

Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signed by:
Signature: 
FFE1CA66F218421...

Harshita Jagdish Sahijwani
Name

8/7/2024 | 9:13 PM EDT
Date

Title Intent Prediction and User Preference Modeling in Conversational Search and Recommendation

Author Harshita Jagdish Sahijwani

Degree Doctor of Philosophy


Program Computer Science and Informatics

Approved by the Committee


DocuSigned by:

C81F95707F73454...

Eugene Agichtein
Advisor

DocuSigned by:

C5A3D3420EBA406...

Jinho D. Choi
Committee Member

DocuSigned by:

34E5DE72DFD0456...

Joyce C Ho
Committee Member

DocuSigned by:

74C55B8F6C6C43E...

Vanessa Murdock
Committee Member

Committee Member

Committee Member

Accepted by the Laney Graduate School:

Kimberly Jacob Arriola, Ph.D, MPH
Dean, James T. Laney Graduate School

Date

Intent Prediction and User Preference Modeling in Conversational Search
and Recommendation

By

Harshita Jagdish Sahijwani

Advisor: Eugene Agichtein, PhD

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science
2024

Abstract

Intent Prediction and User Preference Modeling in Conversational Search and Recommendation
By Harshita Jagdish Sahijwani

Conversational systems have emerged as potent tools for increasing the accessibility of user-facing applications, in particular, search and recommendation. This dissertation addresses two key challenges in conversational systems: intent prediction and user preference modeling.

Identifying the intent of each user utterance in conversational systems is a crucial step for all subsequent language understanding and response tasks. The high cost of collecting conversational training data makes the task of intent prediction challenging. The first main research question addressed in this dissertation is: Can we use external knowledge and synthetic data to improve intent prediction? I propose methods for knowledge-aware intent prediction in three settings, including open-domain social bots, conversational information elicitation systems, and web-based domain-specific search systems. In addition, I study the impact of synthetic data on intent prediction in these systems.

User preference modeling is another essential part of effective conversational systems. The second main research question addressed in this dissertation is: Can we anticipate the user's next topic of interest by constructing a user profile using conversation context? I propose methods to represent the user based on the conversation history. Moreover, a sequence modeling approach is proposed to predict the user's next topic of interest in conversational systems.

Despite the capacity of large language models to implicitly perform end-to-end intent detection and user preference modeling, they are not universally applicable. They cannot be used with private data. Moreover, modular systems with specialized components allow for more interpretability and control over the system. Systems with modules for intent detection and user preference modeling are thus still relevant.

Together, the proposed methods enable a better understanding of the user's immediate needs and long-term preferences in all types of conversational systems. The findings of this research hold implications for improving the accuracy and performance of conversational search and recommender systems.

Intent Prediction and User Preference Modeling in Conversational Search
and Recommendation

By

Harshita Jagdish Sahijwani

Advisor: Eugene Agichtein, PhD

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science
2024

Intent Prediction and User Preference Modeling in Conversational Search and Recommendation

Harshita Jagdish Sahijwani
Advisor: Eugene Agichtein, Ph.D.
Computer Science Department
Emory University
hsahijw@emory.edu

August 9, 2024

Abstract

Conversational systems have emerged as potent tools for increasing the accessibility of user-facing applications, in particular, search and recommendation. This dissertation addresses two key challenges in conversational systems: intent prediction and user preference modeling.

Identifying the intent of each user utterance in conversational systems is a crucial step for all subsequent language understanding and response tasks. The high cost of collecting conversational training data makes the task of intent prediction challenging. The first main research question addressed in this dissertation is: Can we use external knowledge and synthetic data to improve intent prediction? I propose methods for knowledge-aware intent prediction in three settings, including open-domain social bots, conversational information elicitation systems, and web-based domain-specific search systems. In addition, I study the impact of synthetic data on intent prediction in these systems.

User preference modeling is another essential part of effective conversational systems. The second main research question addressed in this dissertation is: Can we anticipate the user's next topic of interest by constructing a user profile using conversation context? I propose methods to represent the user based on the conversation history. Moreover, a sequence modeling approach is proposed to predict the user's next topic of interest in conversational systems.

Despite the capacity of large language models to implicitly perform end-to-end intent detection and user preference modeling, they are not universally applicable. They cannot be used with private data. Moreover, modular systems with specialized components allow for more interpretability and control over the system. Systems with modules for intent detection and user preference modeling are thus still relevant.

Together, the proposed methods enable a better understanding of the user's immediate needs and long-term preferences in all types of conversational systems. The findings of this research hold implications for improving the accuracy and performance of conversational search and recommender systems.

Contents

1	Introduction	4
2	Related Work	7
2.1	Intent Prediction	7
2.1.1	Intent Prediction in Open-Domain Conversational Systems	7
2.1.2	Intent Prediction in Information Elicitation Conversational Systems	8
2.1.3	Intent Prediction in Web Search	9
2.2	Conversational Search	9
2.2.1	Entity linking	10
2.3	Entity-based Text Representation	11
2.4	Conversational Systems for Interviewing Users	11
2.5	Knowledge Injection	12
2.6	Dialog Management	13
2.7	Clarification Question Generation	13
2.8	Conversational Recommendation	15
2.9	Summary	15
3	Intent Detection in Conversational Search and Recommendation	16
3.1	ConCET: Entity-Aware Intent Classification for Open-Domain Conversational Agents	16
3.1.1	Introduction	17
3.1.2	ConCET System Overview	18
3.1.3	Conversational Entity Linking	19
3.1.4	ConCET: Concurrent Entity-Aware Intent Classifier	22
3.1.5	Conversational Dataset Overview	25
3.1.6	Experimental Setup	28
3.1.7	Results and Discussion	29
3.1.8	Limitations	34
3.1.9	Conclusion	34
3.2	Contextual Intent Classification for Automated Structured Interviews	34
3.2.1	Introduction	35
3.2.2	Data Collection	37
3.2.3	Methods	38
3.2.4	Experimental Setting	40
3.2.5	Results and Discussion	41
3.2.6	Limitations	44

4	Query Intent Classification for Domain-Specific Web Search	45
4.1	Introduction	45
4.2	Methodology	46
4.2.1	Knowledge Injection	46
4.2.2	Data Augmentation	49
4.3	Experimental Setting: Data and Methods Compared	50
4.3.1	Data	50
4.3.2	Knowledge Base and Representation	52
4.3.3	Validation Experiments	53
4.3.4	Methods Compared	54
4.3.5	Evaluation Metrics	54
4.4	Results	55
4.4.1	Main Results	55
4.4.2	Data Augmentation Results	55
4.4.3	In-Context Learning Results	55
4.5	Discussion	57
4.5.1	Ground Truth Data	57
4.5.2	Entity Linking	58
4.6	Limitations and Future Work	58
4.7	Conclusion	58
4.8	Future Work	58
5	User Modeling	59
5.1	User Modeling for Topic Recommendation in Open-Domain Conversational System	59
5.1.1	Introduction	59
5.1.2	Conversational Topic Suggestion (CTS): Problem Definition	60
5.1.3	CTS-Seq Approach	62
5.1.4	CTS-Seq: Models	63
5.1.5	Experimental Setup	67
5.1.6	Results and Discussion	71
6	Discussion and Conclusions	76
6.1	RQ1: Intent Prediction	76
6.1.1	RQ1a	76
6.1.2	RQ1b	76
6.1.3	RQ1c	77
6.1.4	Relevance in the LLM Era	78
6.2	RQ2: User Modeling	79
6.2.1	RQ2a	79
6.3	Limitations	79
6.4	Conclusions	80
A	Prompts for Experiments with LLMs	81
A.1	Prompts for Synthetic Data Generation	81
A.1.1	Baseline	81
A.1.2	Entity-based	82
A.2	Prompts for In-Context Learning	82

	3
B Additional Results	84
Bibliography	85

1 Introduction

Conversational Systems have gained significant popularity as tools for performing everyday tasks. Recent advancements in technology have led to the development of conversational agents like ChatGPT ¹ and Gemini ², which leverage large language models to provide information from the web in a conversational manner. Voice-based personal assistants like Alexa and Google Home are capable of fulfilling a range of user requests, including package tracking, making reservations, answering questions, and controlling home automation devices. Furthermore, embodied conversational agents are promising tools to make interactions with machines even more natural and human-like. In this thesis, I focus on conversational search (CSI) and conversational recommender systems (CRS). Specifically, I study the intent prediction and user preference modeling tasks in various conversational settings.

An IR system needs to understand both the user’s current needs and long-term preferences well to produce relevant and personalized results. Effective methods for performing these tasks have greatly improved the performance of web search and recommendation systems. However, for conversational systems, intent understanding and user preference modeling have not yet been perfected.

While end-to-end conversational systems have majorly improved in their capabilities, there are specific applications where providing meticulously curated responses and generating follow-up questions based on predefined logic is crucial. For example, it is not feasible to risk providing incorrect information in a medical setting. Generative models that have even a slight possibility of hallucinating facts are not suitable for such applications. In such cases, modular systems with specialized components for intent classification and user modeling are commonly employed in practice. Modular systems are also useful when training data is available for individual components. A modular design enables building a system that is a combination of specialized supervised models [101]. This thesis focuses on conversational systems with dedicated modules for intent recognition and user preference modeling.

Intent prediction and user modeling are more challenging in the conversational setting because of the following reasons:

1. Conversational training data is expensive to collect and is not easily available, especially in the case of conversational information seeking and conversational recommendations. Moreover, the set of possible conversations is so large that it is not possible to collect a comprehensive set of dialogs.
2. Conversational data is much harder to interpret. Click data, although noisy, is a rich source of user preference information that is generally not available in the conversational setting. The system only has the conversational context to understand the user’s needs.

This thesis addresses the complementary problems of intent classification and implicit user preference modeling. I propose methods for incorporating external knowledge and augmenting training data with synthetic samples to improve intent classification. In addition, I performed experiments with a conversational system to conduct interviews. I also propose methods for

¹<https://chatgpt.com/>

²<https://gemini.google.com>

learning users' preferences from their conversations, which are then used to predict their next topic of interest.

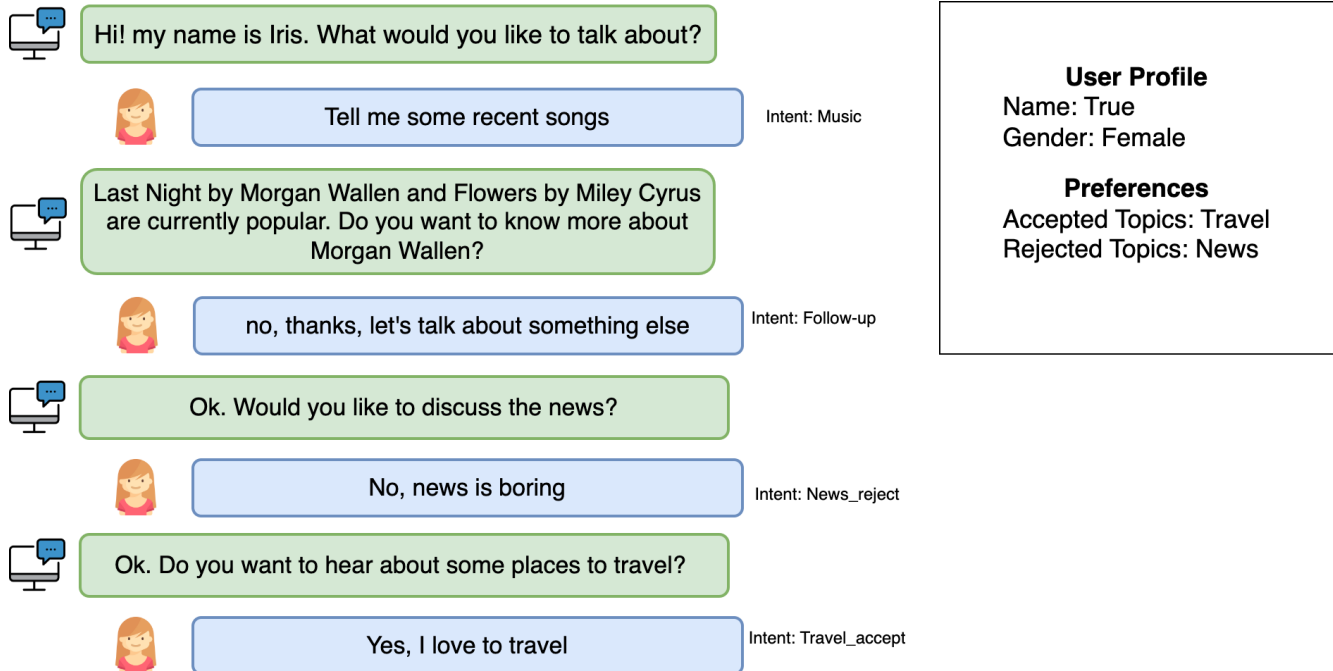


Figure 1.1: Example conversation between an open-domain conversational system and a user. The user starts the conversation with an information request. The bot recognizes the topic as “Music” and responds appropriately. When the user wants to switch topics, the system asks the user about their preference for “News”. In parallel, the system keeps track of the user’s preferred topics to use for future topic suggestions.

I address the following research questions for conversational search and recommendation:

RQ1: Can we use external knowledge to improve intent prediction? The specific research questions are:

- (a) How can we incorporate entity-type information and conversational context to perform intent classification in an open-domain conversational system?
- (b) In a conversational system that conducts interviews, how can we interpret the participant’s response, for example, by matching it to one or more predefined options? Also, can we improve intent classification by incorporating external knowledge?
- (c) How can we improve intent classification for health-related queries using entity information from an external source and synthetic training data from a large language model?

This question is studied in the context of conversational systems and web-based systems.

RQ2: Can we anticipate the user’s next topic of interest by constructing a user profile using conversation context? The specific research question is:

- (a) In an open-domain conversational system, how can we represent the user to anticipate their interest in new topics of conversation?

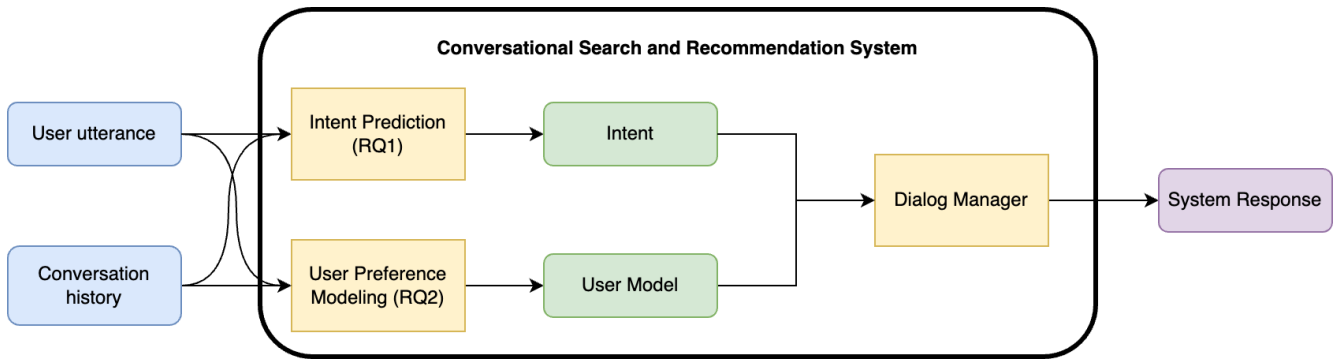


Figure 1.2: The figure shows the modules of a conversational system corresponding to the research questions.

This question is studied in the context of conversational systems.

Figure 1.1 illustrates the research questions addressed with an example. The system recognizes the user intent on each turn (intent classification). It also asks questions about the user’s preferences when appropriate, and keeps track of the user’s accepted topics and other features to perform topic recommendations (user modeling). 1.2 shows the modules of a conversational system that correspond to these problems.

In summary, the contributions of this thesis are:

1. Methods for incorporating external knowledge and conversational or session context for intent prediction (Chapter 3 and Chapter 4).
2. Methods for learning a user preference model from conversation history or session history to predict their next topic of interest (Chapter 5).

The rest of this thesis is organized as follows: Chapter 2 describes related work. Chapter 3 describes experiments I have done to address RQ1 in the conversational setting. Chapter 4 discusses my work on query intent prediction in domain-specific web search. Chapter 5 describes experiments I have done to address RQ2. Chapter 6 concludes the thesis with a discussion of contributions, potential impact, limitations, and future work.

Now, I describe the related work to place my contributions in context.

2 Related Work

This chapter describes research that is relevant to the work outlined in this dissertation. The focus is on work that also addresses my research questions as well as on approaches utilized in the implementation of these systems which may not directly relate to the research questions. To begin, a review of the literature on intent prediction is provided, as it pertains to Research Question 1 (RQ1). Subsequently, papers on conversational search are explored, as they tackle challenges similar to those encountered in both Research Question 1 (RQ1) and Research Question 2 (RQ2). Next, state of the art in conversational recommender systems is discussed, which is relevant to the topic recommendation method discussed in Chapter 5.

2.1 Intent Prediction

This section discusses existing work that deals with RQ1.

2.1.1 Intent Prediction in Open-Domain Conversational Systems

Open-domain and domain-specific conversational agents usually have a predefined set of intents and slot values that they can identify and process. Existing intent classifiers employ various approaches such as transformer-based models [100], hierarchical text classification [139], and knowledge-guided pattern matching [138] to map user utterance to the relevant intent. However, intent classification becomes challenging when the number of intents is large, and training data is scarce.

In the past, intent prediction was modeled as a text classification problem, using handcrafted features like bag-of-words, tf*idf, part-of-speech tagging, and tree kernels [99, 134]. Later, semantic and implicit information in the text was encoded using word embeddings [85] and sentence embeddings [21]. At the time of publication of ConCET in 2019, the state-of-the-art intent classifier was ADAN [38], proposed by Guo et al. Guo et al. proposed “Attentional Deep Averaged Network (ADAN)”, an unsupervised neural network method that learns relevant keywords’ saliency for the corresponding topic classes. A limitation of ADAN is that it does not perform well on utterances with recent entities. Chapter 3 describes ConCET, an intent classification approach for open-domain conversational systems that I developed in collaboration with my colleagues during the Alexa Prize 2018.

ConCET overcame this limitation of ADAN. It explicitly incorporates entity-type information and applies data augmentation to generate additional training data, leading to significant performance improvements compared to ADAN. Since then, implicit representation-based models have made substantial progress, rendering explicitly defined features less necessary. I have conducted additional experiments to assess whether external information from dynamic knowledge bases remains beneficial when employing recent semantic representation models such as BERT and T5. I use more recent language models for text representation, and knowledge adapters and retrieval augmentation as sources of external information for domain-specific web search queries in Chapter 4.

In [125], Wang et al. argue that existing methods do not work in large-scale industrial settings like the e-commerce dialog agent AliMe by Alibaba. They propose a multi-task learning user intent classifier trained using multi-task learning and designed to work for a large number of intents

from distinct domains. I face the challenge of a large number of intents in the work described in Chapter 3.2. However, I have much less training data than this approach requires. There are several approaches to deal with the data scarcity problem. In [65], Li et al. address 2 problems faced in continuous few-shot intent detection (CFID): catastrophic forgetting during continuous learning and negative knowledge transfer across tasks. They use the SimCSE [37] model, a contrastive learning approach for regularizing pre-trained embeddings to learn better sentence representations. This approach shows promising results with a small number of training examples. Another approach is data augmentation, which has been explored in [70]. The newest solution for few-shot settings is to use a pre-trained language model (PLM) [83, 84, 114]. In [84], Mi et al. address the question, how can we best prompt pre-trained language models (PLM) to perform 3 tasks related to Task-oriented Dialog (ToD): intent classification, dialog state tracking, and natural language generation. I plan to use large language models to generate more training data.

2.1.2 Intent Prediction in Information Elicitation Conversational Systems

Conversational systems for information elicitation, specifically for conducting structured interviews given a protocol, are a relatively unexplored area of research that I focus on in this dissertation. There has been extensive prior work on closely related problems like intent prediction and slot-filling for conversational systems [100, 138, 139], dialog representation [81, 93], knowledge grounded language models [150], and domain-specific language models [12]. However, these methods rely on the availability of extensive training data and the intents and slots being limited in number. In the structured interview setting, users often give long descriptive answers to open-ended questions, which makes it hard to apply these intent classification models.

A distinguishing feature of this setting is the large set of intents and the need for a model that can handle long, open-ended responses. A possible way to address the problem of a large number of intents is to use a hierarchical intent classifier [111]. In [111], Schuurmans et al. show promising results by leveraging class taxonomy. In the structured interview setting, I work with, the dialog logic is simple, and the user response can be assumed to be relevant to the last question asked. Therefore, at a given time, the number of possible intents is limited. Therefore, in my work described in Section 3.2, we model the problem as a text classification task.

Reading comprehension tasks that require answering multiple-choice questions based on some given context are also closely related to my work. [77] propose a BERT-based framework for handling multiple-choice questionnaires focused on reference passages. [43, 92] address the problems of history selection and dialog representation for conversational reading comprehension. However, answers in reading comprehension tasks are generally factual and precise, as opposed to ones in structured interviews. Therefore, the challenges involved in training models for them are different.

Language Models pre-trained on dialog [140, 153] are also relevant to this work. TOD-BERT [140], after being pre-trained on nine human-human and multi-turn task-oriented dialogue datasets, outperformed strong baselines like BERT on four downstream task-oriented dialogue applications. I use TOD-BERT in the experiments to study the advantages of dialog pre-training for this task.

External knowledge bases and knowledge graphs have been incorporated in many approaches for NLP and IR tasks to yield promising results [10, 53, 69, 74, 146]. Most of these approaches rely on the existence of a knowledge graph with relevant information. Domain-specific models like SciBERT [12] and BioBERT [60] have shown that downstream tasks can significantly benefit from models pre-trained on in-domain data.

The experiments on this topic described in Chapter 3 were conducted with a structured interview about skincare preferences. Although this data is domain-specific, there isn't a pre-trained model or knowledge graph tailored for this setting. Therefore, I perform knowledge injection using ConceptNet neighbors of terms in conversations to experiment with the effects of incorporating external knowledge.

2.1.3 Intent Prediction in Web Search

I also address intent classification in a non-conversational setting, i.e., health-related web search queries. Specifically, the task is to classify health-related queries into a predefined set of intents, for which the search system has specialized widgets for a smoother user experience. These widgets provide the interface for further query refinement. The intent prediction task, in this case, is related to conversational search since it enables interactive, multi-turn search. But it is also different in that there is even less context available than in a conversational setting. Also, the input, web search queries, is stylistically different from conversational utterances. A specific type of web search setting is addressed in my work, that of search on a health insurance website. It is a relatively small-scale domain-specific search engine.

Query intent classification in web search has been worked on in the information retrieval community for decades [19]. Two types of query intent classification are commonly studied in the literature:

1. For mapping to a vertical search engine: Here, the goal is to predict the intent behind a user's query to help a search engine automatically route the query to some corresponding vertical search engines, like in [42].
2. For identifying relevant topics: Here, the goal is to provide more relevant documents by matching the topics of the query and the documents. The topics are referred to as intents in this chapter [19].

Leveraging search results is a more widely used way of gathering additional context for query classification [19, 35]. In recent work by Srinivasan et al. [115], the authors demonstrate that providing the titles and URLs of retrieved documents as context to LLMs improves query intent classification. They further go on to introduce a knowledge distillation method to alleviate the high inference cost. However, in the setting of a small-scale domain-specific search engine with limited user interaction data, the document or URL that is appropriate for the user's intent may not be ranked highly or even present in the search results. I, therefore, propose methods for query classification that do not rely on search results. Another motivation to use entity information as a source of contextual knowledge is that the website has its specialized knowledge base of named entities like providers and facilities. Although only Wikipedia entities are used in the experiments, the proposed methods can be extended to inject knowledge about any entity, for example, names of *new* providers or drugs.

2.2 Conversational Search

Conversational search is a subfield of information retrieval that focuses on the interaction between a user and a search system in the form of a conversation. Although not directly addressed in this dissertation, conversational search is relevant to my work on intent prediction and user modeling in conversational systems. It is also somewhat related to intent prediction in the context of web-based

search for health-related queries, as discussed in Section 2.1.3. Here, I provide an overview of related problems in the context of conversational search that apply to the intent prediction task.

Query representation is an essential component of conversational search systems. In [40], the authors address the problem of learning embeddings for ambiguous search queries. In [39], the authors address the problem of representing the conversation in a conversational search scenario that includes clarifying questions. They enrich conversation representations learned by transformer networks using a novel attention mechanism from external information sources that weights each term in the conversation. This representation is then used for document retrieval and next clarifying question selection. They propose a model that can generate semantically different intents and their appropriate representations for queries if they have more than one possible intent. [161] introduces a method for conversational product search where they learn the semantic representations of user, query, item, and conversation via a unified generative framework. These representations are then used for product retrieval. The goal is to assign each query to one of a predefined set of classes. That is, intents are defined as that set of classes, as in [42], and not the space of all possible information needs represented by the query as in the above papers.

In [79], the authors introduce an approach for conversational search session representation. They propose a sparse Lexical-based Conversational REtriever (LeCoRE), which leverages these representations. This is relevant my work on conversational user preference modeling using conversation context in Chapter 5.

There are some datasets that have greatly aided research in conversational search. In [102], Qu et al. introduce the MSDialog dataset with intent annotations for information-seeking conversations. In [142], Xiao et al. introduce a conversational dataset for online shopping. The dataset was constructed without crowdsourcing. They also introduce an end-to-end conversational search method. In [28], Chu et al. introduce a Chinese conversational search behavior dataset that was collected using a WOZ approach. I use the TripClick dataset [104] as a source of web search session data for my experiments on intent prediction health-related queries to evaluate the method on publicly available data.

