) Process Improvement through Software Operation Knowledge

2011-44 Boris Reuderink (UT) Robust Brain-Computer Interfaces

2011-45 Herman Stehouwer (UvT) Statistical Language Models for Alternative Sequence Selection

2011-46 Beibei Hu (TUD) Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

2011-47 Azizi Bin Ab Aziz(VU) Exploring Computational Models for Intelligent Support of Persons with Depression

2011-48 Mark Ter Maat (UT) Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

2011-49 Andreea Niculescu (UT) Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

==== **2012** ====

2012-01 Terry Kakeeto (UvT) Relationship Marketing for SMEs in Uganda

2012-02 Muhammad Umair(VU) Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

2012-03 Adam Vanya (VU) Supporting Architecture Evolution by Mining Software Repositories

2012-04 Jurriaan Souer (UU) Development of Content Management System-based Web Applications

2012-05 Marijn Plomp (UU) Maturing Interorganisational Information Systems

2012-06 Wolfgang Reinhardt (OU) Awareness Support for Knowledge Workers in Research Networks

2012-07 Rianne van Lambalgen (VU) When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions

2012-08 Gerben de Vries (UVA) Kernel Methods for Vessel Trajectories

2012-09 Ricardo Neisse (UT) Trust and Privacy Management Support for Context-Aware Service Platforms

2012-10 David Smits (TUE) Towards a Generic Distributed Adaptive Hypermedia Environment

2012-11 J.C.B. Rantham Prabhakara (TUE) Process Mining in the Large: Preprocessing, Discovery, and Diagnostics

2012-12 Kees van der Sluijs (TUE) Model Driven Design and Data Integration in Semantic Web Information Systems

2012-13 Suleman Shahid (UvT) Fun and Face: Exploring non-verbal expressions of emotion during playful interactions

2012-14 Evgeny Knutov(TUE) Generic Adaptation Framework for Unifying Adaptive Web-based Systems

2012-15 Natalie van der Wal (VU) Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.

2012-16 Fiemke Both (VU) Helping people by understanding them - Ambient Agents supporting task execution and depression treatment

2012-17 Amal Elgammal (UvT) Towards a Comprehensive Framework for Business Process Compliance

2012-18 Eltjo Poort (VU) Improving Solution Architecting Practices

2012-19 Helen Schonenberg (TUE) What's Next? Operational Support for Business Process Execution

2012-20 Ali Bahramisharif (RUN) Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing

2012-21 Roberto Cornacchia (TUD) Querying Sparse Matrices for Information Retrieval

2012-22 Thijs Vis (UvT) Intelligence, politie en veiligheidsdienst: verenigbare grootheden?

2012-23 Christian Muehl (UT) Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction

2012-24 Laurens van der Werff (UT) Evaluation of Noisy Transcripts for Spoken Document Retrieval

2012-25 Silja Eckartz (UT) Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application

2012-26 Emile de Maat (UVA) Making Sense of Legal Text

2012-27 Hayrettin Gurkok (UT) Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games

2012-28 Nancy Pascall (UvT) Engendering Technology Empowering Women

2012-29 Almer Tigelaar (UT) Peer-to-Peer Information Retrieval

2012-30 Alina Pommeranz (TUD) Designing Human-Centered Systems for Reflective Decision Making

2012-31 Emily Bagarukayo (RUN) A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure

2012-32 Wietske Visser (TUD) Qualitative multi-criteria preference representation and reasoning

2012-33 Rory Sie (OUN) Coalitions in Cooperation Networks (COCOON)

2012-34 Pavol Jancura (RUN) Evolutionary analysis in PPI networks and applications

2012-35 Evert Haasdijk (VU) Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics

2012-36 Denis Ssebugwawo (RUN) Analysis and Evaluation of Collaborative Modeling Processes

2012-37 Agnes Nakakawa (RUN) A Collaboration Process for Enterprise Architecture Creation

2012-38 Selmar Smit (VU) Parameter Tuning and Scientific Testing in Evolutionary Algorithms

2012-39 Hassan Fatemi (UT) Risk-aware design of value and coordination networks

2012-40 Agus Gunawan (UvT) Information Access for SMEs in Indonesia

2012-41 Sebastian Kelle (OU) Game Design Patterns for Learning

2012-42 Dominique Verpoorten (OU) Reflection Amplifiers in self-regulated Learning

2012-43 Withdrawn

2012-44 Anna Tordai (VU) On Combining Alignment Techniques

2012-45 Benedikt Kratz (UvT) A Model and Language for Business-aware Transactions

2012-46 Simon Carter (UVA) Exploration and Exploitation of Multilingual Data for Statistical Machine Translation

2012-47 Manos Tsagkias (UVA) Mining Social Media: Tracking Content and Predicting Behavior

2012-48 Jorn Bakker (TUE) Handling Abrupt Changes in Evolving Time-series Data

2012-49 Michael Kaisers (UM) Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions

2012-50 Steven van Kervel (TUD) Ontologogy driven Enterprise Information Systems Engineering

2012-51 Jeroen de Jong (TUD) Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

==== **2013** ====

2013-01 Viorel Milea (EUR) News Analytics for Financial Decision Support

2013-02 Erietta Liarou (CWI) MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing

2013-03 Szymon Klarman (VU) Reasoning with Contexts in Description Logics

2013-04 Chetan Yadati(TUD) Coordinating autonomous planning and scheduling

2013-05 Dulce Pumareja (UT) Groupware Requirements Evolutions Patterns

2013-06 Romulo Goncalves(CWI) The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience

2013-07 Giel van Lankveld (UvT) Quantifying Individual Player Differences

2013-08 Robbert-Jan Merk(VU) Making enemies: cognitive modeling for opponent agents in fighter pilot simulators

2013-09 Fabio Gori (RUN) Metagenomic Data Analysis: Computational Methods and Applications

2013-10 Jeewanie Jayasinghe Arachchige(UvT) A Unified Modeling Framework for Service Design.