The use of LLMs is becoming increasingly common in conversational search. In [88], Mo et al. propose ConvGQR, a new framework based on generative pre-trained language models (PLMs) for query reformulation in conversational search. In [78], Mao et al. use GPT-3 to rewrite queries for conversational search. In this work, I use LLMs to generate synthetic queries for data augmentation in Chapter 4.

2.2.1 Entity linking

Entity linkers identify entity mentions in text and resolve and map the mentions to a unique identifier in an associated knowledge base, a common choice for which is Wikipedia. Babelfy [8] uses a graph-based approach for jointly performing word sense disambiguation and entity linking. DBpedia Spotlight [82] links entity mentions to their DBpedia [11] URIs using cosine similarity between Vector Space Model representations of the DBpedia resource and the context to disambiguate mentions. The SMAPH [30] system for linking web-search queries piggybacks on a web search engine to put the query into a larger context, and further uses a supervised ranking model to predict the joint annotation of the full query.

In Section 3.1, to annotate the dataset with entity information, I used DBpedia Spotlight as well as a customized PMI-based entity linker that I built, that is described in section 3.1. Entity linking requires an associated knowledge base with all the entities that it links entity mentions to. DBpedia Spotlight links entity mentions to their DBpedia URI. It uses Lingpipe’s string-matching

algorithm implementation together with a weighted cosine similarity measure to recognize and disambiguate mentions. Other entity linkers include AIDA, which links to the YAGO2 database, and TAGME, which is suitable for short, incomplete texts and links entities to Wikipedia pages. The experiment results in Section 3.1 show that DBpedia Spotlight is an effective entity linker for the datasets used. However, the customized entity linker I built specifically for our application leads to even better results on incorporating entity information into the text representation.

For the experiments in Chapter 4, I use the ReFinED [119] entity linker that uses Wikidata as the knowledge base. I did not explore the benefits of using an application-specific entity linker in this setting. However, the results show that incorporating entity information into the text representation improves the performance of the intent classifier. It is reasonable to assume that using a more accurate entity linker would further improve the performance of the intent classifier.

2.3 Entity-based Text Representation

Entity-based text representation has been studied for different research areas such as information retrieval [143], question answering [144], and coherence modeling [47, 90]. Yamada et al. [144] proposed a model to encode entity information from a corpus like Wikipedia into a continuous vector space. This model jointly learns word and entity representations from Wikipedia and DBpedia. [128] proposed a CNN-based model for merging the text and entities extracted from a large taxonomy knowledge base for short-text classification. I propose incorporating entities, their most likely types, and positional order into a neural network architecture and processing pipeline for conversational intent classification in Chapter 3.

In Chapter 4, I use TransE [16] embeddings to represent entities in text. This method was originally proposed in [154] to add entity information to a pre-trained language model, specifically BERT. In my work, I study its effectiveness in pretrained language models of different sizes and compare it to other methods of incorporating entity information.

2.4 Conversational Systems for Interviewing Users

Conversational systems make for an intuitive and user-friendly interface. This makes them the perfect tool for collecting data from users in a natural way. They can also serve as assistants or coaches, e.g. while users are performing a task or working on behavior change. In [62], conduct a user study with a conversational system that encourages self-disclosure in mental health patients. Their results confirm the effectiveness of conversational systems in eliciting information from users. In [55], the authors introduce Reflection Companion, a mobile conversational system that supports engaging reflection on personal sensed data, specifically physical activity data collected with fitness trackers. Their results based on a user study with Fitbit users show that the system helped facilitate behavior change in the users. Seltzer et al. in [113] use a virtual agent to ask probing questions while a participant fills out a market research survey form. I also address the same problem of eliciting more informative responses from users in a conversational setting. In [158], Zhou et al. propose a conversational system for health coaching. They focus specifically on low-resource settings. They release their dataset of conversations between coaches and users. This is one of the few real-world human-human conversational datasets available. In [159], the authors gather data using a WOZ setting to train a dialog system for motivational coaching. This system does not use any hand-coded rules. They learn “dialog phase and scenario embeddings” to inform their

response generator of the context and type of dialog.

Preference elicitation for CRS is an active area of research. Deciding what questions to ask and in what order is a critical problem to solve, and is addressed in [27, 152, 160]. The systems in [27, 63, 160] ask yes/no questions about the user’s preference for an item or an attribute that would be the most helpful in converging to a recommendation. In [152], the system asks the user for their preference in specific attributes and can interpret responses with values of those attributes. Zhao et al. in [155] and Li et al. [68] address the preference elicitation problem in CRS using a multi-arm bandit approach.

The above work proves that conversationally eliciting free-form information from users is a relevant problem. I study specifically the intent classification for such conversations as part of RQ1 in Section 3.2. I built an information elicitation conversational system with a simple dialog policy for the purpose of my research.

2.5 Knowledge Injection

Injecting new knowledge into LLMs has been an active area of research in recent years. [98] proposes E-bert, where Wikipedia2Vec entity vectors are projected into BERT’s wordpiece vector space. This way, entity information from external vectors can be leveraged without additional pre-training of the BERT encoder. The PELT [147] approach also involves building an entity lookup table in the representation space of PLM token embeddings. However, here, the external representations are obtained from textual corpora instead of a knowledge graph. The output representations of a specific entity in multiple contexts are aggregated to build a contextualized, “knowledge-aware” entity representation. Another popular method is K-adapters [129]. Here, the authors introduce a transformer-based adapter layer. Two kinds of adapters are added to RoBERTa, one for factual knowledge from Wikipedia, and another for linguistic knowledge from dependency parsing. In [154], the authors propose a method of incorporating entity information into a model already fine-tuned for a classification task. Their method, called “map tuning”, projects TransE embeddings [16] of entities into the token embedding space of a language model. I apply this method for knowledge injection for query intent classification.

A recently active area of research is knowledge injection into large language models. LLMs have a large amount of intrinsic world knowledge. However, in the cases where the knowledge is not present in the pre-training data, it is necessary to inject external knowledge. And this is a challenging problem because the knowledge must be injected in a way that does not disrupt the existing knowledge in the model, and is also not prohibitively expensive in terms of computation. In [80], the authors introduce a new data generation process for incorporating previously unseen knowledge into LLMs. [94] compares the effectiveness of unsupervised fine-tuning with RAG in using external datasets to incorporate new information. This dissertation only explores knowledge injection into language models of size upto 3 billion parameters. I instead investigate whether LLMs can be used to generate synthetic queries for data augmentation in Chapter 4 as a way to incorporate external knowledge into smaller intent classification models.

Retrieval augmentation is another way of incorporating external knowledge that I use in Chapter 4. Lewis et al. proposed retrieval augmented generation (RAG) in [64], arguing that traditional models store factual knowledge in their parameters but struggle with precise knowledge access and manipulation, leading to suboptimal performance on tasks requiring dynamic knowledge integration. This is consistent with the experimental results in Chapter 4 that show that retrieval augmentation improves the performance of language models on intent classification.

The idea of conditioning a language model’s generated text on retrieved passages has been extensively explored since then. Leveraging retrieved context to improve the knowledge-groundedness of models has been studied during various stages of model training and inference. It’s effectiveness has been demonstrated during pretraining [17], fine-tuning [71, 110], and inference [46, 116, 131].

I apply retrieval augmentation to improve the performance of intent classification models in Chapter 4. In the specific setting studied, dense retrieval of relevant paragraphs is not feasible. I instead apply a simple yet effective method of using explicit entity linking to get entity information from a knowledge base to augment the text representation of the query.

2.6 Dialog Management

Although this work does not directly address dialog management, it is an important component of the conversational systems used in the experiments. [122] introduces a new dialog task, *Conversational Tree Search*. Here, all the possible dialog flows are predefined in the form of a tree. They also introduce a novel deep reinforcement learning architecture to carry out these interactions with a user. They find that their framework leads to a higher goal completion rate. [86] presents some interesting findings for conversational agents used to enable micro-coaching dialogs in nutrition — brief coaching conversations related to specific meals, to support the achievement of nutrition goals. The authors compared reinforcement learning (RL), rule-based, and scripted approaches for dialog management and found that while the data-driven RL chatbot succeeded in shorter, more efficient dialogs, surprisingly, the simplest, scripted chatbot was rated as higher quality, despite not fulfilling its task as consistently. The authors hypothesize that this could be because the scripted chatbot’s sequence of questions might have sounded more intuitive to the user. The paper highlights unintuitive/unexpected dialog flow as a potential drawback of RL-based strategies. This also explains why real-world systems often still rely on carefully designed conversational interactions instead of data-driven RL and generative approaches. Several tools have been proposed to facilitate dialog design. For example, [26], the authors introduce ProtoChat, a crowd-powered chatbot design tool built to support the iterative process of conversation design. This work supports the continued relevance of conversational systems that use scripted dialog flows and cannot be implemented end-to-end using generative models.

2.7 Clarification Question Generation

This thesis focuses on implicit intent classification in conversational systems. However, a complementary and relevant task is clarification question generation for intent clarification when it is hard to build any practical conversational system that can always understand the user’s needs without asking clarifying questions. Therefore, this problem has been extensively studied, especially in the context of conversational search. In [162], the authors investigate the effect of the quality of clarification questions in web search on user behavior and satisfaction. They find that low-quality clarification questions harm search performance and user satisfaction. However, asking clarification questions to understand user intent is a critical capability for any conversational system.

Two of the most commonly used datasets for this task are ClariQ [89] and MIMICS [149]. The ClariQ dataset was released by Aliannejadi et al. in [89] to enable answering the following research questions: 1) When to ask clarifying questions during dialogues? and 2) How to generate the

clarifying questions? Zamani et al. in [149] introduce the MIMICS, which is a collection of search clarification datasets for real web search queries sampled from Bing query logs. Each clarification question comprises a question and up to five candidate answers.

In [157], the authors propose a method for generating clarification questions based on top search results for the query. Another way of generating clarification questions is to use templates and fill them with the most likely query facets. In [127], the authors propose a method for generating clarification questions using templates. They train a model whose objective jointly optimizes selecting question templates and filling question slots. They use the MIMICS [149] dataset in their experiments. In [133], the authors address the problem of zero-shot clarification question generation using question templates and query facets. Facet ranking or generation is an important component of such systems. In [108], the authors analyze and combine 3 different methods of query facet generation: facet extraction as sequence labeling, facet generation as autoregressive text generation, and extreme multi-label classification. They find that these methods produce complementary facets and propose ways to aggregate them.

Research has also been carried out to identify the most useful types of clarification questions. In [118], Tavakoli et al. analyze human-generated clarification questions on Stack Exchange. They identify six types of clarification questions: 1. Ambiguity/Incompleteness (e.g., "How much money did you assume to start with?"); 2. Confirmation (e.g., "Does it have to be a single word?"); 3. General (e.g. "Would you like to make the question more specific?"; 4. Incorrectness (e.g., "Are you sure it is 62 and not 66?"); 5. Paraphrasing (e.g., "Are you asking how to write an exchange simulator?"); and 6. Suggestion (e.g., "Can the book "Monte Carlo simulation in financial engineering" by Glasserman help you?"). In their survey paper [49], Keyvan et al. list the types of ambiguous queries that have been researched in work on conversational ambiguous queries.

Salle et al. in [107] investigate the effectiveness of conversational search refinement and clarification using a user simulator. They also analyze the types of queries for which facet ranking algorithms perform the best. They find that when the list of facets is large, the success rate of converging to the facet relevant to the user is generally low. This is especially the case for ambiguous queries. This is consistent with the challenges I encounter with response interpretation for questions with a large number of possible answers in Section 3.2. Thus, clarification for queries with a large number of facets is still an open problem. This work on intent prediction in this setting is applicable because it helps narrow down the number of possible intents for a query.

The tradeoff between search result quality and search efficiency is an important consideration in real-world conversational search systems. In [135], the authors propose a reward-free conversation policy imitation learning framework which also optimizes efficiency.

In [52], Amazon Alexa AI explore the problem of preventing too many confirming questions in the context of ASR/spoken language understanding confirmation. In [6], the authors analyze the best time for asking clarification questions and asking for feedback in mixed-initiative conversational search systems. In [132], the authors propose a reinforcement learning agent for conversational search that explicitly takes the risk of asking irrelevant clarification questions into account. This research highlights the importance of accurate intent prediction in conversational systems to avoid asking clarification questions unless necessary.

In [112], the authors propose a transformer-based model to predict the user engagement resulting from asking a clarification question in web search. In [95], the authors develop a user simulator to generate conversational search data with feedback from the user. The simulator, once initialized with an information need description, is capable of providing feedback to the system's responses, as well as answering potential clarifying questions. They extend proven models and show that effective utilization of user feedback can lead to improved retrieval performance. In [7], the authors

collect and release a new clarifying question dataset focused on open-domain single-and multi-turn conversations. They also propose a pipeline consisting of offline and online steps for evaluating the quality of clarifying questions in various dialogues.

2.8 Conversational Recommendation

Before generative models became capable enough to carry out coherent conversations, conversational systems would use a repository of predefined utterances and rank them based on the user's input [145]. However, this approach was mainly applicable to a non-modular conversational system. In practice, due to the vast number of possible utterances coming into a social bot, many conversational systems rely on multiple response modules where each response module would be responsible for a particular domain or set of domains [51]. When the user is passive or gets fatigued with a particular topic, the system needs to switch to a different component with domain-specific capabilities to keep the user engaged. This thesis formalizes the problem of suggesting the best next interesting topic. This is closely related to the problem of conversational recommendation [27, 67, 117].

2.9 Summary

In this chapter, I have reviewed the related work in the areas of intent prediction, conversational search, entity-based text representation, conversational systems for interviewing users, knowledge injection, dialog management, and clarification question generation.

My thesis contributes entity-information-based methods for intent prediction in multiple settings. It also explores data augmentation using entity information.

It also contributes to work on conversational systems for interviewing users by studying the specific challenges in intent prediction in this setting.

It formalizes and proposes a solution to the problem of suggesting the best next interesting topic in an open-domain conversational system. However, its contribution to the area of conversational recommendation is limited because it applies to a very specific setting. Also, newer work has made it less relevant.

3 Intent Detection in Conversational Search and Recommendation

This chapter addresses RQ1, i.e., can we improve intent detection with external knowledge?

I want to clarify that in this dissertation, the term “Intent” represents different things depending on the type of conversational system. In modular open-domain conversational systems, intent classification is the critical step that routes the user utterance to the appropriate module for queries in that domain. Here, intent informs us of the topic the user is interested in. This is how the term is used in prior work like [42], and my work described in Section 3.1. On the other hand, in information elicitation conversations, intent represents a characteristic of the user. In particular when there is a structured interview with a list of questions and possible answer options given, intent refers to the answer options. That is the definition used in Section 3.2. In conversational search and recommender systems, intent represents the user’s information and/or product need, and intent prediction is often modeled as a clustering task. However, in this dissertation, in all three settings where intent prediction is studied, there is a predefined set of intents that I want to classify the given input into.

In Section 3.1, I address intent prediction for an open-domain conversational search and recommendation social bot, Irisbot [3]. The work described was originally published in the proceedings of CIKM 2019 [5]. My key contributions to this paper were:

1. Coming up with the idea of incorporating entity-type information for intent prediction.
2. Implementation of entity linking and representation modules.
3. Independent evaluation of entity linkers
4. Analyzing the effect of entity information on intent prediction.

I describe the experiments I conducted in collaboration with my team as part of the Alexa Prize Challenge in 2018. Our entry to the contest, IrisBot [3], is a conversational agent that aims to help customers be informed about the world around them while being entertained and engaged. It incorporates real-time search, informed advice, and the latest news recommendation into a coherent conversation.

Following that, Section 3.2 describes experiments for conversational response interpretation, which I model as an intent prediction problem. This research was published in WebConf 2023’s Companion Proceedings [106].

3.1 ConCET: Entity-Aware Intent Classification for Open-Domain Conversational Agents

Identifying the intent (in this section’s context, that refers to the domain that the user is interested in talking about) of each user’s utterance in open-domain conversational systems is a crucial step for all subsequent language understanding and response tasks. In particular, for complex domains, an utterance is often routed to a single component responsible for that domain. Thus, correctly mapping a user utterance to the right domain is critical. To address this problem, ConCET: a

Concurrent Entity-aware conversational intent classifier is introduced, which incorporates entity-type information together with the utterance content features. Specifically, ConCET utilizes entity information to enrich the utterance representation, combining character, word, and entity-type embeddings into a single representation. In addition, to complement the model, a simple and effective method for generating synthetic training data is proposed, to augment the typically limited amounts of labeled training data, using commonly available knowledge bases as to generate additional labeled utterances. ConCET and the proposed training method are extensively evaluated first on an openly available human-human conversational dataset called Self-Dialogue, to calibrate the approach against previous state-of-the-art methods; second, ConCET is evaluated on a large dataset of human-machine conversations with real users, collected as part of the Amazon Alexa Prize. The results show that ConCET significantly improves intent classification performance on both datasets, reaching 8-10% improvements over deep learning methods that do not use entity type information.

Since the publication of this work, many new intent classification methods have been built on top of larger language models such as BERT. Although these models encode entity information implicitly, explicitly encoding entity information can improve the performance of these models, especially for unseen entities [64]. The effectiveness of incorporating entity information for improving intent classification is investigated in Chapter 4.

3.1.1 Introduction

Open-domain conversational agents are often built using a modular architecture, where each module is responsible for a specific domain. For example, the *Movie* Bot is responsible for answering questions about movies, and the *Music* Bot is responsible for answering questions about music. Although end-to-end generative models have been built, production environments still use a combination of specialized modules, some of which are created and maintained by third-party developers. The first step in this architecture is to identify the intent of the user’s utterance, and route it to the appropriate module.

Classification of utterances in open-domain dialogue systems is a much more challenging task compared to general text classification due to four main factors: 1) Human utterances are often short; 2) Errors in Automatic Speech Recognition (ASR); 3) Users frequently mention out-of-vocabulary words and entities; 4) Lack of available labeled open-domain human-machine conversation data.

This makes keyword-based classification insufficient. Domain-specific keywords or triggers might help for queries like “Let’s talk about my dog”, since the word “dog” appears frequently in utterances from the *Pets_Animals* intent. However, they do not enable us to correctly classify utterances containing ambiguous keywords that can refer to multiple entities. For example, to correctly classify utterances like “When is the next Hawks game?”, we need to take into account all the possible types of entities that the word “Hawks” might be referring to, i.e. the bird hawk and the sports team Atlanta Hawks, as well as the context, which mentions “game”.

Moreover, the creation of new entities, like recent movies, makes the model obsolete with time. To fix this problem, it would be necessary to constantly keep updating the model by incorporating new information about people, organizations, movies and other entities, which can cause unintended effects in the model, and would be inefficient. To address these problems, a novel, data-driven approach to entity-aware conversational intent classification is introduced: a deep learning algorithm named Concurrent Entity-aware Intent classifier (ConCET) augmented with external knowledge about entities and their types, retrieved dynamically from a knowledge base, using either a publicly available entity linker, or one fine-tuned for the expected utterances. ConCET combines the implicit

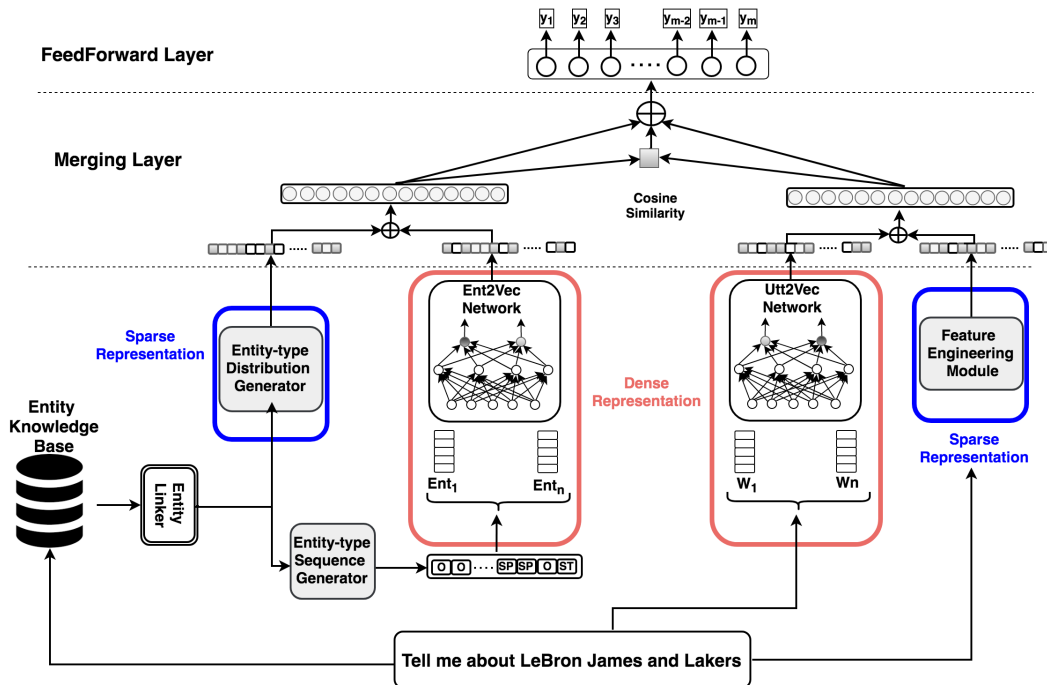


Figure 3.1: The overall network architecture for Entity-Aware Intent Classifier (ConCET) model, where “SP” and “ST” stand for *Sports_Player* and *Sports_Team* entity-types.

and explicit representations of the utterance text, together with the semantic information retrieved about the mentioned entities. To train ConCET, a synthetic dataset, created from the expected entities and entity-types, to augment the limited labeled conversational data is introduced. This dataset is modeled to approximate the real human-machine conversations observed with real users, as described below. ConCET is evaluated on an openly available human-human conversational dataset, and a large dataset of human-machine conversations with real users, collected as part of the Amazon Alexa Prize 2018. The results show that ConCET significantly improves intent classification performance on both datasets, reaching 8-10% improvements compared to state-of-the-art deep learning methods.

In summary, the contributions of this work are: (1) The development of ConCET, a novel entity-aware intent classifier by combining implicit and explicit representations of an utterance and fusing them with handcrafted features; (2) Incorporating external knowledge about entities retrieved from a knowledge base; and (3) creation of a new large-scale synthetic yet realistic dataset for training intent classification systems designed for open-domain conversational agents.

The following subsections detail the proposed approach and the experimental results.

3.1.2 ConCET System Overview

ConCET system is now introduced at a high level, before diving into implementation details. The proposed ConCET model is illustrated in Figure 3.1.

ConCET utilizes both textual and entity information from an utterance. To represent textual and entity information, ConCET extracts both sparse and dense representations. To this end, a pipeline of deep neural networks and handcrafted feature extraction modules is designed. This pipeline consists of four components namely Utterance-to-Vector (Utt2Vec) network, feature engineering module, Entity-to-Vector (Ent2Vec) network, and the Entity-type distribution generator. The

Entity-type distribution generator module uses an entity linker to get the entity-type distribution corresponding to each entity in the utterance.

Utt2Vec and the feature engineering module extract the textual representation. Utt2Vec is a deep neural network model which utilizes character, word, and POS tags for utterance representation. Feature engineering module extracts handcrafted features such as LDA and LSA topic distribution from an utterance. Finally, they are combined through a fully-connected neural network.

To model the entity information, ConCET utilizes both the entity-type distribution and the order of entity-types appearing in the utterances. Ent2Vec network is responsible for mapping this entity sequence representation to a high dimensional vector. Entity-type distribution features and the output of the Ent2Vec network are combined through a fully-connected neural network.

Next, the cosine similarity¹ between textual and entity representations is computed. This similarity value, concatenated with the textual and entity representations, is fed to a feed-forward layer to compute the final softmax distribution of intents.

To summarize, ConCET proposes an entity-aware text representation model that learns a ternary representation of character, word and entity information. In the next section, the entity linking methods used to derive entity-based information are introduced. The experiments were conducted using two different entity linkers to measure the sensitivity of the ConCET model to the entity linking step. Then, in Section 3.1.4 I explain the details of the ConCET model.

3.1.3 Conversational Entity Linking

In this section, I describe the two entity linkers that were used for detecting entities and their type distributions. The type information is used for semantic representation in the ConCET model.

I emphasize that the focus of this work is *not* on developing a novel entity linker, which is an important area of research on its own. Rather, I experiment with an off-the-shelf entity linker, DBpedia Spotlight², and my own PMI-based domain-specific entity linker (PMI-EL), designed to cover in more depth some of the conversation domains and entity-types most relevant to the conversational agent. The experiments with different off-the-shelf entity linkers during the development of our conversational agent showed these two linkers are the most effective for intents that our bot supported. I describe both entity linkers in depth in the next section. Here I want to emphasize that the proposed classifier model can incorporate the output of any available entity tagger or linker.

DBpedia Spotlight

DBpedia Spotlight annotates DBpedia resources mentioned in the text as described in reference [82]. It annotates DBpedia resources of any of the 272 classes (more than 30 top-level ones) in the DBpedia Ontology. It performs entity annotations in 3 steps, 1) spotting, 2) candidate selection, and 3) disambiguation. It uses the Aho-Corasick string matching algorithm for finding all the phrases which could potentially be entity mentions or surface forms. It then finds candidate entities for each surface form using the DBpedia Lexical Dataset. For disambiguation, each candidate DBpedia resource is first represented in a Vector Space Model (VSM) as the aggregation of all paragraphs mentioning that concept in Wikipedia. The candidates are then ranked by their $tf * icf$ cosine similarity score with respect to the context, where the icf score estimates how discriminating

¹Dot product also can be used. In this case, the entity vector should be normalized to unit length.

²<https://github.com/dbpedia-spotlight/spotlight-docker>

a word is, which is assumed to be inversely proportional to the number of DBpedia resources it is associated with.