2013-11 Evangelos Pournaras(TUD) Multi-level Reconfigurable Self-organization in Overlay Services

2013-12 Marian Razavian(VU) Knowledge-driven Migration to Services

2013-13 Mohammad Safiri(UT) Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly

2013-14 Jafar Tanha (UVA) Ensemble Approaches to Semi-Supervised Learning Learning

2013-15 Daniel Hennes (UM) Multiagent Learning - Dynamic Games and Applications

2013-16 Eric Kok (UU) Exploring the practical benefits of argumentation in multi-agent deliberation

2013-17 Koen Kok (VU) The PowerMatcher: Smart Coordination for the Smart Electricity Grid

2013-18 Jeroen Janssens (UvT) Outlier Selection and One-Class Classification

2013-19 Renze Steenhuizen (TUD) Coordinated Multi-Agent Planning and Scheduling

2013-20 Katja Hofmann (UvA) Fast and Reliable Online Learning to Rank for Information Retrieval

2013-21 Sander Wubben (UvT) Text-to-text generation by monolingual machine translation

2013-22 Tom Claassen (RUN) Causal Discovery and Logic

2013-23 Patricio de Alencar Silva(UvT) Value Activity Monitoring

2013-24 Haitham Bou Ammar (UM) Automated Transfer in Reinforcement Learning

2013-25 Agnieszka Anna Latoszek-Berendsen (UM) Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System

2013-26 Alireza Zarghami (UT) Architectural Support for Dynamic Homecare Service Provisioning

2013-27 Mohammad Huq (UT) Inference-based Framework Managing Data Provenance

2013-28 Frans van der Sluis (UT) When Complexity becomes Interesting: An Inquiry into the Information eXperience

2013-29 Iwan de Kok (UT) Listening Heads

2013-30 Joyce Nakatumba (TUE) Resource-Aware Business Process Management: Analysis and Support

2013-31 Dinh Khoa Nguyen (UvT) Blueprint Model and Language for Engineering Cloud Applications

2013-32 Kamakshi Rajagopal (OUN) Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development

2013-33 Qi Gao (TUD) User Modeling and Personalization in the Microblogging Sphere

2013-34 Kien Tjin-Kam-Jet (UT) Distributed Deep Web Search

2013-35 Abdallah El Ali (UvA) Minimal Mobile Human Computer Interaction Promotor: Prof. dr. L. Hardman (CWI/UVA)

2013-36 Than Lam Hoang (TUe) Pattern Mining in Data Streams

2013-37 Dirk Börner (OUN) Ambient Learning Displays

2013-38 Eelco den Heijer (VU) Autonomous Evolutionary Art

2013-39 Joop de Jong (TUD) A Method for Enterprise Ontology based Design of Enterprise Information Systems

2013-40 Pim Nijssen (UM) Monte-Carlo Tree Search for Multi-Player Games

2013-41 Jochem Liem (UVA) Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning

2013-42 Léon Planken (TUD) Algorithms for Simple Temporal Reasoning

2013-43 Marc Bron (UVA) Exploration and Contextualization through Interaction and Concepts

==== **2014** ====

2014-01 Nicola Barile (UU) Studies in Learning Monotone Models from Data

2014-02 Fiona Tuliyano (RUN) Combining System Dynamics with a Domain Modeling Method

2014-03 Sergio Raul Duarte Torres (UT) Information Retrieval for Children: Search Behavior and Solutions

2014-04 Hanna Jochmann-Mannak (UT) Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation

2014-05 Jurriaan van Reijsen (UU) Knowledge Perspectives on Advancing Dynamic Capability

2014-06 Damian Tamburri (VU) Supporting Networked Software Development

2014-07 Arya Adriansyah (TUE) Aligning Observed and Modeled Behavior

2014-08 Samur Araujo (TUD) Data Integration over Distributed and Heterogeneous Data Endpoints

2014-09 Philip Jackson (UvT) Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language

2014-10 Ivan Salvador Razo Zapata (VU) Service Value Networks

2014-11 Janneke van der Zwaan (TUD) An Empathic Virtual Buddy for Social Support

2014-12 Willem van Willigen (VU) Look Ma, No Hands: Aspects of Autonomous Vehicle Control

2014-13 Arlette van Wissen (VU) Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains

2014-14 Yangyang Shi (TUD) Language Models With Meta-information

2014-15 Natalya Mogles (VU) Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare

2014-16 Krystyna Milian (VU) Supporting trial recruitment and design by automatically interpreting eligibility criteria