PMI-based Entity Linker (PMI-EL)

A domain-specific entity linker called PMI-EL is created for our conversational system for the Alexa Prize, which annotates the 20 entity-types most relevant to our system. It links entities to an associated knowledge base containing all the entities supported by our conversational agent. PMI-EL follows similar steps to DBpedia Spotlight. However, it does not use the utterance context in the disambiguation step and relies solely on an estimated prior distribution of types for a given entity for disambiguation. The main reason was that most of the user utterances were short (average utterance length of 3.07 words), and sometimes consisted of just the entity name. Thus, the context was often not helpful or present, and type inference based on prior probabilities may be sufficient for this setting. I next describe the process by which the knowledge base was constructed, and how the prior type probabilities were estimated for entity-type inference.

Movie_Name	Celebrities	Authors	Bands
Sports_Team	Sportname	Companies	Food
Organization	Politicians	Universities	Singers
Songname	Animal	Country	Actors
Hotels_Foodchains	Tourist_points	Genre_Books	City

Table 3.1: Entity-types recognized by PMI-EL.

PMI-EL knowledge base construction

The knowledge base starts with entities from a snapshot of DBpedia from 2016. Additionally, to provide coverage of current entities of potential interest to the user, the knowledge base is augmented by adding entities that our open-domain conversational agent supports. Entities from the following sources and domains are periodically retrieved:

- Persons, Organizations and Locations: from news provided by Washington Post³
- Cities and Tourist Attractions: from Google Places API⁴
- Bands and Artists: from Spotify⁵ and Billboard⁶
- Books and Authors: from Goodreads⁷ and Google Books⁸
- Actors and Movies: from IMDb⁹ and Rotten Tomatoes¹⁰

An index of all the entities and their corresponding types is maintained using ElasticSearch¹¹, which is used in the online entity linking step.

³<https://www.washingtonpost.com/>

⁴<https://developers.google.com/places/web-service/search>

⁵<https://www.spotify.com/us/>

⁶<https://www.billboard.com/>

⁷<https://www.goodreads.com/>

⁸<https://developers.google.com/books/>

⁹<https://www.imdb.com/>

¹⁰<https://www.rottentomatoes.com/>

¹¹<https://www.elastic.co/products/elasticsearch>

PMI-based type distribution

For entities with more than one type, the estimated pointwise mutual information (PMI) [18] of the entity with all its types is also indexed. PMI is a measure of how much the actual probability of a particular co-occurrence of events $p(x, y)$ differs from what we would expect it to be on the basis of the probabilities of the individual events and the assumption of independence of events x and y , and is calculated as:

$$PMI(x, y) = \ln \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (3.1)$$

To predict the most likely type for entities with multiple types, the point-wise mutual information (PMI) of the entity with each type is estimated by counting the co-occurrences of the entity and the type’s name in a large corpus, which has been shown to correlate with the probability of association [18]. More formally, the entity-type PMI score is computed as:

$$PMI(m, t_i) = \frac{|(Docs(m, C) \cap Docs(t_i, C))|}{|Docs(m, C)|} \quad (3.2)$$

where m is an entity mention, t_i is a type, C is a corpus and $Docs(phrase, C)$ is a set of documents in C containing a given phrase. For the experiments, a publicly available corpus of 46 million social media posts from a snapshot of Reddit is used as C .¹²

For example, to disambiguate the mention “Kings” in a user’s utterance, the number of times each type name co-occurred with the word “Kings” in the corpus is computed, normalized by the total number of occurrences of the word “Kings” itself. The relative values of PMI are used to come up with a ‘type distribution’ for the entity, which indicates the probability of the entity mention being linked to an entity of a certain type. In this example, the type distribution for a string “Kings” is: [*Sports_Team* : 0.54, *Movie_Name* : 0.44, *City* : 0.02]. Because of the large size and diversity of the corpus, PMI is expected to be a good estimate of type distribution. Despite potential noise in estimating type distribution for some polysemous entities, the ConCET model is able to use the type distribution, as is demonstrated empirically under a variety of conditions.

PMI-EL entity detection in utterances

To support efficient entity linking at run-time, an inverted n-gram entity index was constructed for all entities in the knowledge base. At runtime, entities are detected via n-gram matching against an entity index. For example, if the utterance is “who won the Hawks and Kings game”, the index is queried for “the Hawks”, “Kings”, “Hawks” and every other possible n-gram with less than 6 words. For this utterance, the response from the entity index would be the entities and the type distributions associated with them, e.g. “Hawks”: [*Sports_Team* : 0.88, *Animal* : 0.11, *City* : 0.01] and “Kings”: [*Sports_Team* : 0.54, *Movie_Name* : 0.44, *City* : 0.02].

The entity detection step has time complexity $O(n^2)$ in the number of words in the utterance since $O(1)$ look-ups are performed for $O(n^2)$ n-grams for each utterance. The running time for entity linking is 16 ms on an average for utterances with 4 words which were common, and 100 ms for utterances with 32 words, which were among the longest utterances encountered. However, PMI-EL would not be efficient if used on very long text.

¹²<https://files.pushshift.io/reddit/submissions/>

The output from the entity linker is passed to the Entity Representation Model, described in 3.1.4, which converts it into a suitable representation for the ConCET model.

3.1.4 ConCET: Concurrent Entity-Aware Intent Classifier

In this section, I present the details of ConCET model. First, Section 3.1.4, describes the model for the textual representation of the utterance. Then, Section 3.1.4, presents the proposed entity representation model. Finally, Section 3.1.4 discusses the merging and decision layer of the ConCET model.

Textual Representation

Character, word, and POS tagging are used to model the textual representation. Then, the representation is enriched with the unsupervised topic distribution, as described in detail next.

Utterance to vector (Utt2Vec) network

Utt2Vec network takes word tokens Utt_w , characters Utt_c and POS tags Utt_p of an utterance Utt as inputs:

$$Utt_w = [w_1; w_2; w_3 \dots w_n] \quad (3.3)$$

$$Utt_c = [[c_{11} \dots c_{1k}]; [c_{21} \dots c_{2k}]; \dots [c_{n1} \dots c_{nk}]] \quad (3.4)$$

$$Utt_p = [p_1; p_2; p_3 \dots p_n] \quad (3.5)$$

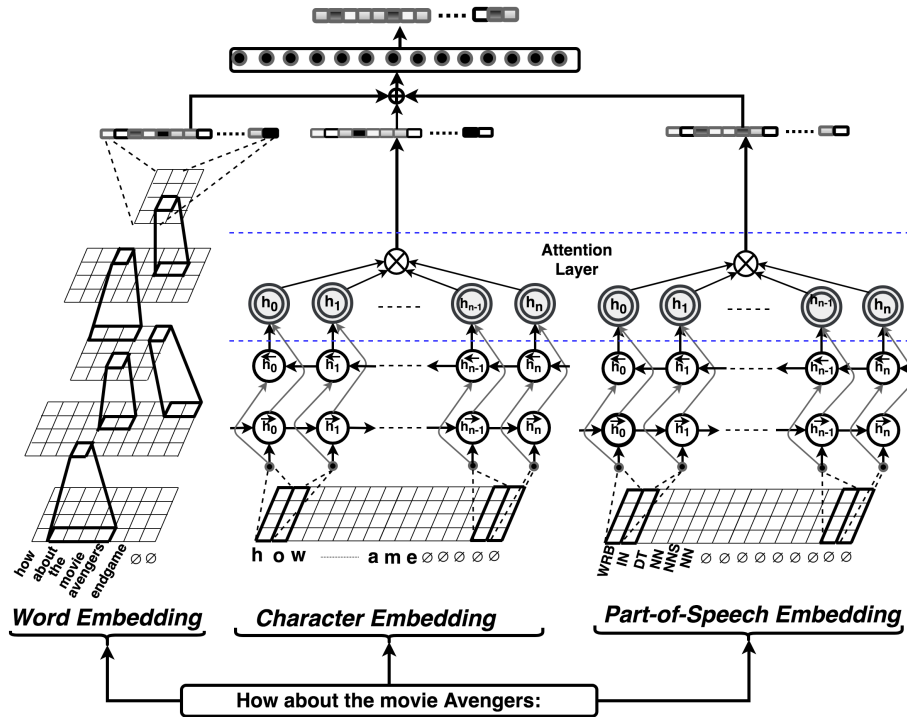


Figure 3.2: Utt2Vec network.

The NLTK¹³ library is used for extracting POS tags. Utt2Vec network allows freedom of combining different deep learning architectures such as CNN and RNN to extract features. I define three functions f_w , f_c , and f_p that each take these inputs and output learned hidden representations (h):

$$h_w = f_w(Utt_w) \quad (3.6)$$

$$h_c = f_c(Utt_c) \quad (3.7)$$

$$h_p = f_p(Utt_p) \quad (3.8)$$

For the implementation, f_w is a 3-layered CNN with max pooling. For f_c and f_p , 1-layered BiLSTM network with global attention is used. For the word embedding layer, the weights are pre-initialized using Word2Vec vectors with size 300. The weights on the word embedding layer are tuned during training. For character and POS embeddings, the embedding layer with size 16 is randomly initialized. Given the hidden representations of each timestamp h_i in LSTM cells, dot product similarity score s_i is computed based on a shared trainable matrix M , context vector c and a bias term b_i . Softmax activation is applied on similarity scores to obtain attention weights α . Lastly, using learned α , weighted sum on BiLSTM hidden representations is applied to obtain the output \hat{h} as follows:

$$s_i = \tanh(M^T h_i + b_i) \quad (3.9)$$

$$\alpha_i = \frac{\exp(s_i^T c)}{\sum_{i=1}^n \exp(s_i^T c)} \quad (3.10)$$

$$\hat{h} = \sum_{i=1}^n \alpha_i h_i \quad (3.11)$$

M , c , and b are randomly initialized and jointly learned during training. The three outputs from word-CNN (h_w), char-BiLSTM (\hat{h}_c), and POS-BiLSTM (\hat{h}_p) are concatenated to produce Utt2Vec output:

$$Utt2Vec_{out} = [h_w; \hat{h}_c; \hat{h}_p] \quad (3.12)$$

This final output is fed to a linear layer of size 256 with ReLU activation and a dropout rate of 0.5 to obtain the utterance vector.

Feature engineering module

The goal of this module is to provide the flexibility of incorporating various external features in ConCET. Since the intents correspond to different domains, unsupervised topic modeling features are extracted. However, depending on the data and the task, any type of feature extraction pipeline can be incorporated here. Two different topic modeling algorithms are combined, LDA and LSA, and implemented models using the Gensim library¹⁴. Given hyperparameter n , these models output the unsupervised topic distribution of size n . By concatenating the two outputs described in the table below, a topic distribution vector of size $2n$ is obtained. Default parameter values in Gensim were used for training the LDA, LSA, and cosine similarity models.

¹³<http://www.nltk.org>

¹⁴<https://radimrehurek.com/gensim/>

Features	Short Description
F_{LDA}	LDA topic distribution
F_{LSA}	LSA topic distribution

The outputs of these two vectors are concatenated to produce F_{out} :

$$F_{out} = [F_{LDA}; F_{LSA}] \quad (3.13)$$

Entity Representation

I now describe how the entity information from the entity linker is encoded as input to the model. There are two modules to do this encoding

- Entity-type sequence generator
- Entity-type distribution generator.

Entity-type sequence generator converts the input word sequence to an entity-type sequence so that the model can learn to predict the intent based on the order in which different entity-types appeared in the utterance. This sequence is fed into the Ent2Vec network, which creates a high-dimensional vector representation for the sequence. The Entity-type distribution generator constructs an overall entity-type distribution for the utterance by aggregating type distributions for all the entities. Finally, the output of Ent2Vec is concatenated with the entity-type distribution to generate the final entity representation. I now describe these modules in detail.

Entity-type sequence generator

The input of this module is the list of entities and their type distributions derived from the entity linker. To generate this entity sequence, the best type corresponding to each entity needs to be assigned. The words that are not a part of an entity are assigned *Other* or *O*. For example, for “who won the Hawks and Kings game”, a possible entity sequence vector would be [“who”/*O*, “won”/*O*, “the”/*ST*, “Hawks”/*ST*, “and”/*O*, “Kings”/*ST*, “game”/*O*]. However, different entity linkers can differently assign entity-types to each word. Consequently, the resulting entity vector has the exact length of the utterance.

$$Utt_{ent} = [e_1; e_2; e_3 \dots e_n] \quad (3.14)$$

Entity-type distribution generator

For this module, I first have the total number of entity-types that the model will support. For example, for the PMI-based linker, 20 types are supported, and for DBpedia Spotlight, the 1000 most frequent entity-types from the training set are supported. After determining the size, the distribution value for each entity-type is either 0, or the maximum value for that type in the list of entity-type distributions. For the example from the previous section, “who won the Hawks and Kings game”, the type distributions for the two entities from the PMI-based linker are, respectively, [*Sports_Team* : 0.88, *Animal* : 0.11, *City* : 0.01] for “Hawks”, and [*Sports_Team* : 0.54, *Movie_Name* : 0.44, *City* : 0.02] for “Kings”. In that case, if the entity linker identifies 20 types in total, the final entity-type distribution is [*Sports_team* : 0.88, *Movie_Name* : 0.44, *Animal* : 0.11, *City* : 0.02]. The value corresponding to the remaining types in Table 3.1 is 0.0 in the final output vector of length 20.

Entity to vector (Ent2Vec) network

The input to Ent2Vec network is a list of resolved entity-types per word for Utt_{ent} from entity-type sequence generator:

$$Utt_w = [w_1; w_2; w_3 \dots w_n] \quad (3.15)$$

$$Utt_{ent} = [e_1; e_2; e_3 \dots e_n] \quad (3.16)$$

A function f_e that takes Utt_{ent} and outputs learned hidden representations is defined as follows:

$$h_e = f_e(Utt_{ent}) \quad (3.17)$$

A 1-layered BiLSTM network is used as f_e function. An entity embedding layer that has 16 trainable weights is randomly initialized for each entity-type. Then, the same attention mechanism as in Section 3.1.4 is applied to h_e to obtain \hat{h}_e or Ent2Vec_{out}. Lastly, entity-type distribution Ent_{dist} is concatenated with \hat{h}_e to obtain the final Entity output:

$$Ent_{out} = [Ent2Vec_{out}; Ent_{dist}] \quad (3.18)$$

This output is fed to a linear layer of size 100 with ReLU activation and a dropout rate of 0.5 to obtain the final entity vector.

Merging and FeedForward Layer

The three different outputs are obtained from Utt2Vec network, feature engineering module and Ent2Vec network. $Utt2Vec_{out}$ is first concatenated with F_{out} to obtain the following final textual representation $Text_{out}$ of an utterance:

$$Text_{out} = [Utt2Vec_{out}; F_{out}] \quad (3.19)$$

I feed $Text_{out}$ to a linear layer of size 100 with ReLU activation to obtain vector of the same length as Ent_{out} . Cosine similarity between these two vectors are computed and concatenated to obtain 201-dimensional $ConCET_{out}$:

$$ConCET_{out} = [Ent_{out}; Text_{out}; Cos(Ent_{out}, Text_{out})] \quad (3.20)$$

According to [144], cosine similarity represents the normalized likelihood that entity-type Ent_{out} appears in $Text_{out}$. Finally, softmax activation is applied to generate a probability distribution over n possible domains.

3.1.5 Conversational Dataset Overview

In this section, I describe the conversational data collected during the 2018 Alexa Prize and another publicly available dataset called Self-Dialogue. I also describe the algorithm designed to generate synthetic training samples, which will be used to augment the original data.

Amazon Alexa Prize 2018

The data for evaluation of the proposed models is collected from the 2018 Alexa Prize, a competition held by Amazon every year since 2017 to advance conversational AI. My team was one of the 8 semi-finalist teams funded by Amazon for the competition. Users were asked to talk to our conversational bot and give a rating from 1.0 to 5.0 (inclusive) based on their experience.

Obtaining True Labels for Alexa Data

Two hundred conversations from the Alexa Prize data were randomly chosen, which consist of 3,000 utterances and responses. These utterances were manually labeled by three different human annotators, whom I call annotator A, B, and C. The matching and kappa scores between the annotator pairs (A, B), (A, C), and (B, C) are (0.82, 0.78), (0.72, 0.65), and (0.80, 0.75), respectively. Overall, these metrics indicate *substantial agreement* between all annotators. The final true labels were selected by majority voting. When there was no majority, one of the labels was randomly selected. The final distribution of annotated intents is shown in Table 3.2.

Movie	31%	Music	20%	News	16%
Pets_Animal	6%	Sci_Tech	6%	Sports	6%
Travel_Geo	2.5%	Celebrities	2.5%	Weather	1.5%
Literature	1.5%	Food_Drinks	1.5%	Other	1.5%
Joke	1%	Fashion	1%	Fitness	1%
Games	1%				

Table 3.2: Intent distribution in Alexa Data.

I randomly selected 90 conversations for training and 10 conversations for validation. The remaining 100 conversations were reserved for evaluation.

Self-Dialogue Dataset

Self-Dialogue dataset¹⁵ released by one of the Alexa Prize teams [57] is a human-human conversational dataset collected by using Amazon Mechanical Turk. Given a predefined topic, two workers talked about anything related to this topic for 5 to 10 turns. Although this dataset is not comprised of human-machine conversations, it is one of the few publicly available datasets which has a very similar structure to real human-machine conversations, except that the utterances are syntactically richer. This dataset contains 24,165 conversations from 23 sub-topics and 4 major topics (intent): *Movie*, *Music*, *Sports*, and *Fashion*. The intent distribution for the Self-Dialogue dataset is 41.6%, 35.1%, 22.2%, and 1.1% for *Movie*, *Music*, *Sports*, and *Fashion*, respectively.

For training, all subtopics are merged into the 4 major topics. I also filtered 198 conversations that were designed only for transitions from *Movie* to *Music* topics and 216 conversations with mixed *Movie* and *Music* labels because I could not assign a unique label. In addition, some of the utterances in the dataset are non-topical chit-chat utterances. They are mostly used for conversational follow-ups such as *Yes-Answers*, *Backchannel*, and *Conventional-opening*. Since these utterances are unrelated to the set of intents I want to identify, I removed these types in both the training and the test set. To do this, I annotated all the utterances using pre-trained ADAN [50] classifier, which supports 25 topical domains and one *Phatic* domain. The *Phatic*

¹⁵https://github.com/jfainberg/self_dialogue_corpus

domain represents all chit-chat and non-topical utterances and any utterance annotated as *Phatic* is removed from both the training set and the test set. To verify the accuracy of ADAN classifier, I randomly selected 20 conversations and asked one human annotator to label each utterance as *Phatic* or *Non-Phatic*. Based on this setup, inter-annotator agreement of 0.87 and Kappa score of 0.82 were achieved, indicating substantial agreement. The final processed dataset consists of 23,751 conversations (363,003 utterances) on 4 main intents (topics). Finally, I divided the dataset into 70%, 10% and 20% for training, validation, and evaluation, respectively.

A summary of the Alexa data and Self-Dialogue dataset statistics is reported in Table 3.3. Utterances from the Alexa data are significantly shorter (3.07 words on average compared to 9.79 in Self-Dialogue), indicating that often entities may be mentioned without extensive context, e.g., as a response to a system question. The statistics also may clear the significant differences in the human-human and human-machine conversations by all criteria.

Dataset	Words per Utterance	Turns per Conversation	Vocabulary Size
Alexa	3.07	16.49	16,331
Self-Dialogue	9.79	5.84	117,068

Table 3.3: Alexa and Self-Dialogue data statistics.

Synthetic Training Data Generation

A simple yet effective approach is proposed to generate many synthetic utterances for training intent classification models. As will be shown, this ability can be particularly useful for augmenting real data when limited manual labels are available, to train deep neural network models which require large amounts of labeled training data. The approach is summarized in Algorithm 1.

For each topic, a small number of predefined intent templates are created. These templates are designed by engineers who developed each domain-specific module. The rules described in Amazon Alexa developers’ guide¹⁶ were applied in order to capture the most common topic-specific intents and accommodate enough lexical and syntactic variations in the text. The templates contain slots to be filled with either entities or keywords, for example, “Play a *KEYWORD_MUSICGENRE* music from *NER_SINGER*” and “tell me some *KEYWORD_MOVIEGENRE* films played by *NER_ACTOR*”. Each slot starting with NER is filled by an entity from the knowledge base, and each slot starting with KEYWORD is filled using a predefined list of intent-oriented keywords. For instance, the slot *KEYWORD_MUSICGENRE* is randomly filled using a list of popular music genres like *rock*, *pop* and *rap*. I first generated these predefined keywords manually and expanded the lists with the 10 most similar words from WordNet¹⁷ for each keyword. To fill in the entity slots, I used the corresponding lists from the knowledge base (described above), prioritizing the most popular entities, and the most common templates according to domain knowledge and most frequent utterance statistics. While the possible number of generated utterances is the direct product of the number of templates, keyword values, and entity-values, the process ends after a predefined number of synthetic utterances is reached. For the experiments, I control the size of the synthetic dataset with a parameter named ρ . This value is determined based on the number

¹⁶<https://developer.amazon.com/docs/custom-skills/best-practices-for-sample-utterances-and-custom-slot-type-values.html>

¹⁷<https://wordnet.princeton.edu>

of available templates for an intent, importance of an intent, and the overall number of covered intents. I conducted an experiment on this value described in Section 3.1.7. I decided to choose 400K to make a trade-off between time and accuracy and to make the experiments manageable.

Movie	28%	Music	15%	Pets_Animal	13%
Travel_Geo	12%	News	10%	Games	10%
Sports	5%	Sci_Tech	3%	Celebrities	2.5%
Fashion	1%	Weather	1%	Literature	1%
Food_Drinks	0.9%	Other	0.1%		

Table 3.4: Intent distribution in Synthetic Dataset.

Any other external dataset can be incorporated into the synthetic generator above to enrich classes lacking sufficient samples. In the experiments, I did not have as many utterances from *Technology* and *Sports* domain compared to *Movies* and *Music* domains. Hence, I used an open-source Yahoo-Answers question-answer corpus to add questions for these classes. Since human-machine utterances tend to be short, as reported in Table 3.3, I only added questions shorter than 10 words. The final topic distribution of the synthetic dataset is shown in Table 3.4.

Template and Entity-based Synthetic Utterance Generator

```

for intent in intent_list do
  for template in common_topical_templates do
    tmp = read(template);
     $\rho$  = SYNTHETIC_DATASET_SIZE;
    e.g. tmp = "Fun facts for NER_ANIMALS"
    e.g. tmp = "The best KEYWORD_LEAGUE team"
    slot_list = find(slots);
    for entity_type and keyword_type in slot_list do
      for entity in entity_type and keyword in keyword_type do
        if entity in common_entity_list and keyword in common_keyword_list then
          generate_utterance(tmp);
          generate_label();
          if len(dataset) >  $\rho$  then
            | return dataset;
          else
            end
          else
            | continue;
          end
        end
      end
    end
  end
end

```

Algorithm 1: Algorithm to generate the synthetic template-, keyword-, and entity-oriented utterances

3.1.6 Experimental Setup

In this section, I first describe baseline methods in Section 3.1.6. Experimental metrics and procedures are described in Section 3.1.6. All experiments were implemented in Python 2.7 using

TensorFlow 1.12.0¹⁸ library. (For Dataset details, please refer to Section 3.1.5.)

State-of-the-Art Baselines

Three state-of-the-art methods were used as baselines:

- **ADAN** [50]: ADAN was proposed by Amazon for conversational topic classification, and it was trained on over 750K utterances from internal Alexa user data for 26 topics.
- **FastText** [15]: FastText is a text classification model from Facebook Research. FastText operates on character n-grams and uses a hierarchical softmax for prediction, where word vectors are created from the sum of the substring character n-grams.
- **VDCNN** [29]: This model was proposed as a character-based text classification model. VDCNN, like FastText, can model misspelled words (potentially mitigating ASR problems in human-machine conversations) more robustly than word-embedding based models.

Training Parameters

In this section, I describe different parameters for the training of the proposed ConCET model and the baseline methods described in Section 3.1.6.

To train the ConCET model, the parameters for CNN and BiLSTM described in Figure 3.2 were chosen based on my experience and previous literature. Finally, I trained the overall model with an Adams optimizer and a learning rate of 0.001. All experiments for ADAN were conducted using the intent classifier API made available to the teams by the Amazon Alexa Prize [50]. To train the FastText model¹⁹, character 5-grams with word embedding of size 300 were used. Finally, VDCNN results are reported based on a publicly available implementation.²⁰ The results are reported for a 29-layer VDCNN, based on the original paper.

Evaluation metrics. I used two standard classification metrics, **Micro-Averaged Accuracy** and **Micro-Averaged F1** [87], to evaluate the approach.

3.1.7 Results and Discussion

I begin this section by reporting the performance of ConCET in comparison to the baseline models described in Section 3.1.6. Then, I illustrate the impact of the entity, external, and utterance features through a feature ablation study.

Main Results

Table 3.5 summarizes the performance of the models on Alexa and Self-Dialogue datasets. The results show that both variations of ConCET outperform the state-of-the-art classifier baselines Fasttext, VDCNN, and ADAN on Alexa dataset by large margins of 13%, 23%, and 10%, respectively in terms of Micro-Averaged F1 score. Among the baselines, ADAN has the best results on the Alexa dataset, while VDCNN achieves the best results on the Self-Dialogue dataset. All the improvements are statistically significant using one-tailed Student’s t-test with p-value < 0.05.