2014-17 Kathrin Dentler (VU) Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability

2014-18 Mattijs Ghijsen (UVA) Methods and Models for the Design and Study of Dynamic Agent Organizations

2014-19 Vinicius Ramos (TUE) Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support

2014-20 Mena Habib (UT) Named Entity Extraction and Disambiguation for Informal Text: The Missing Link

2014-21 Kassidy Clark (TUD) Negotiation and Monitoring in Open Environments

2014-22 Marieke Peeters (UU) Personalized Educational Games - Developing agent-supported scenario-based training

2014-23 Eleftherios Sidirourgos (UvA/CWI) Space Efficient Indexes for the Big Data Era

2014-24 Davide Ceolin (VU) Trusting Semi-structured Web Data

2014-25 Martijn Lappenschaar (RUN) New network models for the analysis of disease interaction

2014-26 Tim Baarslag (TUD) What to Bid and When to Stop

2014-27 Rui Jorge Almeida (EUR) Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty

2014-28 Anna Chmielowiec (VU) Decentralized k-Clique Matching

2014-29 Jaap Kabbedijk (UU) Variability in Multi-Tenant Enterprise Software

2014-30 Peter de Cock (UvT) Anticipating Criminal Behaviour

2014-31 Leo van Moergestel (UU) Agent Technology in Agile Multiparallel Manufacturing and Product Support

2014-32 Naser Ayat (UvA) On Entity Resolution in Probabilistic Data

2014-33 Tesfa Tegegne (RUN) Service Discovery in eHealth

2014-34 Christina Manteli(VU) The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.

2014-35 Joost van Ooijen (UU) Cognitive Agents in Virtual Worlds: A Middleware Design Approach

2014-36 Joos Buijs (TUE) Flexible Evolutionary Algorithms for Mining Structured Process Models

2014-37 Maral Dadvar (UT) Experts and Machines United Against Cyberbullying

2014-38 Danny Plass-Oude Bos (UT) Making brain-computer interfaces better: improving usability through post-processing.

2014-39 Jasmina Maric (UvT) Web Communities, Immigration, and Social Capital

2014-40 Walter Omona (RUN) A Framework for Knowledge Management Using ICT in Higher Education

2014-41 Frederic Hogenboom (EUR) Automated Detection of Financial Events in News Text

2014-42 Carsten Eijckhof (CWI/TUD) Contextual Multidimensional Relevance Models

2014-43 Kevin Vlaanderen (UU) Supporting Process Improvement using Method Increments

2014-44 Paulien Meesters (UvT) Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.

2014-45 Birgit Schmitz (OUN) Mobile Games for Learning: A Pattern-Based Approach

2014-46 Ke Tao (TUD) Social Web Data Analytics: Relevance, Redundancy, Diversity

2014-47 Shangsong Liang (UVA) Fusion and Diversification in Information Retrieval

==== **2015** ====

2015-01 Niels Netten (UvA) Machine Learning for Relevance of Information in Crisis Response

2015-02 Faiza Bukhsh (UvT) Smart auditing: Innovative Compliance Checking in Customs Controls

2015-03 Twan van Laarhoven (RUN) Machine learning for network data

2015-04 Howard Spoelstra (OUN) Collaborations in Open Learning Environments

2015-05 Christoph Bösch (UT) Cryptographically Enforced Search Pattern Hiding

2015-06 Farideh Heidari (TUD) Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes

2015-07 Maria-Hendrike Peetz(UvA) Time-Aware Online Reputation Analysis

2015-08 Jie Jiang (TUD) Organizational Compliance: An agent-based model for designing and evaluating organizational interactions

2015-09 Randy Klaassen(UT) HCI Perspectives on Behavior Change Support Systems

2015-10 Henry Hermans (OUN) OpenU: design of an integrated system to support lifelong learning

2015-11 Yongming Luo(TUE) Designing algorithms for big graph datasets: A study of computing bisimulation and joins

2015-12 Julie M. Birkholz (VU) Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks

2015-13 Giuseppe Procaccianti(VU) Energy-Efficient Software

2015-14 Bart van Straalen (UT) A cognitive approach to modeling bad news conversations

2015-15 Klaas Andries de Graaf (VU) Ontology-based Software Architecture Documentation

2015-16 Changyun Wei (UT) Cognitive Coordination for Cooperative Multi-Robot Teamwork

2015-17 André van Cleeff (UT) Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs

2015-18 Holger Pirk (CWI) Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories

2015-19 Bernardo Tabuenca (OUN) Ubiquitous Technology for Lifelong Learners

2015-20 Loïs Vanhée (UU) Using Culture and Values to Support Flexible Coordination

2015-21 Sibren Fetter (OUN) Using Peer-Support to Expand and Stabilize Online Learning

015-22 Zhemin Zhu (UT) Co-occurrence Rate Networks

2015-23 Luit Gazendam (VU) Cataloguer Support in Cultural Heritage

2015-24 Richard Berendsen (UVA) Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation

2015-25 Steven Woudenberg (UU) Bayesian Tools for Early Disease Detection

2015-26 Alexander Hogenboom (EUR) Sentiment Analysis of Text Guided by Semantics and Structure

2015-27 Sándor Héman (CWI) Updating compressed colomn stores

2015-28 Janet Bagorogoza(TiU) KNOWLEDGE MANAGEMENT AND HIGH PERFORMANCE; The Uganda Financial Institutions Model for HPO

2015-29 Hendrik Baier (UM) Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains

2015-30 Kiavash Bahreini(OU) Real-time Multimodal Emotion Recognition in E-Learning

2015-31 Yakup Koç (TUD) On the robustness of Power Grids

2015-32 Jerome Gard(UL) Corporate Venture Management in SMEs

2015-33 Frederik Schadd (TUD) Ontology Mapping with Auxiliary Resources

2015-34 Victor de Graaf(UT) Gesocial Recommender Systems

2015-35 Jungxao Xu (TUD) Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

==== **2016** ====

2016-01 Syed Saiden Abbas (RUN) Recognition of Shapes by Humans and Machines

2016-02 Michiel Christiaan Meulendijk (UU) Optimizing medication reviews through decision support: prescribing a better pill to swallow

2016-03 Maya Sappelli (RUN) Knowledge Work in Context: User Centered Knowledge Worker Support

2016-04 Laurens Rietveld (VU) Publishing and Consuming Linked Data

2016-05 Evgeny Sherkhonov (UVA) Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

2016-06 Michel Wilson (TUD) Robust scheduling in an uncertain environment

2016-07 Jeroen de Man (VU) Measuring and modeling negative emotions for virtual training

2016-08 Matje van de Camp (TiU) A Link to the Past: Constructing Historical Social Networks from Unstructured Data

2016-09 Archana Nottamkandath (VU) Trusting Crowdsourced Information on Cultural Artefacts

2016-10 George Karafotias (VUA) Parameter Control for Evolutionary Algorithms

2016-11 Anne Schuth (UVA) Search Engines that Learn from Their Users

2016-12 Max Knobbout (UU) Logics for Modelling and Verifying Normative Multi-Agent Systems

2016-13 Nana Baah Gyan (VU) The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

2016-14 Ravi Khadka (UU) Revisiting Legacy Software System Modernization

2016-15 Steffen Michels (RUN) Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

2016-16 Guangliang Li (UVA) Socially Intelligent Autonomous Agents that Learn from Human Reward

2016-17 Berend Weel (VU) Towards Embodied Evolution of Robot Organisms

2016-18 Albert Meroño Peñuela (VU) Refining Statistical Data on the Web

2016-19 Julia Efremova (Tu/e) Mining Social Structures from Genealogical Data

2016-20 Daan Odijk (UVA) Context & Semantics in News & Web Search

2016-21 Alejandro Moreno Celleri (UT) From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground

2016-22 Grace Lewis (VU) Software Architecture Strategies for Cyber-Foraging Systems

2016-23 Fei Cai (UVA) Query Auto Completion in Information Retrieval

2016-24 Brend Wanders (UT) Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach

2016-25 Julia Kiseleva (TU/e) Using Contextual Information to Understand Searching and Browsing Behavior

2016-26 Dilhan Thilakarathne (VU) In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains

2016-27 Wen Li (TUD) Understanding Geo-spatial Information on Social Media

2016-28 Mingxin Zhang (TUD) Large-scale Agent-based Social Simulation - A study on epidemic prediction and control

2016-29 Nicolas Höning (TUD) Peak reduction in decentralised electricity systems -Markets and prices for flexible planning

2016-30 Ruud Mattheij (UvT) The Eyes Have It

2016-31 Mohammad Khelghati (UT) Deep web content monitoring

2016-32 Eelco Vriezekolk (UT) Assessing Telecommunication Service Availability Risks for Crisis Organisations

2016-33 Peter Bloem (UVA) Single Sample Statistics, exercises in learning from just one example

2016-34 Dennis Schunselaar (TUE) Configurable Process Trees: Elicitation, Analysis, and Enactment

2016-35 Zhaochun Ren (UVA) Monitoring Social Media: Summarization, Classification and Recommendation

2016-36 Daphne Karreman (UT) Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies

2016-37 Giovanni Sileno (UvA) Aligning Law and Action - a conceptual and computational inquiry

2016-38 Andrea Minuto (UT) MATERIALS THAT MATTER - Smart Materials meet Art & Interaction Design

2016-39 Merijn Bruijnes (UT) Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect

2016-40 Christian Detweiler (TUD) Accounting for Values in Design

2016-41 Thomas King (TUD) Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

2016-42 Spyros Martzoukos (UVA) Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

2016-43 Saskia Koldijk (RUN) Context-Aware Support for Stress Self-Management: From Theory to Practice

2016-44 Thibault Sellam (UVA) Automatic Assistants for Database Exploration

2016-45 Bram van de Laar (UT) Experiencing Brain-Computer Interface Control

2016-46 Jorge Gallego Perez (UT) Robots to Make you Happy

2016-47 Christina Weber (UL) Real-time foresight - Preparedness for dynamic innovation networks

2016-48 Tanja Buttler (TUD) Collecting Lessons Learned

2016-49 Gleb Polevoy (TUD) Participation and Interaction in Projects. A Game-Theoretic Analysis

2016-50 Yan Wang (UVT) The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

==== **2017** ====

2017-01 Jan-Jaap Oerlemans (UL) Investigating Cybercrime

2017-02 Sjoerd Timmer (UU) Designing and Understanding Forensic Bayesian Networks using Argumentation