¹⁸<https://www.tensorflow.org>

¹⁹<https://fasttext.cc>

²⁰<https://github.com/zonetrooper32/VDCNN>

Method	Dataset			
	Alexa		Self-Dialogue	
	Accuracy	F1	Accuracy	F1
FastText [15]	54.54	58.34	79.21	79.32
ADAN [50]	62.01	66.10	46.64	59.66
VDCNN [29]	46.48	48.56	79.98	80.61
ConCET (S)	68.75 (+10.9%)	68.73 (+4.0%)	84.58 (+5.7%)	84.71 (+5.1%)
ConCET (P)	71.46 (+15.2%)	71.72 (+8.5%)	84.59 (+5.7%)	84.66 (+5.0%)

Table 3.5: Intent classification on Alexa and Self-Dialogue datasets, where (S) stands for Spotlight entity linker and (P) stands for the domain-specific PMI-EL entity linker. The relative improvements over ADAN and VDCNN are shown on the Alexa and Self-Dialogue datasets, respectively.

The results indicate that ConCET exhibits the highest Accuracy and F1-score on both datasets, significantly outperforming the baseline models.

Interestingly, the performance of the VDCNN and ADAN methods switches for the human-machine and human-human datasets, as ADAN relies only on keywords, which is not sufficient for complex human-human utterances, while VDCNN exhibits the worst performance for short human-machine utterances. In contrast, ConCET exhibits robust and consistently high performance on both human-human and human-machine conversations.

Detailed Performance Analysis

ConCET is a complex model consisting of different steps built based on deep learning models like CNN and RNN. A comprehensive feature ablation analysis is performed to evaluate the effect of each subsection on the overall performance of the system.

Entity linker evaluation

While entity linking is not the focus of this paper, since entities and their types play a central role in the approach, entity linking performance could have a significant effect on the overall classifier performance. To quantify the downstream effects of the entity linking accuracy, and to understand whether ConCET can operate with inaccurate entity linkers, I manually annotated entity-types for 350 utterances, which contained entities spotted by at least one entity linker. The distribution over classes is similar to that indicated in Table 3.2, with a higher number of utterances from *Movies*, *Music*, and *Travel_Geo* compared to the other classes. Table 3.6 presents the accuracy and F1 values of PMI-EL and Spotlight on different classes of utterances.

The two entity linkers exhibit comparable performance, with PMI-EL showing higher Accuracy on the *Movies*, *Music*, *Travel_Geo*, and *News* intents, but DBpedia Spotlight exhibiting higher overall F1 scores. As I will show later in this section, ConCET can perform well with either entity linker.

Impact of textual representation

To evaluate the impact of the textual representation choices, I conducted a feature ablation study. Table 3.7 summarizes the results, which indicate that all of the implemented components are significantly contributing to the final performance. Both Utt2Vec and TopicDist representations

Class	Entity Linker			
	PMI-EL		Spotlight	
	Accuracy	F1	Accuracy	F1
Movie	80.00	77.19	71.83	78.46
Travel_Geo	80.77	87.50	75.47	82.47
Music	65.51	59.37	64.44	72.5
Sports	70.56	63.16	84.00	91.30
News	76.47	78.78	66.66	70.59
Others	53.68	54.84	50.69	62.12
Overall	68.30	68.18	63.48	72.67

Table 3.6: Accuracy and F1 scores of entity detection by PMI-EL and DBPedia Spotlight entity linkers.

contribute to the classification performance, but the contributions are greater in Alexa dataset, due to a stronger correlation between the keywords with the user intents.

Method	Dataset			
	Alexa		Self-Dialogue	
	Accuracy	F1	Accuracy	F1
CNN	47.59	42.93	79.61	79.73
CNN+ <i>BiLSTM</i> _{pos}	51.60	48.14	82.82	82.75
	(+8.4%)	(+12.1%)	(+4.0%)	(+3.8%)
CNN+ <i>BiLSTM</i> _{char}	52.40	48.65	83.12	83.01
	(+10.1%)	(+13.3%)	(+4.4%)	(+4.1%)
Utt2Vec	54.27	50.84	83.33	83.35
	(+14.0%)	(+18.4%)	(+4.6%)	(+4.5%)
Utt2Vec+TopicDist	55.88	53.09	83.45	83.75
	(+17.4%)	(+23.6%)	(+4.8%)	(+5.0%)

Table 3.7: Intent classification Accuracy and F1 for different textual representations Alexa and Self-Dialogue datasets.

Impact of entity-type representation

The model utilizes two variants of entity-type representations, namely entity-type distribution (TypeDist) and entity-type sequence modeling (Ent2Vec). I evaluate both entity representation vectors separately on both Alexa and Self-Dialogue datasets. Moreover, I report the result when different combinations of the entity representations are joined with the Utt2Vec network. Table 3.8 reports the contribution of each entity representation to the final performance. While both representations contribute greatly to the classifier performance, the effects are greater in the Alexa dataset, due to the strong correlation between the entity-types and the user intents.

Impact of synthetic dataset on ConCET

To evaluate the effectiveness of the synthetic dataset, I augmented the Alexa and Self-Dialogue datasets using the synthetic data described above and re-trained the models. The results are reported in Table 3.9. Even though the synthetic dataset is effective in the real human-machine conversations with Alexa, it has a negligible impact on the Self-Dialogue dataset. I attribute this

Method	Dataset			
	Alexa		Self-Dialogue	
	Accuracy	F1	Accuracy	F1
Utt2Vec	54.27	50.84	83.33	83.35
Ent2Vec	26.93	19.93	52.45	50.32
TypeDist	33.73	25.54	58.95	57.00
Ent2Vec+TypeDist	35.66	26.33	60.22	57.91
Utt2Vec+Ent2Vec	60.26	57.93	84.48	84.83
	(+11.3%)	(+14.6%)	(+1.4%)	(+1.5%)
Utt2Vec+TypeDist	63.46	61.03	84.43	84.71
	(+17.0%)	(+20.0%)	(+1.4%)	(+1.6%)
Utt2Vec+TypeDist+Ent2Vec	64.80	61.59	84.51	84.86
	(+17.4%)	(+23.6%)	(+1.4%)	(+1.8%)

Table 3.8: Ablation study for different entity representations.

effect to the large size of the Self-Dialogue dataset. I argue that even a portion of this dataset is enough for a model to reach its asymptotic performance. To evaluate this hypothesis, I re-trained ConCET in two different settings. First, I randomly sampled 1% of Self-Dialogue dataset and used it as the training set. Then, I added the synthetic dataset to the sampled portion and trained the model again. In the former case, ConCET reached the Accuracy of (72.01 ± 0.1) , while in the latter case it reached the Accuracy of (73.12 ± 0.09) . Each experiment was performed 5 times. This confirms that the size of the labeled dataset is indeed affecting the extent to which the synthetic data can be helpful. An experiment is conducted to determine an estimate for the value of ρ using DBPedia Spotlight as the entity linker. The results are shown in Figure 3.3, which indicate that a value of 400K samples is appropriate for ρ in Algorithm 1, due to the classifier peaking at this point with more than 61% Accuracy.

Train On	Dataset			
	Alexa		Self-Dialogue	
	Accuracy	F1	Accuracy	F1
Synthetic (S)	61.60	57.44	75.62	75.52
Synthetic (P)	62.93	63.83	58.73	59.03
Alexa data (S)	64.81	61.92	-	-
Alexa data (P)	62.93	60.24	-	-
Alexa data+Synthetic (S)	68.75	68.73	-	-
	(+6.1%)	(+10.7%)	-	-
Alexa data+Synthetic (P)	71.46	71.72	-	-
	(+13.5%)	(+19.0%)	-	-
Self-Dialogue (S)	-	-	84.61	85.86
Self-Dialogue (P)	-	-	84.55	84.71
Self-Dialogue+Synthetic (S)	-	-	84.58	84.71
	-	-	(-0.0%)	(-1.3%)
Self-Dialogue+Synthetic (P)	-	-	84.59	84.66
	-	-	(-0.0%)	(-1.4%)

Table 3.9: Performance of ConCET with and without training on the synthetic dataset, where “S” stands for the Spotlight entity linker and “P” stands for domain-specific PMI-EL entity linker.

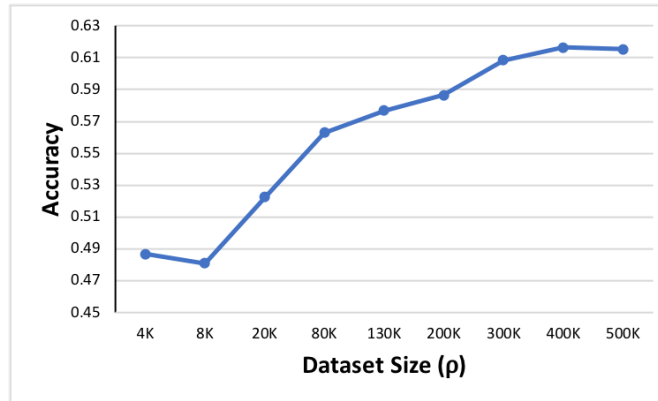


Figure 3.3: ConvCET Accuracy on Alexa Prize dataset for varying ρ values in Algorithm 1.

Discussion

I now discuss the strengths and potential limitations of the proposed approach. Generally, entity-aware classifiers are prone to overfitting to the majority entity-type. I addressed this difficulty by adding sparse and dense representations of the entity-types, which helps in smoothing the entity representation. In other words, using an additional network and separately training the entities reduced the bias towards entity-types. Furthermore, there are entity-types like *Movie_Names*, which are notoriously problematic for classification. For example, the utterance “Fabulous how are you echo” can be easily mis-classified if the entity-aware model is biased toward certain entity-types. In this example, “Fabulous” could be a *Movie_Name*, and “Echo” could be a *City* located in Oregon. In such cases, the ConCET model avoids this error in two different ways. First, because combinations like these appear in all classes, the classifier tends to be less biased to these entities. Second, two different joint deep network layers are used in ConCET model, which makes the system more robust to entity-type errors.

The ConCET model enriches the textual representation of an utterance with entity information for intent classification. By simultaneously learning the text and entity-types, ConCET captures the likelihood of the appearance of a specific entity-type in an utterance text to thereby learn a specific intent label. Moreover, to model semantic (dense) representations of the entity-types, I computed an entity-type sequence as Equation 3.14. The interactions between entity-types, when more than one entity-type appear in the utterance, as well as the order of their appearances in an utterance, can, therefore, be inferred. As a result, ConCET can jointly learn a semantic (dense) representation and the distribution of entity-types with textual information to represent an utterance.

Although ConCET outperforms all of the baselines with either entity linker, I observe higher improvements on Alexa data with the PMI-EL domain-specific linker. I conjecture that this is because PMI-EL is designed to identify the entity-types supported by the conversational agent, which are better aligned with the target domains. In general, entity-types which correspond more directly to the class labels would be more useful as features for classifying intents as opposed to generic entity-types. Nevertheless, ConCET exhibits significant improvements over the previous state of the art with an off-the-shelf generic entity linker, and, when available, can take advantage of the domain-specific entity linking for additional improvements.

A reference implementation of ConCET and the associated entity linker implementations,

training data, models, and the Knowledge Base snapshot was released to the research community²¹.

Deploying a complex system like ConCET in production could potentially degrade system performance by introducing higher response latency. This is an important issue, as response latency has a dramatic effect on the user experience. Interestingly, the classification latency for the proposed approach is not substantially higher compared to the baseline classifier that operates on an utterance text alone. The main reason is that all the 4 stages of the ConCET can be run in parallel. In addition, while entity linking requires a knowledge base lookup, modern in-memory KB storage implementations support candidate entity retrieval and matching in only 10s of milliseconds, which does not introduce perceptible increases to response latency. Finally, ConCET can be executed in parallel for different conversations, allowing the system higher overall throughput without increasing latency for each user.

3.1.8 Limitations

The main limitation of this work is that it uses very simple word embeddings compared to SOTA. Although these embeddings were the best available at the time based on the preliminary analysis on a development phase, I now have significantly more advanced language models. It is possible that the improvements from entity-type features will not hold if I use embeddings from a foundation model like GPT-4.

In the next section, I apply the main components of the ConCET framework, entity knowledge injection and synthetic data generation, to newer embedding models: BERT and T5.

3.1.9 Conclusion

I have successfully demonstrated that entity-type information from external sources leads to better intent prediction when using relatively shallow classification models. The approach, ConCET, uses a combination of text and entity-type features and data augmentation. However, I need to check whether external knowledge provides any benefit if large language models is used for classification or generating embeddings.

In the next chapter, I apply the main components of the ConCET framework, entity knowledge injection and synthetic data generation, to newer embedding models: BERT and T5. Also, I study whether using synthetic data generated using a large language model instead of a template-based approach can lead to better results.

3.2 Contextual Intent Classification for Automated Structured Interviews

In this section, I address the intent classification problem for information elicitation dialog agents. An information elicitation dialog agent (IEDA) is defined as a conversational agent that, given a set of questions and possible answers, will conduct an interview with the user to elicit their preferences and other information. In this context, intent prediction is the task of matching the participant’s response to one or more predefined answer options.

Structured interviews are used in many settings, importantly in market research on topics such as brand perception, customer habits, or preferences, which are critical to product development,

²¹Available at <https://github.com/emory-irlab/ConCET>

marketing, and e-commerce at large. Such interviews generally consist of a series of questions that are asked to a participant. These interviews are typically conducted by skilled interviewers, who interpret the responses from the participants and can adapt the interview accordingly. Using automated conversational agents to conduct such interviews would enable reaching a much larger and potentially more diverse group of participants than currently possible. However, the technical challenges involved in building such a conversational system are relatively unexplored. To learn more about these challenges, I convert a market research multiple-choice questionnaire to a conversational format and conduct a user study. I address the intent prediction task involved in conducting structured interviews, namely matching the participant’s response to one or more predefined options.

3.2.1 Introduction

Information elicitation conversations, such as when a sales agent tries to understand their customer’s preferences or a medical professional asks about a patient’s history, often begin with a routine set of questions. In e-commerce, market research professionals and companies conduct many such surveys each year, often multiple times, before developing, updating, or launching new products - to collect critical data on customer preferences, interests, and awareness, among other topics.

In structured interviews, an interviewer asks a predetermined set of questions conversationally, adapting them to the user’s responses and behavior. While extremely informative and a de-facto standard in market research (e.g., via focus groups), these studies are limited in scale to a small number of participants and are time-consuming and expensive to conduct.

To expand the reach of such studies, online static multiple-choice questionnaires or surveys are used. However, such online questionnaires have some disadvantages. They need to be shorter than interviews to avoid "respondent fatigue" [20]. There is also a greater risk of missing data because of a lack of probing or supervision. Also, it is difficult to ask open-ended questions [20]. Conversational systems that can conduct structured interviews can thus potentially be more effective tools for preference elicitation. Such a system would, given a structured interview provided by a domain expert, converse with the participant to elicit responses to a series of questions. Ideally, it should also be able to ask clarification questions, prime the user with possible answers, and reorder and skip questions based on the user’s responses. An essential requirement for such an agent to be effective is the ability to interpret the responses, often by matching them to a previously defined set of options.

As a first step towards building a conversational system for conducting structured interviews, I investigate the trade-offs of conducting a structured interview via an automated conversational agent vs. the traditional, static, multiple-choice web-based questionnaire. To this end, I conduct a large online user study where a questionnaire with choices for each question is presented in both a conversational interface and as a static multiple-choice questionnaire. The questionnaire was provided by a reputed Personal Care products company’s marketing team. The company has a wide range of products for skin care, which target specific skin conditions. Market research and brand awareness are critical for ensuring that their products meet their consumers’ needs and that they can find the right product.

I then address the intent prediction problem for this setting, i.e., given a structured interview in the form of a list of questions and the set of possible answers (options) for each question, the model needs to infer the options with which the user’s response matches. For the related problem of intent classification for goal-oriented and open-domain conversational agents, prior work achieves good results by jointly training large language models on intent classification and slot-filling tasks.

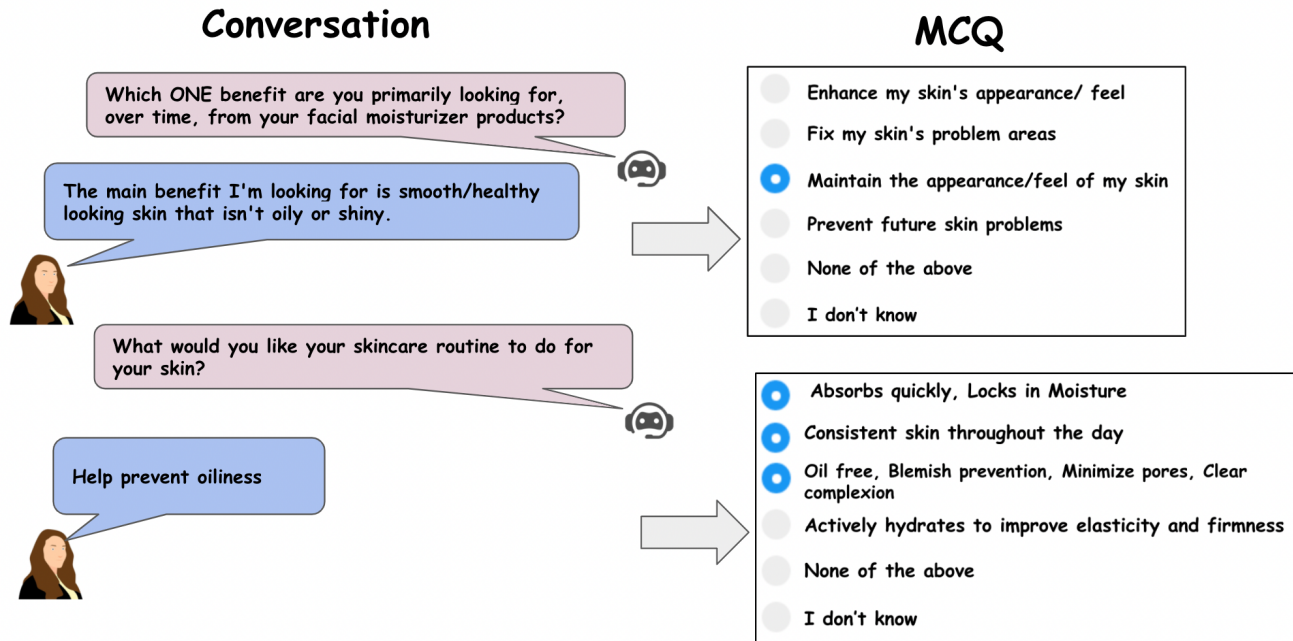


Figure 3.4: The user's conversational responses should be mapped to the correct answer option(s).

However, in a system-initiative conversation where the user is asked open-ended questions about their preferences, intent classification is challenging because 1) interview questions often elicit descriptive answers as opposed to names of entities of an expected type, and 2) it is expensive to collect conversational data for supervised learning. I investigate three approaches for using contextual information for the intent prediction task defined below: 1) using historical probability distribution over the answer options, 2) using previous conversation context, and 3) using external knowledge.

I also analyze the user study data to determine whether the change in interface and the absence of options lead to more informative responses, and what types of questions would benefit from an open-ended conversational interface.

Setting:	Structured interview conducted by a conversational agent with a user
Given:	A conversation consisting of system utterances (in the form of questions) $s_1 \dots s_{n-2}, s_{n-1}, s_n$, and user responses $u_1 \dots u_{n-2}, u_{n-1}, u_n$, and a set of possible answers to s_i given by $A(q = s_i) = a_{i,1}, \dots, a_{i,m}$
Problem:	At conversation turn i , match u_i to a subset M_i of possible answer options $A(q = s_i)$ that represents user intent

Figure 3.5: Intent Prediction Problem Definition

Open-domain and domain-specific conversational agents usually have a predefined set of intents and slot values that they can identify and process. Existing intent classifiers apply a variety of approaches, however, these methods rely on the availability of extensive training data and the intents and slots being limited in number. In the structured interview setting, users often give long descriptive answers to open-ended questions, which makes it hard to apply these intent classification models.

I use TOD-BERT in our experiments to study the advantages of dialog pre-training for our task.

Although our data is domain-specific, there isn't a pre-trained model or knowledge graph tailored for our setting. Therefore, I use ConceptNet neighbors of terms in conversations to experiment with the effects of incorporating external knowledge.

3.2.2 Data Collection

I conducted a user study where participants interacted with both a conversational agent and a static questionnaire. The users' responses in both settings served as a source of training data for intent prediction. I describe the data collection process in this section.

User Study

I conducted a user study with 139 participants to compare the informativeness and other characteristics of *Conversational Interface* responses with *Web-based Questionnaire* responses. I used a questionnaire provided by domain experts from a reputed company, as described in §3.2.1. It contains 25 multiple-choice questions about the client's lifestyle, skin and hair care routines, and preferences. The questionnaire contains 12 single-option questions (the user can select exactly one option) and 13 multi-option questions (the user can select multiple options). The user study consists of 2 phases. In the first phase, the participants interact with a text-based conversational agent that asks a question from the questionnaire, responds to the user's free-form answer with an acknowledgment ("Ok", "Alright" or "I see"), and then proceeds to ask the next question. The participants are then asked to fill out an online web-based survey with the same questions, but this time with options to choose from. They were shown their conversational response to the question and asked to pick the options that matched it. In addition to the responses from the questionnaire, the participants could also choose from two additional options, "None of the above" and "I don't know".

Intent Classification Data

I model the intent prediction task as a binary classification problem. That is, given a <conversational response, answer option> pair, the model predicts the probability that they are semantically equivalent. I use the data from the user study in §3.2.2 as a source of ground truth for <conversational response, answer option> pairs. I split conversations among the train, validation and test sets in a 60:20:20 ratio. I construct a labeled dataset of <conversational response, answer option> pairs from conversations in the train set to train our binary classification models. The <conversational response, answer option> pairs from §3.2.2 are used as positive examples. I add an equal number of randomly selected negative examples. The model is trained on 22865 samples and validated on 7724 samples. It is then evaluated on the holdout set of 20% of the conversations.

3.2.3 Methods

I use a probabilistic model as a baseline because the prior and conditional distributions over the answer options inferred from the training data can intuitively be used to predict the response. I then experiment with fine-tuning pre-trained language models with different context lengths to find the optimal context length for our task. Lastly, I experiment with incorporating external knowledge to improve the model’s performance on out-of-vocabulary terms. This section describes the different methods I use for response prediction.

Using Probabilistic Models Learned from Historical Data

I use purely probabilistic models, which do not consider response text, as baselines.

Context-Less: Using Prior Probability Distributions

In this method, I infer the prior probability distribution over the options for each question using the training data. I infer the probability of an answer option $a_{j,k} \in A(s_j)$ being the match for question s_j as follows:

$$P(M_j = \{a_{j,k}\}) = \frac{\mathcal{N}(a_{j,k})}{\sum_{i=1}^m \mathcal{N}(a_{j,i})} \quad (3.21)$$

where $\mathcal{N}(a_{j,i})$ represents the number of times $a_{j,i}$ is observed as the matching choice M_j for s_j in the training data. The model prediction is therefore $a_{j,k}$, where $k = \operatorname{argmax}_x P(M_j = \{a_{j,x}\})$.

Contextual: Probability Distribution Conditioned on One Previous Response

In this method, I use a conditional probability distribution. Given that $a_i \in A(s_i)$ was the selected option for s_i , the probability that $a_{j,k} \in A(s_j)$ will be selected for s_j , where $i < j$ is given by

$$P(M_j = \{a_{j,k}\} | M_i = \{a_i\}) = \frac{P(M_j = \{a_{j,k}\} \text{ and } M_i = \{a_i\})}{P(M_i = \{a_i\})} \quad (3.22)$$

Intuitively, if the answer to s_i provides some information about the answer to s_j , then $H(M_j) > H(M_j | M_i)$, where $H(x)$ is the entropy of the probability distribution over the values of random variable x .

$$H(x) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (3.23)$$

Our hypothesis is that the answer options matching the user’s conversational responses for different questions in the interview are not independent of each other. Each answer contains information that can be used to infer subsequent answers more accurately. For example, I observe in our dataset that if the user’s response for the question “After applying a facial moisturizer, how do you want your skin to feel?” is known, the entropy of probability distribution over the options for “What type of weather do you usually live in?” is much lower than the prior. I find the conditional probability distribution with the lowest entropy as follows:

$$\operatorname{argmin}_i H(M_j | M_i) \quad (3.24)$$

The model prediction is therefore $a_{j,k}$ where $k = \operatorname{argmax}_x P(M_j = \{a_{j,x}\} | M_i = \{a_i\})$.

Fine-tuning Pre-Trained Language Models

In this approach, I treat intent prediction as a binary classification task. Given a <conversational response, answer option> pair, I train the model to output a score that indicates their semantic similarity. The final prediction is the option with the highest score.

Fine-Tuned BERT Classifier

In this method, I fine-tune BERT [31] to output a score of either 1 (when conversational response and answer option match) or 0 (when conversational response and answer option don't match) when given the conversational response and answer option as input. I employ a linear layer on top of the [CLS] token for classification.

I predict the semantic similarity score of a user response u_j with all the possible answer options for the question s_j as follows: Let q be a question to be posed to the user with option choices given by the function $A(q) = a_1, \dots, a_m$ and let u_n be the user response for the same. Then, I compute all options Y greater than a threshold $t \in [0, 1]$ in the following fashion:

$$S_{j,k} = BERT([CLS]||u_j||[SEP]||a_{j,k}) \quad \forall a_{j,k} \in A(q = s_j) \quad (3.25)$$

The model prediction is $a_{j,k}$, where $k = \operatorname{argmax}_x S_{j,x}$. For multi-correct questions, I can use a threshold to determine the subset of correct answers.