2017-03 Daniël Harold Telgen (UU) Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines

2017-04 Mrunal Gawade (CWI) MULTI-CORE PARALLELISM IN A COLUMN-STORE

2017-05 Mahdieh Shadi (UVA) Collaboration Behavior

2017-06 Damir Vandic (EUR) Intelligent Information Systems for Web Product Search

2017-07 Roel Bertens (UU) Insight in Information: from Abstract to Anomaly

2017-08 Rob Konijn (VU) Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery

2017-09 Dong Nguyen (UT) Text as Social and Cultural Data: A Computational Perspective on Variation in Text

2017-10 Robby van Delden (UT) (Steering) Interactive Play Behavior

2017-11 Florian Kunneman (RUN) Modelling patterns of time and emotion in Twitter #anticipointment

2017-12 Sander Leemans (TUE) Robust Process Mining with Guarantees

2017-13 Gijs Huisman (UT) Social Touch Technology - Extending the reach of social touch through haptic technology

2017-14 Shoshannah Tekofsky (UvT) You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior

2017-15 Peter Berck, Radboud University (RUN) Memory-Based Text Correction

2017-16 Aleksandr Chuklin (UVA) Understanding and Modeling Users of Modern Search Engines

2017-17 Daniel Dimov (UL) Crowdsourced Online Dispute Resolution

2017-18 Ridho Reinanda (UVA) Entity Associations for Search

2017-19 Jeroen Vuurens (TUD) Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

2017-20 Mohammadbashir Sedighi (TUD) Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

2017-21 Jeroen Linssen (UT) Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)

2017-22 Sara Magliacane (VU) Logics for causal inference under uncertainty

2017-23 David Graus (UVA) Entities of Interest— Discovery in Digital Traces

2017-24 Chang Wang (TUD) Use of Affordances for Efficient Robot Learning

2017-25 Veruska Zamborlini (VU) Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search

2017-26 Merel Jung (UT) Socially intelligent robots that understand and respond to human touch

2017-27 Michiel Joosse (UT) Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors

2017-28 John Klein (VU) Architecture Practices for Complex Contexts

2017-29 Adel Alhuraibi (UVT) From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT

2017-30 Wilma Latuny (UVT) The Power of Facial Expressions

2017-31 Ben Ruijl (UL) Advances in computational methods for QFT calculations

2017-32 Thaer Samar (RUN) Access to and Retrievability of Content in Web Archives

2017-33 Brigit van Loggem (OU) Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity

2017-34 Maren Scheffel (OUN) The Evaluation Framework for Learning Analytics

2017-35 Martine de Vos (VU) Interpreting natural science spreadsheets

2017-36 Yuanhao Guo (UL) Shape Analysis for Phenotype Characterisation from High-throughput Imaging

2017-37 Alejandro Montes García (TUE) WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy

2017-38 Alex Kayal (TUD) Normative Social Applications

2017-39 Sara Ahmadi (RUN) Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR

2017-40 Altaf Hussain Abro (VUA) Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems”

2017-41 Adnan Manzoor (VUA) Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle

2017-42 Elena Sokolova (RUN) Causal discovery from mixed and missing data with applications on ADHD datasets

2017-43 Maaike de Boer (RUN) Semantic Mapping in Video Retrieval

2017-44 Garm Lucassen (UU) Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

2017-45 Bas Testerink (UU) Decentralized Runtime Norm Enforcement

2017-46 Jan Schneider (OU) Sensor-based Learning Support

2017-47 Yie Yang (TUD) Crowd Knowledge Creation Acceleration

2017-48 Angel Suarez (OU) Collaborative inquiry-based learning

==== **2018** ====

2018-01 Han van der Aa (VUA) Comparing and Aligning Process Representations

2018-02 Felix Mannhardt (TUE) Multi-perspective Process Mining

2018-03 Steven Bosems (UT) Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

2018-04 Jordan Janeiro (TUD) Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

2018-05 Hugo Huurdeman (UVA) Supporting the Complex Dynamics of the Information Seeking Process

2018-06 Dan Ionita (UT) Model-Driven Information Security Risk Assessment of Socio-Technical Systems

2018-07 Jieting Luo (UU) A formal account of opportunism in multi-agent systems

2018-08 Rick Smetsers (RUN) Advances in Model Learning for Software Systems

2018-09 Xu Xie (TUD) Data Assimilation in Discrete Event Simulations

2018-10 Julienka Mollee (VUA) Moving forward: supporting physical activity behavior change through intelligent technology

2018-11 Mahdi Sargolzaei (UVA) Enabling Framework for Service-oriented Collaborative Networks

2018-12 Xixi Lu (TUE) Using behavioral context in process mining

2018-13 Seyed Amin Tabatabaei (VUA) Using behavioral context in process mining: Exploring the added value of computational models for increasing the use of renewable energy in the residential sector

2018-14 Bart Joosten (UVT) Detecting Social Signals with Spatiotemporal Gabor Filters

2018-15 Naser Davarzani (UM) Biomarker discovery in heart failure

2018-16 Jaebok Kim (UT) Automatic recognition of engagement and emotion in a group of children