Incorporating Conversation Context

To determine if previous conversational history is useful to better identify the correct choice, I include previous conversation turns in the model input. I experiment with different number of previous turns of context to find the optimal context length for our setting. To keep the conversational nature of the text intact, I append each conversational utterance with either a “[SYS]” or a “[USR]” token depending on whether it is a system or a user utterance. Let t_j represent the concatenation of the j^{th} system and user utterances.

$$t_j = [SYS]||s_j||[USR]||u_j \quad (3.26)$$

I experiment with three settings:

- Context of the current turn j :

$$S_{j,k} = BERT([CLS]||t_j||[SEP]||a_{j,k}) \quad \forall a_{j,k} \in A(q = s_j) \quad (3.27)$$

- Context of 1-previous turn:

$$S_{j,k} = BERT([CLS]||t_{j-1}||t_j||[SEP]||a_{j,k}) \quad \forall a_{j,k} \in A(q = s_j) \quad (3.28)$$

- Context of 2-previous turns:

$$S_{j,k} = BERT([CLS]||t_{j-2}||t_{j-1}||t_j||[SEP]||a_{j,k}) \quad \forall a_{j,k} \in A(q = s_j) \quad (3.29)$$

The model prediction is $a_{j,k}$, where $k = \operatorname{argmax}_x S_{j,x}$.

Incorporating Dialog Pre-training

The BERT model [31] in our previous experiment has been pretrained over free-flowing text and fine-tuned over dialog data. I hypothesize that a model pre-trained on dialog tasks would perform better than a generic pre-trained language model in our conversational setting. In this approach, fine-tune TOD-BERT instead of BERT. TOD-BERT has the same architecture as BERT but has been pre-trained on various dialog tasks.

Incorporating External Knowledge

BERT often does not capture the semantic relatedness of domain-specific terms. For example, it does not know that “dry” and “oily” are antonyms. I hypothesize that leveraging knowledge graphs can help the model better understand the domain-specific terms in our dataset. To bridge the vocabulary gap between the user responses and questionnaire answer options, I concatenate one-hop neighbors from ConceptNet ²² of all the terms in the user input to the user input.

I query ConceptNet for all the one hop neighbors for each word. I do not include bigrams. The list of neighbors often contains multiple occurrences of the same term. This is because ConceptNet has several nodes corresponding to the same concept from different sources. They have different URIs but the same text. For example, for the term “whitehead”, the list of neighbors contains “whitehead” (46 times), “whitehead_link” (2 times), “pimple” (2 times), “white”, “alfred_north_whitehead”, “torpedo”, and “sebaceous_gland”. I remove the terms identical to the original word (“whitehead”). I also remove the terms that appear only once (“alfred_north_whitehead”, “torpedo”, and “sebaceous_gland”). This helps in getting rid of noisy, unrelated neighbors. Lastly, I also remove bigrams (“whitehead_link”). After processing, the list of neighbors for “whitehead” contains only “pimple”.

3.2.4 Experimental Setting

I use 5-fold cross-validation for our experiments. I treat each fold as the test set one by one and use the other folds as train and validation. I report the average of results from all test folds.

Models Compared

I use a naive probabilistic model described in §3.2.3 as the baseline. I compare three variants of fine-tuned BERT to study the effect of dialog pre-training and external knowledge. I list all the models below:

- Probabilistic Baseline: I use the conditional probability-based model described in §3.2.3 as the baseline.
- BERT: I fine-tuned bert-base-uncased²³ on our dataset of <conversational response, answer option> pairs (§3.2.3). I experiment with different lengths of conversation context. Results are reported for the best version, which only considers the current conversation turn.
- TOD-BERT: I also tried a BERT model pre-trained on conversational data. Results are reported for TOD-BERT (described in §3.2.3) fine-tuned on our task with 2 previous turns of context.

²²<https://conceptnet.io/>

²³<https://github.com/google-research/bert/blob/master/README.md>

- BERT-CNNNet: Since our dataset is domain-specific and has a different vocabulary than BERT’s pre-training data, I also experiment with augmenting input to BERT with domain-specific keywords. Again, results are reported for the best version that only considers the current conversation turn. (§3.2.3)

Evaluation Metrics

I train and evaluate our models on single-option questions and multi-option questions separately. I use accuracy as the evaluation metric, which I define as the fraction of test questions where the model assigns the highest score to the true answer option based on the ground truth data described in §3.2.2.

For multi-option questions, where the user can select multiple options, I use ROC-AUC as the evaluation metric.

Human Annotation

I observed that in the user study, in the *Web-based Questionnaire*, the participants often selected options that they hadn’t implied in their *Conversational Interface* responses. To measure how difficult intent classification is for humans, I recruited annotators from MTurk who were familiar with and interested in the domain. I asked them to choose the most appropriate option for each question, given the chat responses from the original user study participant. Four different workers annotated each question for a sample of 27 conversations. I use Fleiss Kappa [34] to measure inter-annotator agreement. The average agreement is 0.46, which indicates moderate agreement. Note that this indicates that the task is challenging for humans as well, and our results should be interpreted as a lower bound on the performance of our models. The agreement varied significantly across different questions, as Table 3.12 shows. The average agreement between the MTurkers and original respondents is 0.44, which is also moderate.

I find agreement is highest for the questions which are general in nature (not domain specific), have a small number of options that are also short and non-ambiguous (e.g. store names v/s descriptions of skin texture).

Table 3.10: Main Results: Accuracy on Single-Option Questions

Model	Overall		On High- κ Questions	
	Accuracy	Std	Accuracy	Std
Prob. Baseline	0.51	0.02	0.53	0.02
BERT	0.64 (+24.0%)	0.04	0.71 (+34%)	0.04
TOD-BERT	0.55 (+7.6 %)	0.04	0.63 (+18.8%)	0.03
BERT-CNNET	0.62 (+20.9%)	0.02	0.68 (+28.3%)	0.05

3.2.5 Results and Discussion

I first report the main results of different methods for intent classification, then discuss findings about user behavior, and finally, investigate the factors that make the task challenging.

Table 3.11: Main Results: AUC on Multi-Option Questions

Model	Overall		On High- κ Questions	
	AUC	Std	Accuracy	Std
Prob. Baseline	0.77	0.01	0.76	0.01
BERT	0.78 (+1.4 %)	0.05	0.79 (+3.6 %)	0.04
TOD-BERT	0.76 (-1.3 %)	0.07	0.76 (-0.1 %)	0.06
BERT-CNNET	0.78 (+2.0 %)	0.05	0.78 (+3.0 %)	0.04

Table 3.12: Questionwise Results: Accuracy is reported for the best performing model; Fleiss Kappa is agreement among human annotators; the last row is the fraction of times annotators chose "None of the above". Response length represents the number of words in the response

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Mean
Accuracy	0.76	0.66	0.76	0.81	0.60	0.71	0.58	0.52	0.40	0.41	0.84	0.69	0.65
Fleiss κ	0.88	0.78	0.78	0.74	0.69	0.49	0.44	0.26	0.22	0.21	0.09	0.04	0.47
Number of Options	2.00	3.00	11.00	4.00	5.00	4.00	5.00	4.00	4.00	3.00	4.00	3.00	4.33
Conversational Dwell Time (sec)	11.79	7.38	6.21	12.42	9.38	12.03	12.23	14.77	14.57	13.43	7.52	6.68	10.70
Conversational Response Length	3.30	2.78	3.44	6.70	3.44	6.41	5.48	7.30	4.19	4.63	4.00	4.19	4.65
Questionnaire Dwell Time (sec)	10.59	4.96	10.93	14.37	9.15	7.52	10.04	8.59	20.70	20.56	7.74	11.30	11.37
Questionnaire Response Length	1.23	1.95	3.65	1.40	7.71	4.26	7.46	2.71	5.04	2.73	4.88	1.42	3.70
"None of the above" answers	0.02	0.03	0.08	0.08	0.14	0.41	0.22	0.58	0.37	0.50	0.02	0.64	0.26

Intent Classification Results

Table 3.10 shows the accuracy of all the models on single-option questions. I consider improvement to be statistically significant if ttest on each fold returns a p-value < 0.05 . Significant results are marked in bold text. Table 3.11 shows the AUC of all the models on multi-option questions. The relative performance of models is the same on both types of questions. The accuracy of TOD-BERT is not significantly higher than our probabilistic baseline. This is because the conversations in our setting are different from the goal-oriented dialog that TOD-BERT is pre-trained on. The model is not able to transfer its knowledge to intent classification in a structured interview.

Fine-tuned BERT and BERT-CNNET significantly outperform the baseline.

The highest value of accuracy achieved is 64%, which is relatively low. As discussed in §3.2.4, the inter-annotator agreement is lower on some questions, indicating that intent prediction on these questions is difficult even for humans. I obtain higher accuracy values by excluding questions with low inter-annotator agreement from our test set. I set the threshold for low agreement as 0.4, which is standard for Fleiss Kappa. This leaves us with 7 single-option questions out of 12. Table 3.10 also shows these results. I compare 4 versions of our models, one that uses just the current chat response as the conversation representation, one that uses the current question and its response, one that includes 1 previous turn of conversation in addition to the current, and lastly, one that includes 2 previous turns of conversation in addition to the current. Unfortunately, incorporating previous turns of conversation does not improve accuracy (decreases by $\sim 4\%$ on average). This might be because each information turn asks a new question. A structured interview flows differently than an ordinary conversation. Results are reported for the version that uses the current question and its response as conversation representation.

Tradeoff Between Effort and Information

Table 3.12 summarizes our findings from the user study. The average dwell time (Time elapsed between the question’s appearance and the user’s first click/keypress) for a question was comparable for *Web-based Questionnaire* and *Conversational Interface*. The input time was much longer for *Conversational Interface* because participants had to type their responses instead of selecting

options with clicks. On average, the *Conversational Interface* response has more words than the *Web-based Questionnaire* response. In some cases, the extra effort on the users' part resulted in more informative answers. For example, for the questions, "When do you moisturize your face?" (Q4) and "How do you handle unexpected stress?" (Q8), the *Conversational Interface* response is significantly more verbose than the *Web-based Questionnaire* response. These questions elicited descriptive answers that were more informative in *Conversational Interface* .

On the other hand, for the question "What kind of hair day are you having today?" (Q5), users were more likely to give a response like "good" or "not bad". Although the longest conversational response for this question had 13 words, on average *Web-based Questionnaire* elicited more informative responses.

I also observe that 26% of the *Conversational Interface* responses annotated by MTurkers were mapped to "None of the above", which indicates that *Conversational Interface* often collects information that is entirely absent from *Web-based Questionnaire* options. The highest number of "None of the above" responses were observed for questions "After applying a facial moisturizer, how do you like your skin to feel?" (Q10) and "How would you describe your natural hair?" (Q12). This might have been because these questions can be interpreted in different ways, but the options list is small and specific.

Error Analysis and Discussion

Table 3.13 shows the correlation between 4 features of questions with the best model's accuracy (Accuracy) and the inter-annotator agreement (κ) for that question. Contrary to what I expected, a larger number of options does not make the task harder for the model or human annotators. The number of words in the conversational response (Conv. Response Length) negatively correlates with κ more than with Accuracy. That might be because longer responses could partially match more than one answer option and cause disagreement. A longer dwell time indicates that the question is hard to understand or hard to answer. It negatively correlates with Accuracy more than with κ . This might be because it is harder for the model to handle unusual responses it hasn't been trained on.

Thus, we can see that the model fails to generalize to unusual responses. Another case where I observe high error is when matching responses requires some logical reasoning. For example, for the question "Which ONE benefit are you primarily looking for, over time, from your facial moisturizer products?", the user responds by saying "The main benefit I'm looking for is smooth/healthy looking skin that isn't oily or shiny". However, the choices in the questionnaire are "Maintain the appearance/feel of my skin", "Enhance my skin's appearance/ feel", "Fix my skin's problem areas" and "Prevent future skin problems". The model would have to infer that the user's response implies that they want to enhance their skin's appearance. The domain-specific nature of the task also remains a source of error. ConceptNet does not have high enough coverage of skincare terms.

	Pearson		Spearman	
	Accuracy	κ	Accuracy	κ
No. of Options	0.18	0.26	0.05	0.04
Conv. Response Length	-0.1	-0.24	-0.21	-0.42
Dwell Time (Conversational)	-0.61	-0.14	-0.59	-0.21
Dwell Time (Online Survey)	-0.58	-0.30	-0.27	-0.25

Table 3.13: Correlation Values

3.2.6 Limitations

Based on the experiments so far, I have found that BERT-based models have limited capability to effectively represent context even when they are pretrained on dialog data.

The knowledge injection approach did not lead to any improvement in performance. That might be because that was not an effective approach of knowledge injection for this setting. Another approach, like the knowledge probing method introduced in [33] might be more suitable. This approach involves integrating knowledge base adapters (repositories of domain-specific information) with large language models.

4 Query Intent Classification for Domain-Specific Web Search

In this chapter, I present the experiments for RQ1c: Can we improve intent classification for health-related queries with external knowledge and data augmentation? This work has not been published yet.

I explore 3 main research questions:

- (i) How do different methods of incorporating external entity knowledge compare for the task of intent classification for health-related queries?
- (ii) How much improvement can we get using external knowledge on models of different sizes?
- (iii) Can we use large language models to generate queries with entities of specific types to augment training data for infrequent classes?

The search results for doctors or locations will be limited to a 10-mile radius from the ZIP code entered. If you'd like to expand that distance, use [Advanced Search](#).

Search for ZIP code (optional) Keywords (optional) [Advanced search](#)

Select doctors or locations. Enter a 5-digit ZIP code. Enter any keywords, name, facility, or location.

Figure 4.1: Example of a widget for the “find doctors and providers” intent. If the user’s query is classified as this intent, they will be given the option to provide their location to find providers near them.

4.1 Introduction

Users often come to a healthcare website with different intents, such as learning about a disease, finding a provider or a hospital, or reading messages from their doctor. Different interfaces are suitable for different types of information, and classifying the user’s intent can help show relevant information in an appropriate interface. For example, the website of the health insurance organization that this work was done in collaboration with has specialized widgets for the “find doctors and providers” (See Figure 4.1) and “health insurance plans” intents. The goal is to surface the appropriate widget to the user based on their query. To enable this, the user queries must be classified into a predefined set of intents. In this context, the intent is the user’s goal or the type of information they are looking for. I model this as a multi-class, multi-label classification problem, where one or more intents are predicted for each query. The setting I focus on in this chapter is health-related web search.

I conduct experiments on two datasets. In one, the intent labels correspond to application-specific classes, and in one where the intent labels represent the topic of the documents that the query should retrieve. (Section 2.1.3 has more details about these settings.) The methods are designed to be general and can be applied to any domain-specific search engine for either of the above tasks. I perform experiments on publicly available web search log data as well as on real-world proprietary data from the search feature on the website of a reputed health insurance organization.

Query classification is challenging because web search queries are short and often do not contain enough contextual information. Moreover, in health-related searches, it is difficult to retrieve relevant information from external knowledge sources because there is a significant vocabulary difference between web search queries and the medical terminology used in most health-related knowledge bases.

Leveraging search results is an effective way of gathering additional context for query classification. (See Section 2.1.3) However, in the setting of a small-scale domain-specific search engine with limited user interaction data, the document or URL that is appropriate for the user’s intent may not be ranked highly or even present in the search results. Also, many queries are navigational in nature, and the user’s intent is to reach a specific page on the website (See Section 4.3.1). In such cases, the search results are not useful for intent classification. I, therefore, propose methods for query classification that do not rely on search results. I use explicit entity information as a source of contextual knowledge. Although only popular Wikipedia entities are used in the experiments, the proposed methods can be extended to inject knowledge about any entity, for example, names of *new* providers or drugs.

Another direction I explore for improving intent classification in this setting is synthetic query generation for frequently misclassified classes. Query generation is often applied to improve the performance of dense retrievers, which require substantial amounts of training data.

I also propose using external knowledge for query generation. However, I do not use documents as context. I instead leverage the vast knowledge of large language models and entity information to generate synthetic queries. In Chapter 3, we saw significant improvements in intent classification in the *conversational* setting by augmenting the training data. In particular, I used a template-based approach that replaced entities in the given query with other entities of the same type to generate synthetic examples. In this chapter, I explore the effectiveness of this method in the *web search* setting. A large language model is prompted to generate queries. Given a query with an entity mention, the model must return new queries with the same intent, but mention other entities of the same type. This method is compared with a baseline data augmentation method that does not explicitly use entities to prompt the LLM.

4.2 Methodology

In this section, the methodology is described for answering the research questions posed in the introduction. Two types of approaches are studied for improving intent classification performance:

- Knowledge Injection (§4.2.1): Here, RQ1ci and RQ1cii are answered.
- Data Augmentation (§4.2.2): Here, RQ1ciii is answered.

4.2.1 Knowledge Injection

Injecting new knowledge into language models has been an active area of research in recent years (Ref. 2.5). In contrast to most prior work, I use explicit *entity information* as a source of external knowledge. I investigate the effectiveness of using entity information for query intent classification through multiple knowledge injection methods. I conduct experiments on four base models of different sizes and report results on two datasets.

I compare three methods of domain adaptation for domain-specific search intent classification:

- Entity Linking + Knowledge Adapters

- Entity Linking + Retrieval Augmentation
- In-Context Few-shot Learning

Entity Linking + Knowledge Adapters

Using knowledge adapters has been shown to be an efficient way to incorporate external knowledge into a model without retraining the entire model. The most impactful prior work in this area, however, uses BERT and RoBERTa as the base model. In [129] the authors introduce K-adapter, a transformer-based adapter layer that is added to the base model to incorporate external knowledge. Two kinds of adapters are added to RoBERTa, one for factual knowledge from Wikipedia, and another for linguistic knowledge from dependency parsing. In [154], the authors show that incorporating entity information into BERT and RoBERTa models (already fine-tuned for a downstream task) using mapped entity embeddings can improve performance on downstream tasks.

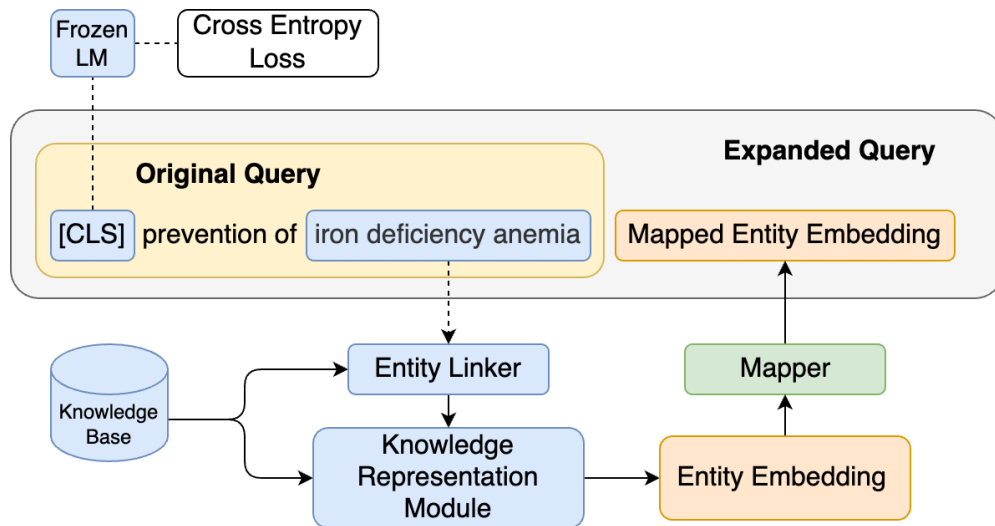


Figure 4.2: Overview of the map tuning method. The adapter is trained to minimize the cross entropy loss on the intent classification task.

Since I want to incorporate entity information similar to [154], I adapt their method for my setting. Figure 4.2 shows an overview of the method. In the following paragraphs, the method is described in detail, using the same paradigm and terminology as used in [154].

Notation: I will refer to the language model fine-tuned on the intent classification task as \mathcal{D} , and the underlying PLM (e.g. BERT) as \mathcal{P} . The method relies on a knowledge base \mathcal{B} that comprises a set of entities and knowledge about these entities. It also requires a knowledge representation model \mathcal{K} , that assigns to each entity an *entity embedding*. An adapter, or *mapper*, \mathcal{M} is used to map the entity embeddings from \mathcal{K} to the language model’s token embedding space. **Method Overview:** Every input sequence is processed as follows. The first step is entity linking, where entities are detected in the query text and linked to \mathcal{B} . The next step is to incorporate the entity information from \mathcal{K} into the model. However, the entity embeddings from \mathcal{K} are not in the same space as the token embeddings of the language model. Therefore, for each entity, the entity embedding is first translated into the language model’s token embedding space using the *mapper*, \mathcal{M} . The mapper is a linear layer that maps the entity embedding to the language model’s token embedding space. The mapped entity embedding is then inserted into the model input right after the text embeddings of the entity mention. (The embedding for “/” is used as a separator between

the entity’s token embeddings and its mapped *entity embedding*.) This input representation is then passed to a language model with a classification head, trained on the intent classification task (\mathcal{D}).

Training the Mapper: The process of training the mapper \mathcal{M} is termed “map tuning”. Two map tuning methods are proposed in [154]:

1. **General map tuning:** Here, the mapper is trained on a Mention-Masked Language Modeling (MMLM) task. This is similar to MLM, but only entity mentions are masked in the input text to ensure that entity information is required to predict the missing tokens. The underlying PLM \mathcal{P} , in combination with the mapper \mathcal{M} , is used to predict the masked entity tokens. The mapper is trained to minimize the cross-entropy loss on the MMLM task, while the PLM is kept frozen.
2. **Task-specific map tuning:** Here, the model parameters are kept frozen while the mapper is trained on the downstream task’s objective. I.e. \mathcal{D} is kept frozen, and \mathcal{M} is trained to minimize the cross-entropy loss on the classification task. The authors show that this method is more effective than general map tuning for improving performance on the downstream task.

The setting in this work is different from [154] in two ways. The first key difference to note here is that the dataset of web search queries used does not contain entity annotations, and entity linking is relatively hard for short, ambiguous text. Therefore, the entity annotations are noisier than for the datasets used in [154].

Another difference is that this method of knowledge adaptation was specifically proposed for the scenario where the downstream model has to use new knowledge *without* any additional training or fine-tuning. That is not a constraint in my setting. I, therefore, finetune the model snapshot to use the mapper effectively after training the mapper.

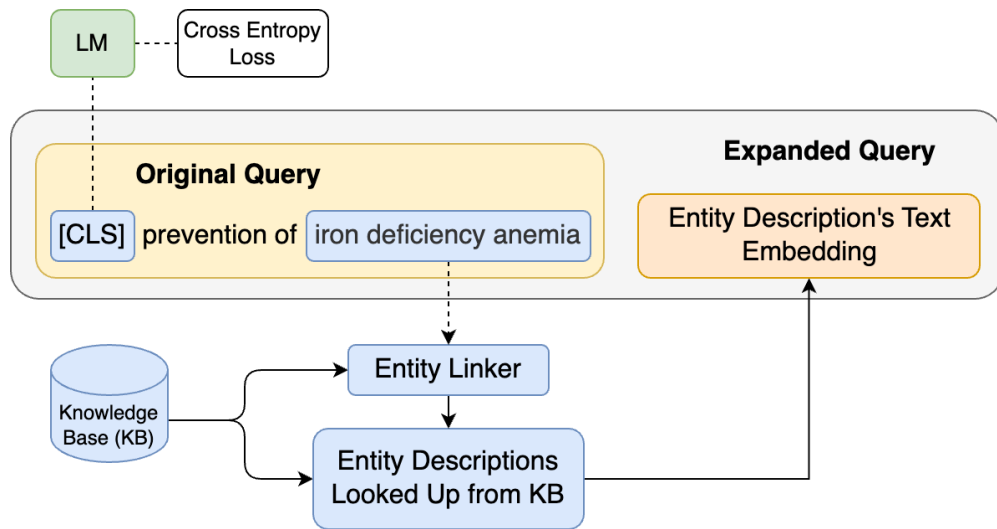


Figure 4.3: Overview of the retrieval aug method. The language model is fine-tuned on the classification task with the augmented input.

Entity Linking + Retrieval Augmentation

A complementary approach to knowledge injection is also explored, in which textual descriptions of entities are used instead of leveraging information from the topological structure of a knowledge graph.

Figure 4.3 shows an overview of the method. The first step in this method is to detect entities in the query text and link them to an associated knowledge base. Then, information about the detected entities is incorporated into the model input by concatenating entity descriptions to the query text. The model is then fine-tuned on the classification task with the augmented input.

Few-Shot In Context Learning

In this method, I rely on a large language model’s vast knowledge for domain adaptation. The model is prompted with few-shot examples to perform the intent classification task. The few-shot examples comprise an equal number of examples from each class. The LLM is prompted to return multiple labels when applicable. But examples with just one correct label are used when possible for in context learning. The prompt describes the dataset and the task and provides examples. Following [136], we assign a persona modifier to the LLM in the prompt, that of a “web search assistant”. The full prompt can be found in the appendix. §A.

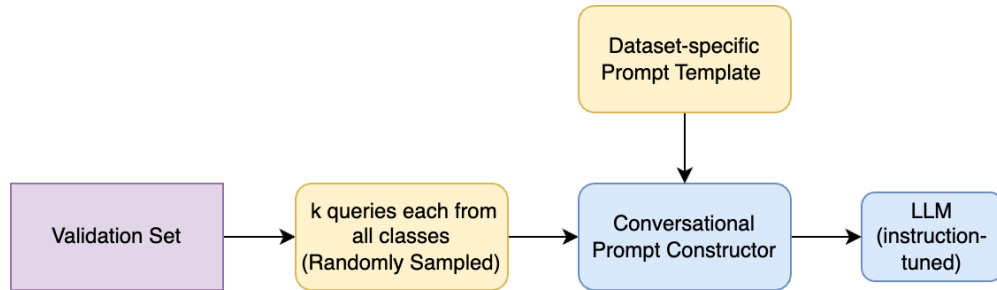


Figure 4.4: Overview of the In-Context Learning method. The large language model is prompted with few-shot examples to perform the intent classification task.

4.2.2 Data Augmentation

Here, the hypothesis that entity information can be used to generate more training data for frequently misclassified classes to improve classification is tested. A large language model is used to generate synthetic training data given a query with an entity mention, the entity type name of the mentioned entity and its intent. The classifier is then trained further on the synthetic queries generated by the LLM.