2018-17 Jianpeng Zhang (TUE) On Graph Sample Clustering

2018-18 Henriette Nakad (UL) De Notaris en Private Rechtspraak

2018-19 Minh Duc Pham (VUA) Emergent relational schemas for RDF

2018-20 Manxia Liu (RUN) Time and Bayesian Networks

2018-21 Aad Slootmaker (OUN) EMERGO: a generic platform for authoring and playing scenario-based serious games

2018-22 Eric Fernandes de Mello Araújo (VUA) Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks

2018-23 Kim Schouten (EUR) Semantics-driven Aspect-Based Sentiment Analysis

2018-24 Jered Vroon (UT) Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots

2018-25 Riste Gligorov (VUA) Serious Games in Audio-Visual Collections

2018-26 Roelof de Vries (UT) Theory-Based And Tailor-Made: Motivational Messages for Behavior Change Technology

2018-27 Maikel Leemans (TUE) Hierarchical Process Mining for Scalable Software Analysis

2018-28 Christian Willemse (UT) Social Touch Technologies: How they feel and how they make you feel

2018-29 Yu Gu (UVT) Emotion Recognition from Mandarin Speech

2018-30 Wouter Beek (VU) The ”K” in ”semantic web” stands for ”knowledge”: scaling semantics to the web

==== **2019** ====

2019-01 Rob van Eijk (UL) Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification

2019-02 Emmanuelle Beauxis- Aussalet (CWI, UU) Statistics and Visualizations for Assessing Class Size Uncertainty

2019-03 Eduardo Gonzalez Lopez de Murillas (TUE) Process Mining on Databases: Extracting Event Data from Real Life Data Sources

2019-04 Ridho Rahmadi (RUN) Finding stable causal structures from clinical data

2019-05 Sebastiaan van Zelst (TUE) Process Mining with Streaming Data

2019-06 Chris Dijkshoorn (VU) Nichesourcing for Improving Access to Linked Cultural Heritage Datasets

2019-07 Soude Fazeli (TUD) Recommender Systems in Social Learning Platforms

2019-08 Frits de Nijs (TUD) Resource-constrained Multi-agent Markov Decision Processes

2019-09 Fahimeh Alizadeh Moghaddam (UVA) Self-adaptation for energy efficiency in software systems

2019-10 Qing Chuan Ye (EUR) Multi-objective Optimization Methods for Allocation and Prediction

2019-11 Yue Zhao (TUD) Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs

2019-12 Jacqueline Heinerman (VU) Better Together

2019-13 Guanliang Chen (TUD) MOOC Analytics: Learner Modeling and Content Generation

2019-14 Daniel Davis (TUD) Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

2019-15 Erwin Walraven (TUD) Planning under Uncertainty in Constrained and Partially Observable Environments

2019-16 Guangming Li (TUE) Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models

2019-17 Ali Hurriyetoglu (RUN) Extracting actionable information from microtexts

2019-18 Gerard Wagenaar (UU) Artefacts in Agile Team Communication

2019-19 Vincent Koeman (TUD) Tools for Developing Cognitive Agents

2019-20 Chide Groenouwe (UU) Fostering technically augmented human collective intelligence

2019-21 Cong Liu (TUE) Software Data Analytics: Architectural Model Discovery and Design Pattern Detection

2019-22 Martin van den Berg (VU) Improving IT Decisions with Enterprise Architecture

2019-23 Qin Liu (TUD) Intelligent Control Systems: Learning, Interpreting, Verification

2019-24 Anca Dumitrache (VU) Truth in Disagreement-Crowdsourcing Labeled Data for Natural Language Processing

2019-25 Emiel van Miltenburg (UVT) Pragmatic factors in (automatic) image description

2019-26 Prince Singh (UT) An Integration Platform for Synchromodal Transport

2019-27 Alessandra Antonaci (OUN) The Gamification Design Process applied to (Massive) Open Online Courses

2019-28 Esther Kuindersma (UL) Cleared for take-off: Game-based learning to prepare airline pilots for critical situations

2019-29 Daniel Formolo (VU) Using virtual agents for simulation and training of social skills in safety-critical circumstances

2019-30 Vahid Yazdanpanah (UT) Multiagent Industrial Symbiosis Systems

2019-31 Milan Jelisavcic (VUA) Alive and Kicking: Baby Steps in Robotics

2019-32 Chiara Sironi (UM) Monte-Carlo Tree Search for Artificial General Intelligence in Games

2019-33 Anil Yaman (TUE) Evolution of Biologically Inspired Learning in Artificial Neural Networks

2019-34 Negar Ahmadi (TUE) EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES

2019-35 Lisa Facey-Shaw (OUN) Gamification with digital badges in learning programming

2019-36 Kevin Ackermans (OUN) Designing Video-Enhanced Rubrics to Master Complex Skills

2019-37 Jian Fang (TUD) Database Acceleration on FPGAs

2019-38 Akos Kadar (OUN) Learning visually grounded and multilingual representations

==== **2020** ====

2020-01 Armon Toubman (UL) Calculated Moves: Generating Air Combat Behaviour

2020-02 Marcos de Paula Bueno (UL) Unraveling Temporal Processes using Probabilistic Graphical Models

2020-03 Mostafa Deghani (UvA) Learning with Imperfect Supervision for Language Understanding

2020-04 Maarten van Gompel (RUN) Context as Linguistic Bridges

2020-05 Yulong Pei (TUE) On local and global structure mining

2020-06 Preethu Rose Anish (UT) Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support

2020-07 Wim van der Vegt (OUN) Towards a software architecture for reusable game components

2020-08 Ali Mirsoleimani (UL) Structured Parallel Programming for Monte Carlo Tree Search

2020-09 Myriam Traub (UU) Measuring Tool Bias & Improving Data Quality for Digital Humanities Research

2020-10 Alifah Syamsiyah (TUE) In-database Preprocessing for Process Mining

2020-11 Sepideh Mesbah (TUD) Semantic-Enhanced Training Data Augmentation Methods for Long-Tail Entity Recognition Models

2020-12 Ward van Breda (VU) Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment

2020-13 Marco Virgolin (CWI) Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming

2020-14 Mark Raasveldt (CWI/UL) Integrating Analytics with Relational Databases

2020-15 Konstantinos Georgiadis (OU) Smart CAT: Machine Learning for Configurable Assessments in Serious Games

2020-16 Ilona Wilmont (RUN) Cognitive Aspects of Conceptual Modelling

2020-17 Daniele Di Mitri (OU) The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences

2020-18 Georgios Methenitis (TUD) Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems

2020-19 Guido van Capelleveen (UT) Industrial Symbiosis Recommender Systems

2020-20 Albert Hankel (VU) Embedding Green ICT Maturity in Organisations

2020-21 Karine da Silva Miras de Araujo (VU) Where is the robot?: Life as it could be

2020-22 Maryam Masoud Khamis (RUN) Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar

2020-23 Rianne Conijn (UT) The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging

2020-24 Lenin da Nobrega Medeiros (VUA/RUN) How are you feeling, human? Towards emotionally supportive chatbots

2020-25 Xin Du (TUE) The Uncertainty in Exceptional Model Mining

2020-26 Krzysztof Leszek Sadowski (UU) GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization

2020-27 Ekaterina Muravyeva (TUD) Personal data and informed consent in an educational context

2020-28 Bibeg Limbu (TUD) Multimodal interaction for deliberate practice: Training complex skills with augmented reality

2020-29 Ioan Gabriel Bucur (RUN) Being Bayesian about Causal Inference

2020-30 Bob Zadok Blok (UL) Creatief, Creatieve, Creatiefst

2020-31 Gongjin Lan (VU) Learning better – From Baby to Better

2020-32 Jason Rhuggenaath (TUE) Revenue management in online markets: pricing and online advertising

2020-33 Rick Gilsing (TUE) Supporting service-dominant business model evaluation in the context of business model innovation

2020-34 Anna Bon (MU) Intervention or Collaboration? Redesigning Information and Communication Technologies for Development

2020-35 Siamak Farshidi (UU) Multi-Criteria Decision-Making in Software Production

==== **2021** ====

2021-01 Francisco Xavier Dos Santos Fonseca (TUD) Location-based Games for Social Interaction in Public Space

2021-02 Rijk Mercuur (TUD) Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models

2021-03 Seyyed Hadi Hashemi (UVA) Modeling Users Interacting with Smart Devices

2021-04 Ioana Jivet (OU) The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning

2021-05 Davide Dell'Anna (UU) Data-Driven Supervision of Autonomous Systems

2021-06 Daniel Davison (UT) ”Hey robot, what do you think?” How children learn with a social robot

2021-07 Armel Lefebvre (UU) Research data management for open science

2021-08 Nardie Fanchamps (OU) The Influence of Sense-Reason-Act Programming on Computational Thinking

2021-09 Cristina Zaga (UT) The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play

2021-10 Quinten Meertens (UvA) Misclassification Bias in Statistical Learning

2021-11 Anne van Rossum (UL) Nonparametric Bayesian Methods in Robotic Vision

2021-12 Lei Pi (UL) External Knowledge Absorption in Chinese SMEs

2021-13 Bob R. Schadenberg (UT) Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning

2021-14 Negin Samaeemofrad (UL) Business Incubators: The Impact of Their Support

2021-15 Onat Ege Adali (TU/e) Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm

2021-16 Esam A. H. Ghaleb (MU) BIMODAL EMOTION RECOGNITION FROM AUDIO-VISUAL CUES

2021-17 Dario Dotti (UM) Human Behavior Understanding from motion and bodily cues using deep neural networks

2021-18 Remi Wieten (UU) Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks

2021-19 Roberto Verdecchia (VU) Architectural Technical Debt: Identification and Management

2021-20 Masoud Mansoury (TU/e) Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems

2021-21 Pedro Thiago Timbó Holanda (CWI) Progressive Indexes

2021-22 Sihang Qiu (TUD) Conversational Crowdsourcing

2021-23 Hugo Manuel Proença (LIACS) Robust rules for prediction and description

2021-24 Kaijie Zhu (TUE) On Efficient Temporal Subgraph Query Processing

2021-25 Eoin Martino Grua (VUA) The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

2021-26 Benno Kruit (CWI & VU) Reading the Grid: Extending Knowledge Bases from Human-readable Tables