Figure 4.6 shows an overview of the proposed method. Here, the LLM is prompted with a template that includes the entity type and the intent label. An example prompt for the topic *pediatrics* and the entity *infant death syndrome* is “The query is "risk of sudden infant death syndrome with position supine" and the topic is "pediatrics". Please rewrite the query with another disease.” Here, “disease” is the entity type obtained from the knowledge base. (The granularity of types depends on the knowledge base used. Wikidata, which is used in this work, has several hundred entity types, including broad ones like “type of chemical entity”, “class of disease”, and “medication”; and specific ones like “developmental defect during embryogenesis”) The training data is first filtered to find queries that have at least one entity. We try to use queries with only one correct label to construct prompts. Some classes never appear as the only label in our dataset. In that case, queries with the fewest labels are used. The LLM generates synthetic queries that are then used to augment the training data.

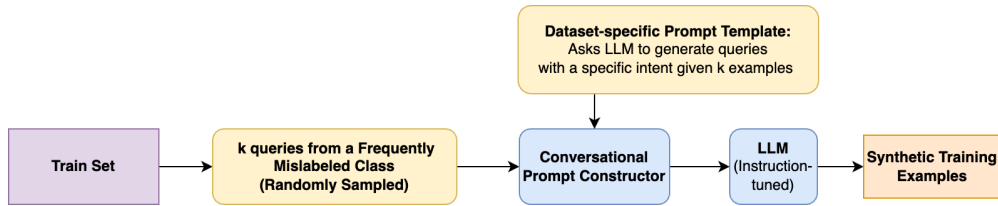


Figure 4.5: Baseline data augmentation method.

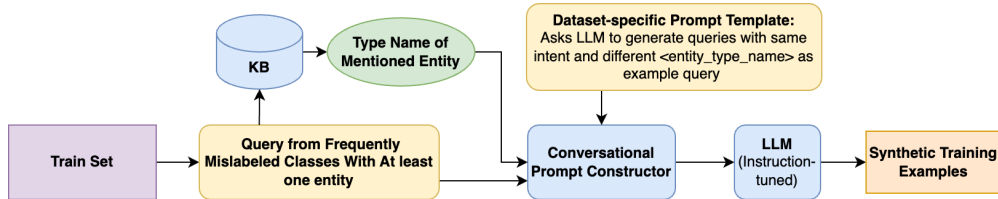


Figure 4.6: Entity-based data augmentation method.

This method is compared with a baseline data augmentation method that does not explicitly use entities to prompt the LLM. Figure 4.5 shows an overview of this method. An example prompt for the baseline method is “Given a topic, you need to generate 10 queries that are likely to retrieve the documents on that topic. The topic is "pediatrics". Here are some example queries with that topic: <list of example queries>”. (The full prompt can be found in the appendix. §A)

4.3 Experimental Setting: Data and Methods Compared

Name	Size	Num Labels	% of Queries with Entities	Train:Val:Test Split
Private Health Search	50475	1	21.154	60:20:20
Tripclick	143269	1.76 (avg)	22.831	60:20:20

Table 4.1: Dataset Description

4.3.1 Data

The impact of the entity-knowledge incorporation and data augmentation approaches is measured on two real-world datasets:

1. TripClick: A large publicly available dataset of click logs from a health search engine
2. Private Health Search: Internal dataset from the search engine of Kaiser Permanente website.

Table 4.1 summarizes the attributes of both the datasets.

Private Health Search (Priv HS) Dataset This dataset includes search logs from a healthcare website’s search engine from January 2022 to September 2023. This data offers insights into user search behavior, including queries and clicked URLs in the healthcare domain.

To build an effective search intent classifier to direct users to relevant pages and widgets, it was first necessary to understand users’ diverse goals on the healthcare website. The user goals were categorized into 14 coarse-grained intents. They broadly fall into three main areas.

1. Self-Service Actions: Queries where users manage their healthcare experience directly through the portal. This includes “Account management”, “Medical records”, “Cost, coverage & benefits”, “Appointments”, and “Communication”.
2. Navigation within the Healthcare System: Queries related to navigating the healthcare system. This includes “Health plans”, “Providers”, “Facilities”, “Health programs & classes”, “Customer support”, and “Careers”.
3. Health Information: Queries related to health information. This includes “Health wellness information”, and, “Drug information”.

Queries that did not fall into any of these categories were labeled as “Other”.

URLs were classified into one of the intent labels with a semi-automatic approach. Healthcare portals often organize content hierarchically, with specific sections for providers, facilities, account management, etc. This structure was used to define URL patterns for each intent category. For example, queries consistently clicking URLs containing the term “email” suggest "Communication" intent. By defining URL patterns for each intent category, we can infer user intent based on clicked URLs. Each query was then assigned labels based on the most frequently clicked URLs for that query.

To get higher quality labels, only the queries with at least 2 clicks were included. Even though our model performs multi-label classification, to minimize noisy labels, we only assign one label to each query, based on its most frequently clicked URL. The dataset is split into training, validation, and test sets in a 60:20:20 ratio. The task is to classify the queries into one of the above predefined coarse-grained intents. This data is not publicly accessible but helps assess the real-world performance of all the methods. Moreover, since this is private data, we can be certain that the queries are not present in the training data of the LLMs (in particular proprietary ones like GPT-4) used for comparison. It gives us reliable estimate of large language models’ performance on this task.

TripClick Dataset TripClick is a large-scale publicly available dataset of click logs in the health domain from the Trip Database health web search engine [104]. The TripClick dataset contains about 5.2 million user interactions collected between 2013 and 2020 and contains the following information: the queries, the clicked documents, and document attributes. The document attributes include the document title, the document URL, and the document topic. For this dataset, the topics of the documents clicked for a query are treated as its intent labels. Only those queries are included that have at least 5 clicks. This dataset has 36 intents, and many queries have more than one correct label. It is split into training, validation, and test sets in a 60:20:20 ratio.

Note that as Table 4.1, both datasets have around 20% of queries with entities. This is a lower bound, as the entity linker used to annotate the queries does not have perfect recall. The entity linker used is ReFinED [119] with Wikidata as the knowledge base. Only the queries where the mention span is linked to a Wikidata entity with high confidence (0.95 or above) are considered to have entities. The knowledge injection methods mainly affect these queries. With a more advanced or domain-specific entity linker, the proportion of queries with entities would be higher.

Table 4.2: Overall Micro F1 and NDCG on TripClick Dataset. Bold means best method for that model, * means statistical significance from the baseline.

Model	Experiment	Micro F1	NDCG
BERT	baseline	0.731	0.848
	retrieval aug	0.731	0.849 (+0.1%)
	map tuning (Gen)	0.733* (+0.3%)	0.850* (+0.2%)
	map tuning (Task)	0.727* (-0.5%)	0.846* (-0.3%)
BERT	data aug baseline	0.647* (-11.5%)	0.792* (-6.7%)
	data Aug	0.683* (-6.6%)	0.818* (-3.6%)
T5-Base	baseline	0.723	0.844
	retrieval aug	0.724 (+0.2%)	0.845 (+0.1%)
	map tuning (Task)	0.721* (-0.3%)	0.843* (-0.2%)
T5-3b	baseline	0.665	0.807
	retrieval aug	0.667 (+0.3%)	0.809* (+0.2%)
	map tuning (Task)	0.665 (-0.1%)	0.807
Llama-3-8b	In-Context Learning	0.644	0.775
GPT-4	In-Context Learning	0.677	0.8014

Table 4.3: Overall Micro F1 and NDCG on Priv HS Dataset. Bold means best method for that model, * means statistical significance from the baseline.

Model	Experiment	Micro F1	NDCG
BERT	baseline	0.891	0.930
	retrieval aug	0.890 (-0.1%)	0.929 (-0.1%)
	map tuning (Gen)	0.894 (+0.4%)	0.932 (+0.2%)
	map tuning (Task)	0.880* (-1.3%)	0.923* (-0.7%)
T5-Base	baseline	0.893	0.932
	retrieval aug	0.895 (+0.1%)	0.933 (+0.1%)
	map tuning (Task)	0.891 (-0.2%)	0.931 (-0.1%)
T5-3b	baseline	0.895	0.933
	retrieval aug	0.897 (+0.2%)	0.934 (+0.1%)
	map tuning (Task)	0.892* (-0.3%)	0.931* (-0.2%)
Llama 3 8b	In-Context Learning	0.631	0.797
GPT-4	In-Context Learning	0.714	0.829

4.3.2 Knowledge Base and Representation

I use Wikidata [124] as the external knowledge base in all my experiments.

For my implementation of map tuning, I follow [154] and use Wikidata5M [130] as the knowledge base. I use TransE [16] embeddings to represent entities. A hashmap is used to look up the TransE embedding for the detected entities. The Wikidata5M dataset was constructed by mapping Wikidata entities to their corresponding Wikipedia pages, and dropping entities with missing pages or with descriptions shorter than 5 words. All relations from Wikidata where both entities belonged to the filtered set were retrieved. This graph was then used to train the TransE embeddings, that I use in the experiments.

4.3.3 Validation Experiments

This section describes the validation experiments that determined the configurations used in the main experiments.

For Retrieval Augmentation

In the preliminary experiments for choosing the best retrieval augmentation method, I also tried a version without an entity linking step. I instead applied dense retrieval over an index of entity descriptions. The full query text was used to find the top k most relevant entity descriptions, which were used to augment the input text.

I experimented with 2 indexes for retrieval:

1. Definitions of concepts/entities from UMLS
2. Descriptions of entities from Wikidata

ColBERT-v2 was used as an efficient dense retrieval model with low latency. I found that retrieval was low-precision, and adding the top- k matched documents to the model input harmed performance on the downstream task. Therefore, for the retrieval augmentation experiments, I report results on the version with an explicit entity linking step to Wikidata, as described in Section 4.2.1.

For Entity Linking

Since our proposed methods aim to leverage entity information to improve intent classification, correctly identifying and mapping entity mentions in queries to a KB is important. Since our queries belong to the healthcare domain, we considered using UMLS [14], a domain-specific knowledge base. It is often used to incorporate domain knowledge for tasks like biomedical question answering [96].

We used a popular and efficient library SciSpacy [91] as the entity linker for the UMLS dataset. Unfortunately, entity mentions were often missed because of the short length of the text and vocabulary mismatch between health search queries and UMLS. We got better results in our validation experiments by using the ReFinED [119] entity linker with Wikidata as the knowledge base. We report metrics that were calculated using the ReFinED entity linker while using Wikidata as the KB in the following sections.

For Map Tuning Approach

As described in Section 4.2.1, [154] propose two approaches for map tuning. I compared both methods for BERT, and the results are included in Tables 4.2 and 4.3. For general map tuning, the Wiki20 [36] dataset was used for Mention-Masked Language Modeling. For task-specific map tuning, the mapper was trained on the multi-label classification task of intent prediction.

As table 4.2 shows, general map tuning leads to slightly better performance than the baseline BERT model. However, task-specific map tuning leads to a decrease in performance. This is inconsistent with the results in [154]. This might be because Wiki20 is a higher quality dataset compared to heuristically determined intent labels in our dataset. Another possible reason is that we do entity annotations using an off-the-shelf entity tagger, while the Wiki20 dataset has precise entity annotations from Wikipedia text and hyperlinks. Even though it performed better in the validation experiment, I did not use general map tuning in the main experiments for the remaining models. This is because the difference in performance is not significant. Moreover Mention-Masked

Language Modeling requires a text corpus that is dense with entity mentions, which would limit the applicability of the method to new entities. Therefore, I use task-specific map tuning for BERT and T5 in the main experiments.

4.3.4 Methods Compared

The main results report classification metrics for two baselines and three methods of domain adaptation. They are described below.

1. In-Context Learning with Large Language Model: The first baseline is a large language model prompted (zero-shot and few-shot) to perform the intent classification task. I report results for Llama3 8b Instruct and GPT-4, which are prompted similarly. This demonstrates the performance of open source and proprietary large language models on the task.
2. Medium-sized models fine-tuned on intent prediction data. The second baseline is BERT, T5-Base, and T5-3b models finetuned on the intent classification task. This approach has been shown to be effective for the related task of joint intent prediction and slot filling in the standard ATIS and SNIPS datasets [23].
3. Knowledge Injection: To answer the first two research questions, we compare two methods of knowledge injection on 3 models of different sizes: BERT, T5-Base, and T5-3b. For all the models, the performance of both types of knowledge injection is compared with that of the base model fine-tuned on the intent classification task.
 - (a) Knowledge Adapter: As discussed in 4.2.1, here, an adapter is used to incorporate entity information into the model input.
 - (b) Retrieval Augmentation: Here, the model input is augmented with text about detected entities, as described in Subsection 4.2.1.
4. Data Augmentation: To answer the third research question, GPT-4 is used to generate synthetic training data for the intent classification task, as described in Section 4.2.2. To investigate whether entity information can be used to generate more training data for frequently misclassified classes, I compare two methods of data augmentation.
 - (a) Entity-based Data Augmentation: The LLM is prompted with a template that includes the entity type and the intent label. The LLM generates synthetic queries that are then used to augment the training data.
 - (b) Lexical Data Augmentation: In this baseline approach, LLM is prompted to generate queries that are likely to retrieve the documents on a given topic. The topic is provided in the prompt. The LLM generates synthetic queries that are then used to augment the training data.

4.3.5 Evaluation Metrics

Two evaluation metrics, Micro F1 and NDCG, measure the model’s ability to classify and rank classes correctly.

4.4 Results

4.4.1 Main Results

Tables 4.2 and 4.3 show the overall results for both datasets. Retrieval Augmentation improves performance in some cases and almost never hurts, which proves that including information about entities has an advantage. However, map tuning often decreases the metrics compared to the baseline. Task-specific map-tuning, in particular, never leads to an improvement for BERT. Thus, neither map tuning nor retrieval augmentation consistently improves intent classification performance on the overall dataset.

The results also show that increasing the size of the model does not necessarily lead to better classification performance.

General vs Task-Specific Map Tuning

Tables 4.2 and 4.3 show that general map tuning leads to a slight improvement in performance for BERT on the TripClick dataset. However, task-specific map tuning leads to a decrease in performance for BERT on both datasets. This is discussed in more detail in Section 4.3.3.

Results on Queries with Entities

Table 4.4 reports the results on only those queries where at least one entity is detected. Here, retrieval augmentation significantly improves classification on queries with entity mentions. Map Tuning actually decreases metrics on the queries with entities for T5. This might be because the mapper for T5 is fine-tuned on the intent classification task using click datasets, which does not impart as much entity knowledge to the mapper as “general map tuning”. Note that this means that if knowledge injection models are only used on queries with entities, they can significantly improve performance over the baseline.

4.4.2 Data Augmentation Results

The preliminary experiments showed that augmenting data for all classes leads to worse intent classification performance. This shows that LLMs are not able to produce realistic web search queries using the baseline or entity-based prompting approach. I report results for the experiment in which the three classes with the lowest F1 scores (and frequency greater than a threshold) were augmented to further examine the effects of data augmentation in Tables 4.5 and 4.6. (The class-wise results showing how these classes were chosen can be found in the appendix. §B) The entity-based data augmentation does lead to bigger improvements in metrics on augmented classes compared to the non-entity-based augmentation method, showing that entity information does improve the quality of the generated queries.

4.4.3 In-Context Learning Results

I experiment with different numbers of few-shot examples for the in-context learning approach with Llama 3 8b. I find that increasing the number of examples stops helping beyond a point. In [2], the authors show that many-shot prompting generally leads to better results, which is in contrast to the results. We experimented with different seeds and the results consistently got worse when examples were increased beyond a certain point. We hypothesize that this is because we

Table 4.4: Micro F1 and NDCG for just the queries where at least one entity was detected. Bold means best method for that model, * means statistical significance from the baseline

Dataset	Model	Experiment	Micro F1	NDCG
TripClick	BERT	baseline	0.687	0.823
		retrieval aug	0.687	0.824 (+0.1%)
		map tuning (Task)	0.662* (-3.6%)	0.808* (-1.9%)
	T5-Base	baseline	0.659	0.805
		retrieval aug	0.678* (+2.8%)	0.819* (+1.6%)
		map tuning (Task)	0.644* (-2.4%)	0.796* (-1.2%)
	T5-3b	baseline	0.588	0.760
		retrieval aug	0.614* (+4.5%)	0.779* (+2.5%)
		map tuning (Task)	0.585 (-0.5%)	0.759 (-0.2%)
	Llama3 8b	In-Context Learning	0.513	0.734
	GPT-4	In-Context Learning	0.559	0.779
	Priv HS	BERT	baseline	0.852
retrieval aug			0.881* (+3.4%)	0.924* (+2.1%)
map tuning (Task)			0.789* (-7.4%)	0.867* (-4.2%)
T5-Base		baseline	0.857	0.909
		retrieval aug	0.877 (+2.3%)	0.921 (+1.4%)
		map tuning (Task)	0.833 (-2.8%)	0.895 (-1.6%)
T5-3b		baseline	0.872	0.919
		retrieval aug	0.882 (+1.1%)	0.925 (+0.6%)
		map tuning (Task)	0.837* (-4.0%)	0.897* (-2.4%)
Llama3 8b		In-Context Learning	0.683	0.797
GPT-4		In-Context Learning	0.767	0.860

Table 4.5: Comparison of Baseline and Lexical Data Augmentation

Experiment	F1	NDCG
Baseline	0.2172	0.804
With Data Aug	0.1882 (-13.38%)	0.791 (-1.61%)

Table 4.6: Comparison of Baseline and Entity-Based Data Augmentation

Experiment	F1	NDCG
Baseline	0.2172	0.804
With Data Aug	0.2202 (+1.34%)	0.8249 (+2.59%)

use a weak supervision dataset. A larger set of few-shot examples would be more likely to include queries that are not representative of the class in question.

Table 4.7: Comparison of Different Number of Shots for In-Context Learning for TripClick Data

Examples per class	F1	NDCG
0	0.6116	0.7508
1	0.6447	0.7757
3	0.6395	0.7744
5	0.6311	0.7701
10	0.3957	0.5451
15	0.5064	0.6199

Table 4.8: Comparison of Different Numbers of Shots for In-Context Learning for Private Health Search Data

Examples per class	F1	NDCG
0	0.4578	0.6564
5	0.6314	0.7971
10	0.5432	0.7276
15	0.6326	0.7882

4.5 Discussion

4.5.1 Ground Truth Data

Collecting ground truth labels for intent classification is expensive and time-consuming. It is especially difficult even for humans without knowing the original searcher’s context [109]. Both datasets are created from real-world search engine logs, and the labels are obtained based on the clicked documents and URLs. Heuristics are used to minimize the noise in the labels. However, the labels are not perfect. That is because, first, if none of the presented documents are relevant, frequently clicked documents will also be irrelevant for that query. Second, some clicked documents are impossible to predict as relevant based just on the query text. Thus, the evaluation metrics on test sets from these datasets are not perfect indicators of the model’s performance. However, the relative performance of different methods on the same dataset is still a reasonable indicator of the effectiveness of the methods.

As proof of the quality of the labels, I calculated Cohen’s Kappa for the intent labels assigned by two human annotators and the automatic method described in Section 4.3.1. The results are shown in Table 4.9. The annotators were asked to label 200 queries from the Private Health Search dataset. The instructions and examples provided to the annotators were the same as those used as input for in-context learning (Section 4.2.1). The values are in the range of 0.4-0.6, which is considered moderate agreement [59]. This shows that the automatic method is able to assign labels that are as consistent with the human annotators as the human annotators are with each other.

	Annotator 1 & Annotator 2	Annotator 1 & Automatic	Annotator 2 & Automatic
Cohen’s Kappa	0.516	0.531	0.438

Table 4.9: Cohen’s Kappa values for different pairs of annotations.

4.5.2 Entity Linking

Many existing methods rely on entity annotations to get the appropriate knowledge to improve the performance of LMs on downstream tasks. However, entity linking for query text is a hard problem because of their usually short length and ungrammatical structure. This means I have to work with noisy entity information, similar to the conversational setting in Chapter 3. Improved entity linking is expected to improve the knowledge injection methods' effectiveness. The current results represent a lower bound on the performance of these methods.

4.6 Limitations and Future Work

The experiments have a few key limitations. I do not perform entity linking on a large-scale knowledge base (like full Wikipedia) or one that includes domain-specific tail entities (like UMLS). I focused on head entities because those were more reliably linked in short input texts in the validation experiments.

Second, I do not fine-tune models larger than T5-3b. The results do not show a direct correlation between model size and domain adaptation method, but that might change if I experimented with more models of different sizes.

Third, I do not use the best possible entity linking method. Since this work was done in a low-resource setting where inference time had to be minimized, I chose a lightweight entity linker with sufficient coverage of entities and put a high similarity threshold to avoid false matches. However, the effect of the quality of entity linkers on the results is an important direction to explore.

4.7 Conclusion

Different ways of incorporating entity knowledge as context into fine-tuned LMs are studied. The results show that input augmentation with entity descriptions is the most effective approach for intent prediction classification models trained using click data and imprecise entity annotations. The results also show that increasing the size of the model does not necessarily lead to better classification performance. The experiments also show that results peak at a small number of examples per class for in-context learning. Since weak supervision datasets are used, a larger set of few-shot examples is likelier to include unrepresentative samples.

The findings of these experiments can potentially lead to better responses and user experience for health-related search. Moreover, the methods for incorporating entity information and generating synthetic training data can also be used for web search in other domains.

4.8 Future Work

The first important direction of future work would be to try other entity linking methods. That might significantly improve the results of the knowledge injection methods, especially map tuning. Another interesting direction would be to combine session context with entity information. Finally, investigating the effect of knowledge injection on larger models than T5-3b could be another interesting direction of future work.

5 User Modeling

In this chapter, I address RQ2: "Can we anticipate the user's next topic of interest by constructing a user profile using conversation context?" Section 5.1 introduces the topic recommendation problem and explores several user representation methods. This work was originally published in the proceedings of CHIIR 2020 as the paper "Would you Like to Talk about Sports Now? Towards Contextual Topic Suggestion for Open-Domain Conversational Agents." My key contributions to this work include:

1. Implementation of code to extract collaborative filtering features from the conversation log data for the implementation of CTS-CF and CTS-Seq-CF models.
2. Implementation of the Collaborative Filtering (CF) baseline model for topic suggestion.
3. Analyzing overall and per-topic performance of all the models compared.

5.1 User Modeling for Topic Recommendation in Open-Domain Conversational System

To hold a true conversation, an intelligent agent should be able to occasionally take initiative and recommend the next natural conversation topic. A topic suggested by the agent should be relevant to the person, appropriate for the conversation context, and the agent should have something interesting to say about it. Thus, a scripted, or one-size-fits-all, popularity-based topic suggestion is doomed to fail.

Therefore, different methods for personalized, contextual topic suggestion are explored for open-domain conversations. Conversational Topic Suggestion problem (CTS) is formalized to more clearly identify the assumptions and requirements. Three possible approaches to solve this problem are also explored, which vary in their representation of the user: (1) model-based sequential topic suggestion to capture the conversation context (CTS-Seq), (2) Collaborative Filtering-based suggestion to capture previous successful conversations from similar users (CTS-CF), and (3) a hybrid approach combining both conversation context and collaborative filtering. To evaluate the effectiveness of these methods, real conversations collected as part of the Amazon Alexa Prize 2018 Conversational AI challenge are used. The results indicate that the user preference modeling methods are effective: the CTS-Seq model suggests topics with 23% higher accuracy than the baseline. CTS-Seq method is found to outperform CTS-CF for this task. However incorporating collaborative filtering signals into a hybrid CTS-Seq-CF model further improves recommendation accuracy by 12%. Together, the proposed models, experiments, and analysis show that conversational context can be leveraged to learn user preferences to enable better topic suggestion.

5.1.1 Introduction

For an open-domain conversational agent to be coherent and engaging, it must be able to drive the conversation to the next topic, and in a way that does not appear scripted. This task is complicated. As for many realistic and complex tasks, extensive knowledge engineering is needed for in-depth domain-specific capabilities, usually handled by specialized components. For a user to remain

engaged, the overall conversational AI system should be able to recommend the next conversation topic (or component) in a natural and coherent fashion. Appropriate topic recommendations are also critical to expose the capabilities of the system to the user, who otherwise may not know that a conversational agent is an expert in particular topics like sports, cars, or video games.

Yet, the right topic to recommend depends on both prior user interests and the conversation context. A topic could be aligned with the user’s interest and still not be appropriate to recommend if it does not fit the context. This makes the problem different from non-conversational topic and content recommendation, where extensive work has been done using content-based [75] and collaborative filtering methods [9, 56, 76]. It is unclear just *how* to adapt recommender system techniques to the conversational setting. In open-domain conversational agents, the recommended items might be agent’s sub-components with dynamically changing content and interactions underneath, specific items, general suggestions, or even clarification questions. In this work, model-based and collaborative filtering recommendation algorithms are extended for topic and content recommendation in the conversational setting. We experiment with different ways to represent user preferences for topic suggestion in the conversational setting. Conversational systems often operate in the cold start setting. This is especially the case in my dataset, where most conversations are with new users without any conversation history. Content-based recommendations have been shown to perform better than collaborative filtering in this setting [76, 126]. Therefore, the primary approach for the conversational topic recommendation is content-based. However, we find that incorporating collaborative-filtering features when they are available improves performance. Other approaches have explored online experimentation (e.g., [66]), and using social media or other metadata (e.g., [22, 97] for recommendation. Unfortunately, these signals are not easily available in the conversational setting. All the attributes of users and their preferences need to be inferred from their interaction with the conversational agent.

The topic suggestion task is different from next utterance suggestion. For example, Yan et al. [145] describe a next-utterance suggestion approach for retrieving utterances from a conversational dataset to use as suggestions, along with the response. The proposed model learns to give suggestions related to the response, to continue the conversation on the same topic. In practice, due to the vast number of possible utterances coming into a social bot, many conversational systems rely on multiple response modules where each response module would be responsible for a particular domain or set of domains [51]. Fine-grained utterance suggestions would be applicable to the implementation of each domain-specific module. However, when the user is passive or gets fatigued with a particular topic, the system needs to switch to a different component with domain-specific capabilities to keep the user engaged. In this paper, we attempt to formalize the problem of suggesting the best next interesting topic.

The contributions are threefold: (1) Formalizing the conversational topic suggestion problem for open-domain conversational agents; (2) Development of a sequential contextual topic suggestion model for this task; (3) Empirical exploration of the effectiveness of model-based, collaborative filtering-based, and hybrid approaches to topic suggestions in the conversational setting. The experiments indicate the value of inferring user preferences from the conversational interaction for contextual and personalized topic suggestion. They also highlight the strengths and limitations of different user preference models for this task.

5.1.2 Conversational Topic Suggestion (CTS): Problem Definition

I now define the conversational topic suggestion problem and introduce the proposed solutions in the following section.

Turn	User Utterance	System Response	Topic	PS	PST	Labels
1	Alexa, let's chat.	Hi! What would you like to talk about?	<i>Music</i>	None	None	<i>chat</i>
2	Tell me recent songs.	Circles by Post Malone, Talk by Khalid, and Bad Guy by Billy Eilish are currently popular. Would you like to talk about Billy?	<i>Music</i>	<i>Opening</i>	<i>Music</i>	<i>Music_accept</i>
3	No I do not.	Alright. How about some info on Khalid?	<i>Music</i>	<i>Music</i>	<i>Music</i>	<i>follow-up</i>
4	Oh, no.	Ok, do you want to hear about some places to travel?	<i>Phatic</i>	<i>Music</i>	<i>Music</i>	<i>chat</i>
5	I love traveling.	Cool! Which country do you want to visit?	<i>Travel</i>	<i>Music</i>	<i>Travel</i>	<i>Travel_accept</i>
6	Somewhere in Australia.	Australia is beautiful, especially the beaches. Do you like beaches?	<i>Travel</i>	<i>Travel</i>	<i>Travel</i>	<i>follow-up</i>
7	Yes.	I recommend Bondi Beach, it is amazing. Wanna here more about it?	<i>Travel</i>	<i>Travel</i>	<i>Travel</i>	<i>follow-up</i>
8	No thanks, let's talk about something else.	Sure, would you like to discuss the news?	<i>Phatic</i>	<i>Travel</i>	<i>Travel</i>	<i>chat</i>
9	No, news is boring.	Sure, Do you want to talk about recent movies?	<i>Phatic</i>	<i>Phatic</i>	<i>News</i>	<i>News_reject</i>
10	Okay, that sounds interesting.	Cool, which genre do you prefer, comedy or drama?	<i>Movies</i>	<i>Phatic</i>	<i>News</i>	<i>Movie_accept</i>
11	I like both.	How about The Favourite by Yorgos Lanthimos?	<i>Movies</i>	<i>Movies</i>	<i>Movies</i>	<i>follow-up</i>
12	I have to go Alexa, bye!	Nice talking to you, good bye!	<i>Phatic</i>	<i>Movies</i>	<i>Movies</i>	<i>chat</i>

Table 5.1: A Conversation example, where PS and PST refer to Previous State and Previous Suggested Topic, respectively.

Features	Description	Example Values
Topic and Behavior $F_1 - F_8$	One-hot encoding for user response for each topic on previous turn, where 1=Accepted, 0=not suggested, and -1=rejected	[+1,0,-1,0,0,0,0,0]
$F_9 - F_{10}$ - Two previous topics	Two previous components that user engaged with	Movies, Music
F_{11} - Previous accepted topic	Previous suggestion that was accepted by the user	Music
F_{12} - Previous rejected topic	Previous topic that was rejected by users	Pets_Animals
User Profile F_{13} - Name	Does user give his/her name	True/False
F_{14} - Gender	What is the user's gender	M/F
F_{15} - Time	Time of the day during the conversation	Morning/Day/Evening/Night

Table 5.2: Dialogue manager state information features used for CTS recommendation. The values are computed up to turn i in the conversation so far.

Consider the example conversation in Table 5.1. While this is not a real user¹, the conversation is typical of those observed with our system during the Alexa Prize challenge. In a regular Alexa conversation, a user may have an initial interest or information need (e.g., “recent songs”) which is handled by a particular system component (in this case, the *Music* component); however, the user might quickly lose interest, and the system (conversational agent) must take the initiative to find the next topic of conversation that this user is likely to be interested in, for example, *Travel*. In the example conversation, the user accepts the suggestion to talk about the topic *Travel*, and a different system component starts interacting with the user to drive the conversation. The next suggested topic *News*, however, is not accepted by the user, and the system has to make another recommendation, which would degrade the user experience.

Conversational Topic Suggestion (CTS) is defined as follows:

Note that this definition focuses on the acceptance of the topic suggestion, and does not explicitly consider the user’s future satisfaction or engagement with the selected topic. This is because the ultimate user satisfaction depends also on the topic-specific component’s ability to engage the user. I also would like to emphasize that CTS is formulated based only on short-term history (conversation or session-level), and not on long-term user interests. In many practical situations, a conversational agent must make coherent topic suggestions for new (cold-start) or inactive users. We focus on topic recommendation in this setting. The models, described next, attempt to capture both the conversation context and the internal system information for this task.

¹Exact user conversations cannot be reproduced due to Alexa Prize terms.

Setting:	Open-Domain mixed initiative conversation with a multi-component conversational agent.
Given:	A conversation C , consisting of a sequence of user utterances $U_{0..i}$, a sequence of system states $S_{0..i}$, and a set of possible conversation topics $t \in T$, (e.g., system components or mini-skills).
Problem:	At conversation turn i , select a topic t_i to suggest for the current user u , to maximize the likelihood of <i>acceptance</i> (i.e., the probability that user u would like to talk about the topic t_i next).

Figure 5.1: Definition 1: Conversational Topic Suggestion (CTS) Problem Statement.

5.1.3 CTS-Seq Approach

For relevant and coherent topic suggestions, it is necessary to consider the conversation context, e.g., the sequence of previous user utterances and system states. For example, if a user is talking about *Movies*, it might be more natural to suggest *Music* as the next topic, as opposed to *Cars*. Also, if a user declined to talk about *Movies* in the past, the system should not suggest this topic or a related topic like *Television* unless explicitly requested. For this reason, a sequence modeling approach is proposed to be used for conversational topic suggestion. I will further show how this approach can be combined with more traditional collaborative filtering-based methods.

Before the specific sequence models are described, we first discuss the conversation *features*, used for both sequence modeling and collaborative filtering-based methods.

System State and User Profile Features

To represent the conversation context, two different groups of features are extracted for each conversation turn, as summarized in Table 5.2. The first group is *Topic and Behavior* features, which represents the user’s previous responses, i.e., the accepted and rejected topic suggestions. These features have the values of **1** for accepted topics, **-1** for rejected topics, and **0** for the topics that have not been proposed yet. This group of features is designed to prioritize the topics that have been accepted **1** or unexplored **0**. *Topic and Behavior* features also model topic classification features (The topic of the utterance as inferred by the topic classifier. We use the classifier described in Section 3.1) and the current conversation context and system state. These features could indicate the historical probability that the current state is a potential topic-switching point, or whether it should be a *follow-up* for the previous topic. The second group of features is *User Profile* features. They contain the inferred gender of the user [-1,1] based on the provided name, and whether they gave their name at the start of the conversation or not (a weak indicator of the user’s openness to sharing information with the bot). Other features like age and location, which are often used for user profiling, are usually not available in the conversational setting. Table 5.2 shows different categories of features that are used in all the CTS-CRF, CTS-CNN, and CTS-RNN models. The values of these feature groups are computed for each conversation turn and stored in separate vectors, which are then concatenated to produce the full conversational state representation, fv , specifically:

$$fv = [F_1; F_2; \dots; F_{15}] \quad (5.1)$$

The user is modeled as a sequence of turns, where each turn is a vector encoding topic and behavior and user profile features. I emphasize that these features will be used for all sequence and CF-based model variations, to explore the trade-offs in modeling, while keeping the actual features constant.

5.1.4 CTS-Seq: Models

In this section, I list three different implementations of the proposed CTS-Seq method. First, Conditional Random Fields (CRF) implementation is described. Then, I describe the CNN-based followed by the RNN-based implementation.

CRF Implementation of CTS-Seq: CTS-CRF

As the first and most straightforward implementation of CTS, the well-known and robust CRF model is used. CRF is an undirected graphical model, which estimates the conditional probability of a sequence of labels (tags) with respect to the observed features, and requires relatively small amounts of training data [58, 148].

Each conversation is represented as a sequence of turns, with observable features extracted from each utterance and system state. Recall that a conversation j is represented as a sequence of turns $Conv_j = [utt_1, \dots, utt_i, \dots, utt_n]$. Then, for each sequence of utterances, a sequence of labels (topics) $[t_1, \dots, t_i, \dots, t_n]$ is generated. The intent classifier described in Section 3.1 is used to assign these labels. The recommended topic t is modeled as the CRF hidden state, and X is the observed variable represented by the features described above. Thus, the CTS-CRF model aims to predict the most likely *next* topic t_{i+1} after observing the first i conversation turns and system states.

More formally, Eq. 5.2 computes the probability of a topic t given the sequence of previous turns and topic decisions, where $Z(X)$ indicates the normalization factor and θ and η are weights that can be tuned using maximum likelihood estimation. Moreover, $f(t_i; X_t)$ and $g(t_i; t_{i-1}; X_t)$ jointly represent the next topic to predict, the context (previous topic) and the features for the current turn x .

$$p(t|x) \propto \frac{1}{Z(X)} \prod_{i=1}^m \exp \left(\sum_{j=1}^m \theta_j f_j(t_i; X_t) + \sum_{k=1}^m \eta_k g_k(t_i; t_{i-1}; X_t) \right) \quad (5.2)$$

The CRF-based implementation of CTS, CTS-CRF, is illustrated in Figure 5.2.

Deep-learning based implementation of CTS-Seq: CTS-CNN and CTS-RNN

Deep learning approaches such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown promising results for different natural language processing tasks, from text classification to dialogue act classification (e.g., [13, 21, 29, 48, 54, 61, 151]). Lee et al. [61] proposed a pipeline of deep learning methods to model a *sequence* of short texts. Inspired by [61], we propose two deep learning models for implementing CTS, namely CTS-CNN and CTS-RNN. CTS-CNN and CTS-RNN respectively use a CNN and a BiLSTM to incorporate

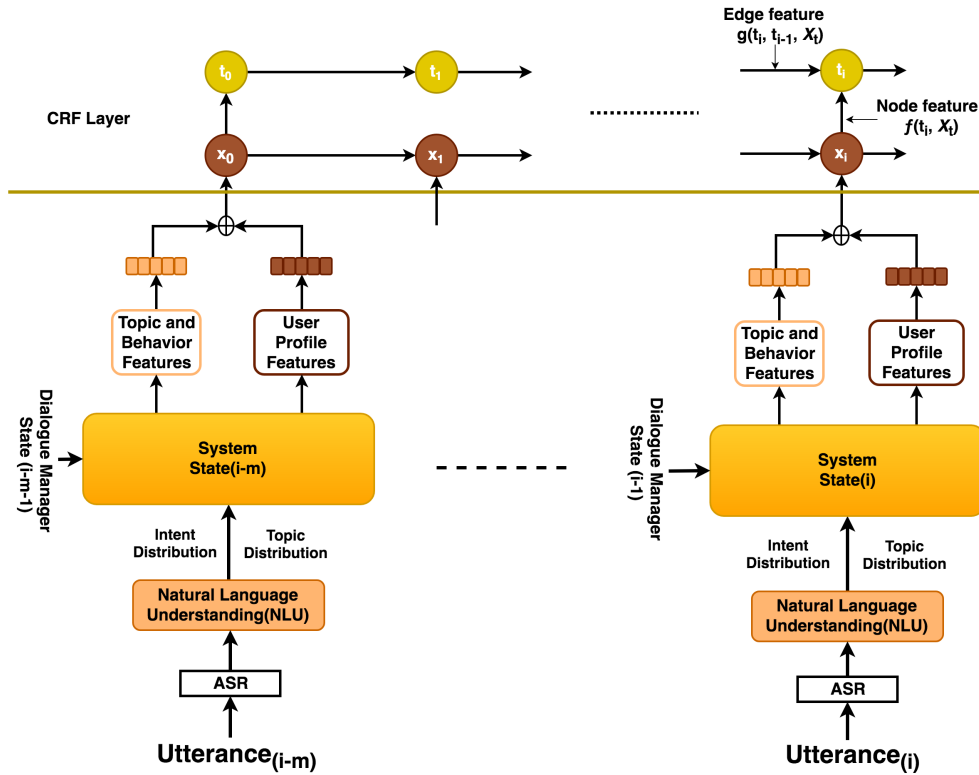


Figure 5.2: CTS-CRF topic suggestion model for conversation turn i . Feature details are reported in Table 5.2 and Section 5.1.3. ASR stands for automatic speech recognition.

textual and contextual evidence gathered so far from the conversation to recommend (predict) the next conversation topic.

CTS-CNN Implementation.

Here, I walk through different steps in the CTS-CNN model. CTS-CNN-CF network takes word tokens, from m consecutive utterances. utt_i stands for the words in the i -th utterance, where w_{ij} stands for j -th word in i -th utterance.

$$utt_{i-m} = [w_{(i-m)1}; w_{(i-m)2}; w_{(i-m)3} \dots w_{(i-m)n}] \quad (5.3)$$

....

$$utt_{i-1} = [w_{(i-1)1}; w_{(i-1)2}; w_{(i-1)3} \dots w_{(i-1)n}] \quad (5.4)$$

$$utt_i = [w_{i1}; w_{i2}; w_{i3} \dots w_{in}] \quad (5.5)$$

A function f_c that takes an utterance as input and outputs the learned utterance representation y_{cnn} is defined:

$$y_{cnn_i} = f_c(utt_i) \quad (5.6)$$

f_c is a 3-layered CNN with max pooling, which is applied in parallel on all the utterances in a window of size m . The first layer is a word embedding layer with pre-initialized weights from

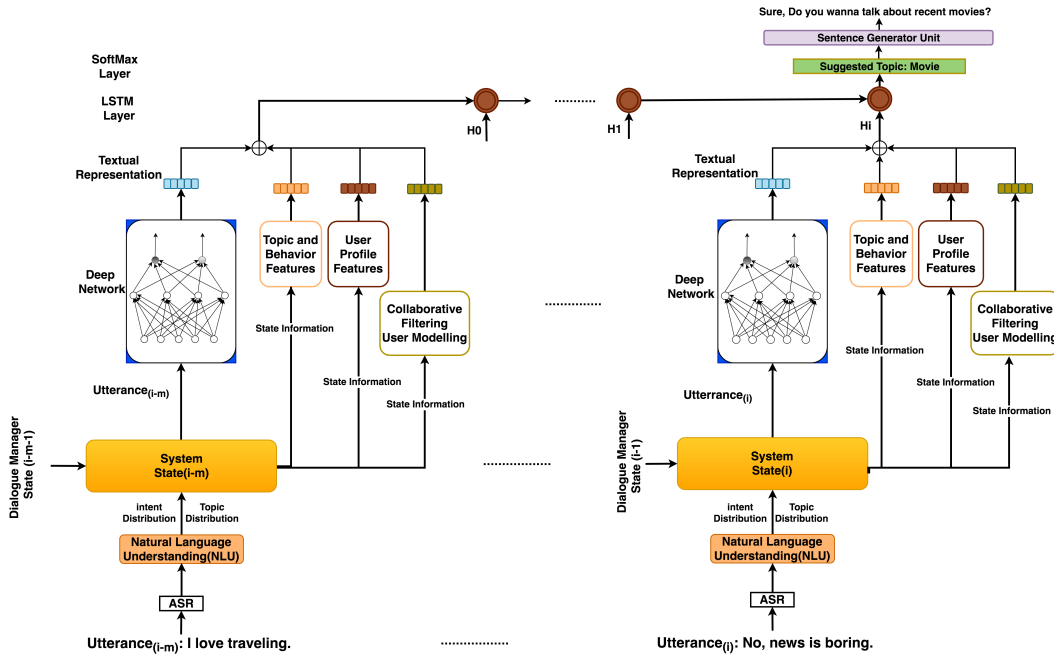


Figure 5.3: CTS-RNN-CF or CTS-CNN-CF model architecture, where *Topic and Behavior* features include the list of all previously suggested, accepted and rejected topics from the beginning of the conversation. *User Profile* features contain the list of Name, predicted Gender, and time of the day. CF features also include the suggested topic distribution extracted from the collaborative filtering model. Feature details are reported in Table 5.2. ASR stands for automatic speech recognition.

Word2Vec² vectors of size 300. The weights on the embedding layer are tuned during training using the cross-entropy loss function.

CTS-RNN Implementation.

CTS-RNN uses a BiLSTM network followed by an attention layer to model the utterance representation. In CTS-RNN, a function f_r is defined that takes an utterance as input and outputs a hidden representation h_i for each utterance:

$$h_i = f_r(utt_i) \quad (5.7)$$

where f_r is a BiLSTM model with 256 hidden layers. It is also applied in parallel on a window of size m in the same way as for f_c . Then, the hidden representation for the i -th utterance is passed to an attention layer to generate the final representation y_{rnn_i} . Given the hidden representations of each timestamp of j in $LSTM_i$ is h_{ij} , dot product similarity score s_{ij} is computed based on a shared trainable matrix M_i , context vector c_i and a bias term b_{ij} . M_i , c_i and b_{ij} are initialized randomly and jointly learned during training. Softmax activation is applied on similarity scores to obtain attention weights α_{ij} . Lastly, using learned α_{ij} , a weighted sum on BiLSTM hidden representations is applied to obtain the output y_{rnn_i} for the i -th utterance as follows:

$$s_{ij} = \tanh(M_i^T h_{ij} + b_{ij}) \quad (5.8)$$

²<https://github.com/mmihaltz/word2vec-GoogleNews-vectors>

$$\alpha_{ij} = \frac{\exp(s_{ij}^T c_i)}{\sum_{j=1}^n \exp(s_{ij}^T c_i)} \rightarrow y_{rnn_i} = \sum_{j=1}^n \alpha_j h_{ij} \quad (5.9)$$

Finally y_{rnn_i} is computed for every utterance located in the window.

Merging and Feed-Forward Layers.

This step is similar for both CTS-CNN and CTS-RNN models, where the output of the textual representation of each utterance is merged with *Topic and Behavior*, and *User Profile* features. Here describe all the details of these layers for the CTS-CNN model are described.

To create the final representation of a w_i in a conversation, y_{cnn_i} is extracted from all the utterances located in the window in parallel. Then, the window is fed to an LSTM network with 100 hidden states. An LSTM is deployed instead of an Bi-LSTM since in real conversation, there is not a backward signal. Finally, the output of the last layer is

$$w_i = [utt_{i-m}; \dots; utt_{i-m+j}; \dots; utt_i] \quad (5.10)$$

$$rep_{w_i} = [y_{cnn_{i-m}}; \dots; y_{cnn_{i-m+j}}; \dots; y_{cnn_i}] \quad (5.11)$$

$$output = LSTM(rep_{w_i}) \quad (5.12)$$

Where, in this equation $j < m < i$. The final output is fed to a feed forward layer of size 256 with a dropout rate of 0.5. A softmax function $f(s)$ is applied to generate a probability distribution over C possible topics. The network was trained with an Adam optimizer with a learning rate of 0.001 using the softmax cross-entropy loss function CE . C is the number of classes, t_i is the one-hot representation of the target label, and s_i are the scores inferred by the model for the i -th class:

$$f(s_i) = \frac{\exp(s_i)}{\sum_{i=j}^C \exp(s_j)} \rightarrow CE = - \sum_{i=1}^C t_i \log(f(output)) \quad (5.13)$$

I summarize the parameters of the (CNN-) and (RNN-) based models in Table 5.3. The parameters were not tuned and were chosen based on my experience and previous literature.

Hybrid Sequential and Collaborative Filtering: CTS-Seq-CF

The new model, CTS-Seq-CF, is not introduced, which augments the sequence modeling approach described above, with additional signals extracted from other users' experiences using collaborative filtering. The proposed models, CTS-CNN-CF and CTS-RNN-CF, incorporate the probability of acceptance of each topic based on similar users' preferences (I describe the collaborative filtering methods used in Section 5.1.5.) as features into the CTS-CNN and CTS-RNN models, respectively. One of the resulting hybrid models, CTS-RNN-CF, is illustrated in Figure 5.3. CTS-CNN-CF follows the same pattern as CTS-RNN-CF, with the semantic utterance representation being generated using a CNN model. They aggregate contextual evidence from the preceding states by considering a window of size m for each turn. Then, all the system state information for the turns in that window is extracted, which includes all the features in Table 5.2, as well as suggested topic

Table 5.3: Detailed configuration parameters of the CTS-CNN and CTS-RNN implementations.

Parameters	Values
Pooling type	max
L2-regularization	0.001
Word embedding length	300
Momentum	0.997
Epsilon	1e-5
Learning rate	0.001
Dropout	0.5
Batch Size	64
Number of layers (CNN)	3
Number of filters(CNN)	128
Filter sizes (CNN)	1,2,3
Hidden state (RNN)	100
Feed forward layer (RNN)	256

distribution predicted by the collaborative filtering method described in Section 5.1.5. Finally, all the utterance vector embeddings within the window are concatenated with them to form the window vector embedding. Here, I walk through the details for CTS-RNN-CF.

y_{rnn_i} is first concatenated with fv_i to obtain the enriched representation $rep_{rnn_i+fv_i}$ of an utterance. Then, they are concatenated with the CF features, which were extracted by collaborative filtering module to generate the final utterance representation rep_{utt_i} .

$$rep_{(cnn_i+fv_i)} = [y_{cnn_i}; fv_i] \quad (5.14)$$

$$rep_{utt_i} = [rep_{(cnn_i+fv_i)}; CF_i] \quad (5.15)$$

To create the final representation of a w_i in a conversation, rep_{utt_i} is extracted from all the utterances located in the window in parallel. Finally, all the outputs are concatenated together to form the final vector.

$$w_i = [utt_{i-m}; \dots; utt_{i-m+j}; \dots; utt_i] \quad (5.16)$$

$$rep_{w_i} = [rep_{utt_{i-m}}; \dots; rep_{utt_{i-m+j}}; \dots; rep_{utt_i}] \quad (5.17)$$

$$output = LSTM(rep_{w_i}) \quad (5.18)$$

Then, rep_{w_i} is fed to an LSTM network with 100 hidden states, later the output of the last layer going through a feed-forward layer followed by a softmax layer as described in Section 5.1.4.

5.1.5 Experimental Setup

I now describe the baselines, data, metrics, and experimental procedures used to evaluate the proposed conversational topic suggestion models.

Baseline 1: Popularity Method

The Popularity method is a heuristic method, which suggests the next conversation topic based on overall frequency (popularity) in previous conversation data and previous user ratings, and the approximate time of day. The order of suggestion is *Movies*, followed by *Music*, followed by *Video Games* or *Travel* or *Animals* depending on the time of day to accommodate the expected differences in user demographics. This heuristic popularity baseline was deployed during the Alexa Prize competition [4].

Baseline 2: Collaborative Filtering (CF)

The classical approach of CF, originally introduced for item recommendation, is adapted to the conversational setting using the K-Nearest Neighbors (KNN) model. Each user is represented by the following features:

- User and time features: F_{13}, F_{14}, F_{15} from Table 5.2.
- Suggestion acceptance and rejection rate: The fraction of topic suggestions accepted by the user and the fraction rejected by the user.
- Topical features: $F_1 - F_8$ from Table 5.2.

For each conversation turn, the feature vector described above is calculated based on the conversation up to this turn. For example, if a user has accepted a suggestion to talk about *Movies* and rejected a suggestion for *Music*, the accept and reject rates would be 0.5. The topical feature vector would contain $\mathbf{1}$ for *Movies* and $-\mathbf{1}$ for *Music*, and then top k users with most similar conversation histories would be retrieved. More formally,

$$\mathbf{U}_a = [F_1(a) : F_8(a), F_{13}(a) : F_{15}(a), r^{accept}(a), r^{reject}(a)] \quad (5.19)$$

$$sim(\mathbf{U}_a, \mathbf{U}_b) = \frac{\mathbf{U}_a \cdot \mathbf{U}_b}{\|\mathbf{U}_a\| \times \|\mathbf{U}_b\|} \quad (5.20)$$

$$pred(\mathbf{U}_a, T) = \frac{\sum_{\mathbf{U}_b \in N} sim(\mathbf{U}_a, \mathbf{U}_b) \times s_{(\mathbf{U}_b, T)}}{\sum_{\mathbf{U}_b \in N} sim(\mathbf{U}_a, \mathbf{U}_b)} \quad (5.21)$$

where \mathbf{U}_a is the user who the topic scores are calculated for, \mathbf{U}_b is one of the neighbors from set N , which is the set of 33 nearest neighbors of \mathbf{U}_a , $r^{accept}(a)$ is the suggestion acceptance rate of user \mathbf{U}_a , $r^{reject}(a)$ is the suggestion rejection rate of user \mathbf{U}_a , $s_{(\mathbf{U}_b, T)}$ indicates the score of topic T for user \mathbf{U}_b , and $pred(\mathbf{U}_a, T)$ represents the predicted score of a topic T for the active user.

For final classification, the predicted topic scores based on 33 nearest neighbors' preferences are fed to a feed-forward layer followed by a softmax layer, as described in Section 5.1.4.