2021-27 Jelte van Waterschoot (UT) Personalized and Personal Conversations: Designing Agents Who Want to Connect With You

2021-28 Christoph Selig (UL) Understanding the Heterogeneity of Corporate Entrepreneurship Programs

==== **2022** ====

2022-1 Judith van Stegeren (UT) Flavor text generation for role-playing video games

2022-2 Paulo da Costa (TU/e) Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey

2022-3 Ali el Hassouni (VUA) A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare

2022-4 Ünal Aksu (UU) A Cross-Organizational Process Mining Framework

2022-5 Shiwei Liu (TU/e) Sparse Neural Network Training with In-Time Over-Parameterization

2022-6 Reza Refaei Afshar (TU/e) Machine Learning for Ad Publishers in Real Time Bidding

2022-7 Sambit Praharaj (OU) Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics

2022-8 Maikel L. van Eck (TU/e) Process Mining for Smart Product Design

2022-9 Oana Andreea Inel (VUA) Understanding Events: A Diversity-driven Human-Machine Approach

2022-10 Felipe Moraes Gomes (TUD) Examining the Effectiveness of Collaborative Search Engines

2022-11 Mirjam de Haas (UT) Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers engagement with robots and tasks during second-language tutoring

2022-12 Guanyi Chen (UU) Computational Generation of Chinese Noun Phrases

2022-13 Xander Wilcke (VUA) Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented

2022-14 Michiel Overeem (UU) Evolution of Low-Code Platforms

2022-15 Jelmer Jan Koorn (UU) Work in Process: Unearthing Meaning using Process Mining

2022-16 Pieter Gijsbers (TU/e) Systems for AutoML Research

2022-17 Laura van der Lubbe (VUA) Empowering vulnerable people with serious games and gamification

2022-18 Mavromoustakos Blom (TiU) Player Affect Modelling and Video Game Personalisation

2022-19 Bilge Yigit Ozkan (UU) Cybersecurity Maturity Assessment and Standardisation

2022-20 Fakhra Jabeen (VUA) Dark Side of the Digital World - Computational Analysis of Negative Human Behaviors on Social Media

2022-21 Seethu Mariyam Christopher (UM) Intelligent Toys for Physical and Cognitive Assessments

2022-22 Alexandra Sierra Rativa (TiU) Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations

2022-23 Ilir Kola (TUD) Enabling Social Situation Awareness in Support Agents

2022-24 Samaneh Heidari (UU) Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values

2022-25 Anna L.D. Latour (LU) Optimal decision-making under constraints and uncertainty

2022-26 Anne Dirkson (LU) Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences

2022-27 Christos Athanasiadis (UM) Emotion-aware cross-modal domain adaptation in video sequences

2022-28 Onuralp Ulusoy (UU) Privacy in Collaborative Systems

2022-29 Jan Kolkmeier (UT-EEMCS) From Head Transform to Mind Transplant: Social Interactions in Mixed Reality

2022-30 Dean De Leo (CWI) Analysis of Dynamic Graphs on Sparse Arrays

2022-31 Konstantinos Traganos (TU/e) Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management

2022-32 Cezara Pastrav (UU) Social simulation for socio-ecological systems

2022-33 Brinn Hekkelman (CWI/TUD) Fair Mechanisms for Smart Grid Congestion Management

2022-34 Nimat Ullah (VUA) Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change

2022-35 Mike E.U. Ligthart (VU) Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction

==== **2023** ====

2023-01 Bojan Simoski (VUA) Untangling the Puzzle of Digital Health Interventions

2023-02 Mariana Rachel Dias da Silva (TiU) Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts

2023-03 Shabnam Najafian (TU Delft) User Modeling for Privacy-preserving Explanations in Group Recommendations

2023-04 Gineke Wiggers (Leiden University) The Relevance of Impact: bibliometric-enhanced legal information retrieval

2023-05 P.A. (Anton) Bouter (CWI) Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization Including Real-World Medical Applications

2023-06 António Pereira Barata (Leiden University) Reliable and Fair Machine Learning for Risk Assessment

2023-07 Tianjin Huang (TU/e) The Roles of Adversarial Examples on Trustworthiness of Deep Learning

2023-08 Lu Yin (TU/e) Knowledge Elicitation using Psychometric Learning

2023-09 Xu Wang (VUA) Scientific Dataset Recommendation with Semantic Techniques

2023-10 Dennis J.N.J. Soemers (UM) Learning State-Action Features for General Game Playing

2023-11 Fawad Taj (VUA) Towards Motivating Machines: Computational Modeling of the Mechanism of Actions for Effective Digital Health Behavior Change Applications

2023-12 Tessel Bogaard (VUA) Using Metadata to Understand Search Behavior in Digital Libraries

2023-13 Injy Sarhan (UU) Open Information Extraction for Knowledge Representation

2023-14 Selma Čaušević (TU Delft) Energy resilience through self-organization

2023-15 Alvaro Henrique Chaim Correia (TU/e) Insights on Learning Tractable Probabilistic Graphical Models

2023-16 Peter Blomsma (TiU) Building Embodied Conversational Agents: Observations on human nonverbal behaviour as a resource for the development of artificial characters

2023-17 Meike Nauta (UT) Explainable AI and Interpretable Computer Vision – From Oversight to Insight