Baseline 3: Contextual Collaborative Filtering: Contextual-CF

Contextual-CF utilizes the collaborative filtering signals extracted from the preceding utterances. Then, a fully connected neural network followed by a softmax is applied to combine the features and provide the final prediction result. To this end, the CF model described in Section 5.1.5 is applied to extract the suggested topics for all the utterances located in a window of size m . To represent the CF features, we considered a one-hot-vector, where the length of the one-hot-vector is equal to the number of available topics that are supported by the conversational agent. The value

corresponding to the topic selected by the CF model is assigned as **1**. Then, the one-hot-vectors are concatenated together to create the final vector for the corresponding window. As a result, a vector of size $[window_size * len(one - hot - vector)]$ is generated. Eq. 5.23 represents the feature vector of w_i .

$$w_i = [utt_{i-m}; \dots; utt_{i-m+j}; \dots; utt_i] \quad (5.22)$$

$$CCF_{w_i} = [CF_{utt_{i-m}}; \dots; CF_{utt_{i-m+j}}; \dots; CF_{utt_i}] \quad (5.23)$$

Where CCF_{w_i} indicates the contextual CF features extracted from i -th window and CF_{utt_i} represents the CF features extracted from the i -th utterance. For final classification, CCF_{w_i} is fed to a feed forward layer followed by a softmax layer as described in Section 5.1.4.

Methods Compared

For convenience, the methods compared in the next section are summarized for reporting the experimental results.

Popularity: A heuristic method, described in Section 5.1.5, using topic frequency in previous conversations.

CF: The collaborative filtering approach, described in Section 5.1.5, using the conversation state (accepted/rejected topic suggestions) as the user profile.

Contextual-CF: The contextual collaborative filtering approach, described in Section 5.1.5, incorporating CF signals from preceding utterances into CF features from the current utterance using a fully connected neural network.

CTS-CRF: The CRF implementation of the CTS approach, described in Section 5.1.4, using only the conversational context (model-based recommendation).

CTS-CNN: The CNN implementation of the CTS approach, presented in Section 5.1.4, using only the conversational context features (model-based recommendation).

CTS-RNN: The RNN implementation of the CTS approach, presented in Section 5.1.4, using only the conversational context features (model-based recommendation).

CTS-CRF-CF: The hybrid model-based and collaborative-filtering based approach, enhancing the CTS-CRF model with collaborative filtering features (Section 5.1.4).

CTS-CNN-CF: The hybrid model-based and collaborative-filtering based approach, enhancing the CTS-CNN model with collaborative filtering features (Section 5.1.4).

CTS-RNN-CF: The hybrid model-based and collaborative-filtering based approach, enhancing the CTS-RNN model with collaborative filtering features (Section 5.1.4).

Dataset: Amazon Alexa Prize 2018

The conversation data were collected by participating in Amazon Alexa Prize 2018 competition [103]. The conversation dataset consisted of 14,707 open-ended conversations longer than four turns (because the first 2-3 turns usually consisted of the required introduction and exchanges of greetings). These conversations were collected from August 1, 2018, to August 15, 2018. The first ten days of conversations were used for training and the rest for testing. The relative topic popularity is shown in Table 5.4. The conversations have an average length of 11.5 turns, where 91% of the conversations contain at least one suggestion, and 60% have at least two explicit topic suggestions.

Table 5.4: Topics distribution in Alexa dataset.

Movie	20.1%	Music	14.4%	News	18.4%
Pets_Animal	10%	Sci_Tech	6%	Sports	6%
Travel	9.1%	Games	6%	Celebrities	2.5%
Literature	1.5%	Food_Drinks	1.5%	Other	1.5%
Weather	1.5%	Fashion	1%	Fitness	1%
Entertainment and Cars	1%				

Table 5.5: Accuracy of the topic suggestion methods compared: Popularity, CF, and Contextual-CF (C-CF) methods, vs. CTS-CRF and CTS-CNN (model-based), vs. CTS-CRF-CF and CTS-CNN-CF (hybrid models). All the results are reported for a window of size five for contextual models. All the improvements are reported based on the strongest baseline, Contextual-CF, where they are statistically significant using a one-tailed Student’s t-test with p-value < 0.05.

Method	Pop.	CF	C-CF	CTS-CRF	CTS-CNN	CTS-RNN	CTS-CRF-CF	CTS-CNN-CF	CTS-RNN-CF
Movies	0.594	0.804	0.827	0.909	0.785	0.794	0.910	0.906	0.909
Music	0.533	0.468	0.807	0.802	0.724	0.741	0.828	0.800	0.813
Travel	0.445	0.863	0.853	0.720	0.801	0.824	0.833	0.875	0.902
Animals	0.425	0.276	0.482	0.780	0.603	0.621	0.812	0.702	0.681
News	0.414	0.164	0.466	0.741	0.518	0.555	0.742	0.543	0.584
Sports	0.232	0.000	0.316	0.621	0.523	0.544	0.663	0.645	0.608
Ent + Cars	0.307	0.752	0.949	0.651	0.831	0.856	0.855	0.881	0.928
Games	0.321	0.010	0.405	0.748	0.572	0.605	0.751	0.689	0.659
Micro-Avg Acc	0.450	0.519	0.640	0.793 _(+23%)	0.669 _(+5%)	0.693 _(+8%)	0.819_(+27%)	0.754 _(+18%)	0.765 _(+20%)
Macro-Avg Acc	0.408	0.482	0.639	0.746 _(+16%)	0.668 _(+4%)	0.692 _(+8%)	0.799_(+25%)	0.755 _(+18%)	0.760 _(+19%)

Evaluation Metrics

To evaluate the approach, the topic suggestion models are computed on off-line data for each of the methods compared. Following the established recommender system research, the following metrics were used:

- **Micro-averaged Accuracy:** The accuracy is averaged across each topic suggestion individually, thus prioritizing more popular topics and potentially longer conversations.
- **Macro-averaged Accuracy:** The accuracy is averaged across each topic class, equally weighing both popular and “tail” topics.

Ground Truth Labels

To create the ground-truth labels, two different scenarios have been followed for training and test data.

For *training*, if a topic t was suggested in turn i , and the user talks about topic t in turn $i + 1$, the label of T_accept is assigned to turn i . If the user rejects the suggestion, or asked for something else, the label was T_reject . Otherwise the label is *follow-up* if the user continues to engage with the same topical component, or *chat* if the utterance is classified as non-informational or *phatic*.

At *test time*, the ground truth labels were assigned as follows: if at turn i , a user rejects the suggested topic T and subsequently, in turn $(i + n)$, requests topic T , then the label for turn i is modified from T_reject to T_accept , because it ultimately matched the user interests. Only

the turns with T_accept labels were used as ground truth labels, because users accepted those suggestions at some point during the conversation. Other turns, without a true (accepted) topic, were not used for evaluation. The same ground truth labels were used for all the baseline and the proposed methods.

Training CTS-CRF Model

To train both Seq-CTS-CRF and CTS-CRF-CF models, a maximum likelihood algorithm is applied, where the parameters are optimized using Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method. For both methods, the context of length five turns is considered. In addition, elastic net ($L1 + L2$) regularization is used to avoid overfitting. Finally, a grid search is deployed to find the optimal values for $L1$ and $L2$, where the values of 0.03 and 0.01 are assigned to $L1$ and $L2$, respectively.

To implement the CF method, the KNN model was trained on the training set with a K value of 33, and the cosine similarity is used as a measure of similarity.

5.1.6 Results and Discussion

I first report the main results of evaluating the proposed topic suggestion models against the popularity-based, CF, and Contextual-CF baselines. I then analyze the recommendation performance for different conversation settings and discuss some limitations of the reported experiments.

Main Results

The main results on most popular classes are reported in Table 5.5. The proposed CTS models outperform the strongest baseline Contextual-CF method, where CTS-CRF, CTS-CNN, CTS-RNN, CTS-CRF-CF, CTS-CNN-CF, and CTS-RNN-CF outperform the Contextual-CF method by 23%, 4%, 8%, 27%, 18%, and 20% respectively. Taking the *sequence* of turns/topics into account leads to significant improvements over a collaborative filtering-based approach.

The results show that CRF significantly outperforms CNN and RNN models, which is surprising for a sequence tagging problem. RNN-based models typically outperform CRF-based methods in similar tasks like entity tagging [24]. We conjecture that CRF outperforms RNNs on this task due to two main reasons: first, the available dataset is relatively small compared to standard entity recognition datasets such as DBpedia [82] and OntoNotes 5.0 [137] with more than 1200K and 1600K samples, respectively. Second, random transitions (e.g., due to dialogue breakdowns) in conversations are more frequent compared to conventional, coherent text. Users usually do not follow a standard conversation with the bot, and may randomly jump between topics. Therefore, even more data are needed to properly model the sequences. However, in contrast to deep RNNs, CRF models need significantly fewer data to be trained.

In general, the collaborative filtering approach appears to perform worse than the other models, including the Popularity-based heuristic baseline (which was manually tuned to optimize the experience of the majority of users). However, incorporating contextual information into the prediction process with CF improves accuracy by 23%. Contextual-CF produces the best results on *Entertainment and Cars*, while it is among the worst results on the other topics like *Games* and *Animals*. We conjecture that this is because *Entertainment and Cars* is a tail topic that few users chose to engage with, and CF is designed to work well for users with rare preferences.

Dropout	Num Filters	#Hidden States	Batch Size	Accuracy
0.5	128	100	64	0.765
0.25	128	100	64	0.762(-0.0%)
0.5	512	100	64	0.774(+1.2%)
0.5	128	300	64	0.770(+0.7%)
0.5	128	100	16	0.764(-0.0%)

Table 5.6: Macro-averaged accuracy for CTS-CNN-CF with different parameter settings.

Method	Context Size	No Features	Topical	User Profile	All Features
CTS-RNN	1	0.563	0.612(+8.7%)	0.601(+6.7%)	0.665(+18.1%)
CTS-RNN	3	0.584(+3.7%)	0.638(+13.3%)	0.621(10.3%)	0.685(+21.6%)
CTS-RNN	5	0.613(+8.8%)	0.674(+19.7%)	0.656(16.5%)	0.693(+23.0%)
CTS-RNN-CF	5	0.701(+24.5%)	0.736(+30.7%)	0.717(+27.3%)	0.765(+35.8%)

Table 5.7: Ablation study on different features of the CTS-based models, where Context Size is measured in conversation turns. All the improvements are reported based on CTS-RNN with no dialogue manager state information feature and no context. The Macro-averaged accuracy is reported, and all of the improvements are statistically significant with p-value < 0.05 .

Similar to Contextual-CF, CTS-CNN, CTS-RNN, and CTS-CRF can effectively capture each specific conversation context, for dramatically more accurate recommendations. In contrast, they are reliable, where they provide high accuracy in all the classes. Interestingly, a hybrid of CTS (model-based) and CF model resulted in a more effective model for topic suggestion, where CTS-CRF-CF and CTS-CNN-CF boost performance by 4% and 9% respectively compared to the CTS-CRF and CTS-CNN models.

Feature Ablation on CTS

CTS-based methods are complex models consisting of different steps built based on deep learning algorithms like CNN and RNN. A comprehensive feature ablation analysis is performed to evaluate the effect of each feature group on the overall performance of the system. Table 5.7 reports the results. Using all the CF, topical, and user profile features in combination, is the most effective approach for CTS-based models. Moreover, the results indicate that the impact of *Topic and Behavior* is higher than *User Profile* information. We conjecture that the *Topic and Behavior* features contain contextual information from previous utterances. Also, as conversations progress, the values of these features are updated for each user. In contrast, *User Profile* information contains static and global information about users, which remain largely unchanged during the conversation, thus having a lower impact.

Parameter Tuning.

To evaluate how parameter tuning contributes to the final results, several experiments were performed with different parameter settings. Table 5.6 shows macro-averaged accuracy of the CTS-CNN method with different parameters.

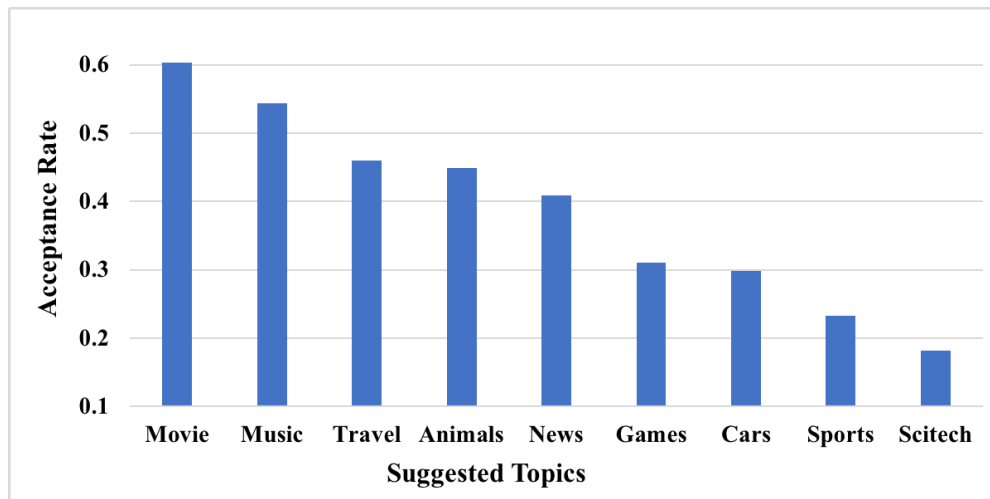


Figure 5.4: Topic acceptance rates in Alexa dataset.

Discussion

I now discuss the strengths and potential limitations of the proposed CTS models on different topics at different stages in the conversation. Finally, I provide the limitations that were encountered during the experiments.

User Topic Acceptance Rate.

Some topics are more popular and interesting for users, such as *Movie* and *Music*. The Popularity baseline described in Section 5.1.5 is designed based on these metrics. Figure 5.4 shows the topic acceptance rate for the most popular topics in Alexa dataset. The results indicate that *Movie* is the most popular topic among users with over 60% acceptance rate, and *Scitech* is the least favorite topic with an acceptance rate less than 20%.

Analyzing CF contribution to RNN- based models.

RNN-based methods are known for finely capturing the contextual information within a sequence. The results in Table 5.5 show that using CF features contributed to the CTS-RNN by extracting relevant knowledge from the dataset that is hidden to CTS-based models. In this specific case, this can be for two reasons, 1) CF features are generated using all the conversation context, while the LSTM model generally considers the history window of size m , and 2) CF features utilize the user-level information like the similarity between user behaviors in accepting or rejecting topics whereas RNN does not consider the user-level information.

Performance for Different Conversation Stages.

As the conversation progresses, the next topic suggestion becomes increasingly challenging, as it is challenging to keep people engaged for long conversations. A proper topic suggestion model could encourage a user to engage more with the conversational agent, which has been shown to be associated with an increase in user satisfaction [25, 123]. Figure 5.5 reports Micro-averaged accuracy for CTS-CRF-CF, CTS-CNN-CF, CTS-RNN-CF, and the baseline Popularity model for a varying number of suggested topics per conversation. Surprisingly, the average accuracy of the

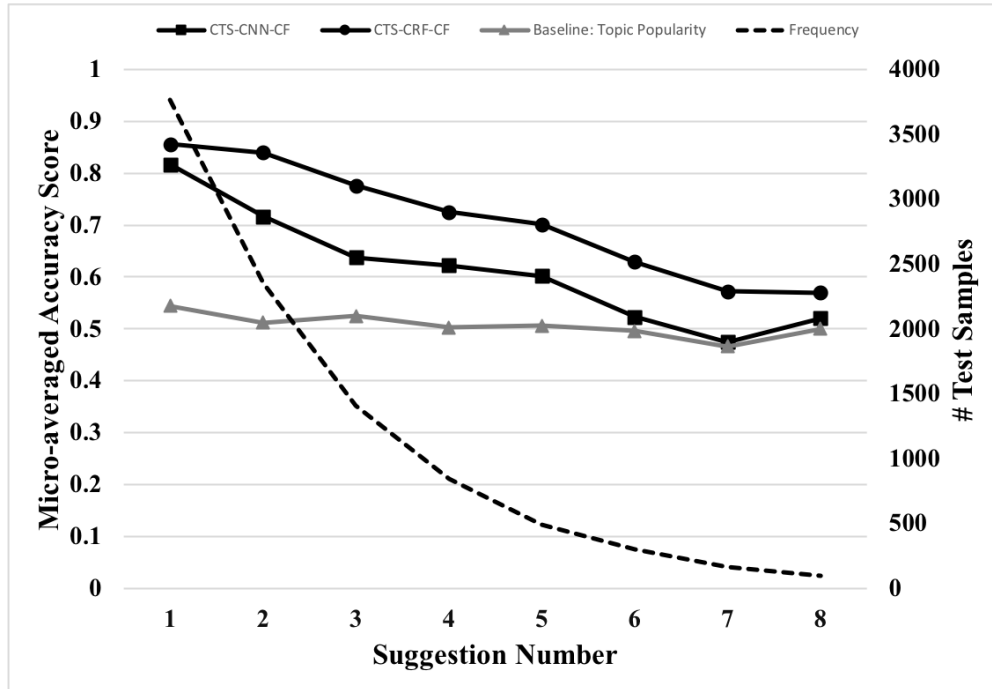


Figure 5.5: Micro-averaged accuracy for CTS-CRF-CF, CTS-CNN-CF, and baseline Popularity model vs. the number of topic suggestions in the conversation.

suggested topic *drops*, as the number of suggestions in a conversation increases. We conjecture that this effect is due to a design decision in the conversational system, where a direct topic suggestion was only invoked if a user was not engaged with the current topic or a domain-specific component has returned conversation’s control back to the main dialogue manager. These situations indicate that the user may already not be sufficiently engaged in continuing a conversation with the conversational agent past the suggestion point. Also, the rejection of the proposed topic may not be (solely) due to the recommendation algorithm but as a result of user fatigue, or other factors. At the same time, fewer people continue talking to the conversational agent for the increased number of suggestions. The vast majority of people only interact with the first one or two suggested topics. Thus, the accuracy of the first handful of suggestions is critical for user experience, as an incorrect first suggestion may cause the user to end the conversation immediately.

Limitations.

The experimental evaluation used offline analysis, and the results might differ in the online setting. However, I do not anticipate that the conclusions would change: I emphasize that the reported results are a *lower bound* on performance since I rely on conversations continuing beyond the current turn in order to give “credit” to the proposed suggestions that were not recommended at the appropriate time during the live competition. Another potential limitation is the form of the recommendations themselves. In this study, the system proposed a general topic like *Sports* for some topics. Still, I found that proposing a specific item for the topic, e.g., “News about the Yankees” instead of just *Sports* may be more effective, and would be a promising complementary direction to the current work, initially explored in [105].

Conclusions

This work introduced and formalized the problem of conversational topic suggestion for mixed-initiative open-domain conversational agents, specifically designed to deliver relevant and interesting information to the user. To address RQ2, I presented and explored three approaches for user preference modeling for this problem, a collaborative filtering based approach, a model-based sequential topic suggestion model (CTS-Seq), implemented using CRF, CNN and RNN models, and a hybrid model which combined sequence modeling approach with traditional collaborative filtering methods (CTS-CRF-CF and CTS-CNN-CF). A hybrid representation that includes both the conversation context and collaborative filtering features outperforms all other approaches in this setting. I showed that contextual, sequence-based recommendation significantly outperforms a heavily tuned, popularity- and time-based baseline, and incorporating collaborative filtering signals further improves performance.

Although CRF outperformed the deep-learning based methods in my experiments, that might not be the case with fine-tuned large language models that have been released since this paper was published. However, the findings about the importance of user-level features and the effectiveness of the hybrid model are likely to remain relevant.

6 Discussion and Conclusions

This thesis presents a thorough investigation of the effectiveness of incorporating external knowledge and synthetic data in intent prediction for different types of conversational systems and for domain-specific web search. It also proposes methods to infer user preferences from conversation history for modular open-domain conversational systems. In this chapter, I summarize this thesis’s contributions, discuss the findings and limitations of the work, and suggest future directions for each research question.

6.1 RQ1: Intent Prediction

The first main research question addressed in this thesis is, “Can we use external knowledge to improve intent prediction?”

6.1.1 RQ1a

In Section 3.1, I address the subquestion “How can we incorporate entity-type information and conversational context to perform intent classification in an open-domain conversational system?”. Intent prediction in this setting is challenging because of short utterance length, out-of-vocabulary words and entities, and a lack of human-machine conversational training data. New entities were a significant challenge in this setting because the conversational system was expected to talk about the most current topics.

The main contribution of this work is the development of ConCET, an entity-aware classifier that incorporates external knowledge about entities and their types, retrieved dynamically from a knowledge base. The second contribution of this work is a simple yet effective method for generating synthetic data for training intent classifiers.

The experiment results show that ConCET significantly improves intent classification performance on a human-machine and a human-human dataset, compared deep learning methods that do not use entity type information. Maintaining an application-specific entity index can lead to further benefits. This thesis thus presents an efficient method to incorporate external knowledge, especially about new entities.

Since the publication of this work, many new intent classification methods have been built on top of larger language models such as BERT. This makes external knowledge less impactful. Although these models encode entity information implicitly, explicitly encoding entity information can improve the performance of these models, especially for unseen entities [64]. A limitation of this work is that the effectiveness of incorporating entity information is not studied for larger models for this setting. However, the general idea of incorporating external knowledge through entities still remains applicable. The effectiveness of incorporating entity information for improving intent classification for domain-specific web search is demonstrated in Chapter 4 for a larger model.

6.1.2 RQ1b

The next research question I studied is “In a conversational system that conducts interviews, how can we interpret the participant’s response, for example, by matching it to one or more predefined

options?” I investigated intent recognition in the unexplored setting of an information elicitation dialog agent (IEDA). This setting, where the user is asked open-ended questions about their preferences, poses unique challenges because 1) interview questions often elicit descriptive answers as opposed to names of entities of an expected type, and 2) it is expensive to collect conversational data for supervised learning.

The first main contribution is the investigation of three approaches for using contextual information for this task: 1) using previous conversation context, 2) using dialog pretraining, and 3) using external knowledge. The second main contribution is findings about what factors can predict the difficulty of a question for a conversational system to interpret. An additional contribution is an analysis of user study data to determine what types of questions would benefit from an open-ended conversational interface.

The experiments show that it is hard to leverage conversational context and external knowledge for this task. The TOD-BERT model with dialog pretraining is not able to transfer its knowledge to intent classification in a structured interview. Including previous conversation turns also did not improve accuracy. The best results are obtained by using the current question and its response for conversation representation. Even though responses to previous questions can potentially contain clues about the current question, BERT-based models are not able to leverage this information effectively. These findings indicate that intent prediction in automated structured interviews will take models with strong reasoning capabilities to interpret contextual information.

Data from the user study indicates that using a conversational interface often elicits information that is entirely absent in the survey form and also gets more verbose responses. This suggests that the conversational interface is more engaging and can potentially lead to more accurate responses.

I also study the attributes of questions that might be indicators of model and human accuracy on them. A larger number of options does not make the task harder for the model or human annotators. Response length negatively correlates with model accuracy, but the correlation is not high. This suggests that the difficulty comes from the complexity of the response more than its length.

This work has some key limitations. The first is that we do not explore more advanced knowledge injection methods. The lack of improvement in performance by incorporating external knowledge might be because we used ConceptNet neighbors to augment input instead of a more advanced approach.

Second, we use a simple dialog policy where the system just returns a generic acknowledgment of the user’s response and asks a new question on each term. Thus, the conversations in our dataset are not perfectly representative of the structured interview setting.

An interesting direction for future work could be collecting or generating a dataset of more realistic interview conversations, where the system can ask follow-up questions and engage in a more natural conversation. Another future experiment could be to use LLMs with Chain-of-Thought prompting and conversation context to perform intent classification in this setting. This could potentially improve the performance of the model by allowing it to leverage the context of the conversation and external knowledge more effectively.

6.1.3 RQ1c

The third research question I address is “Can we improve intent classification for health-related queries with external knowledge and data augmentation?”

Knowledge injection is challenging in this setting because the common method of leveraging search results cannot be used. In the setting of a small-scale domain-specific search engine with

limited user interaction data, the document or URL that is appropriate for the user’s intent may not be ranked highly or even present in the search results. Also, my queries are navigational. Moreover, web search queries are short and often do not contain enough contextual information. Also, in the setting of health-related search, it is hard to retrieve relevant information from external knowledge sources because there is a significant vocabulary difference between web search queries and the medical terminology used in most health-related knowledge bases.

The results indicate that incorporating entity descriptions in the model input is a simple and effective way of improving intent classification for domain-specific web search on models of all sizes compared. The map-tuning method did not show the expected improvement. This shows that for the mapper to be effectively trained to leverage external entity embeddings, it needs to be trained on data with higher-quality entity labels. That was the case in the original paper. Moreover, the data augmentation results show that entity information does improve the quality of the generated queries, but the data quality is still not high enough to improve the performance of the intent classifier. This suggests that generating synthetic data for this task is challenging. Lastly, performance peaks early for in-context learning with LLMs when the few-shot examples come from weak supervision data.

The methods investigated are applicable in small-scale, domain-specific search engines that cannot use web search results as context. An application-specific entity database would further improve performance, like for ConCET.

The experiments have a few key limitations. I do not perform entity linking on a large-scale knowledge base (like full Wikipedia) or one that includes domain-specific tail entities (like UMLS). I focused on head entities because those were more reliably linked in short input texts in the validation experiments. Second, I do not fine-tune models larger than T5-3b. The results do not show a direct correlation between model size and domain adaptation method, but that might change if I experimented with more models of different sizes. Third, I do not use the best possible entity linking method. Since this work was done in a low-resource setting where inference time had to be minimized, I chose a lightweight entity linker with sufficient coverage of entities and put a high similarity threshold to avoid false matches. However, the effect of the quality of entity linkers on the results is an important direction to explore.

There are several possibilities for future work. The first important direction would be to try other entity linking methods. That might significantly improve the results of the knowledge injection methods, especially map tuning. Another interesting direction would be to combine session context with entity information. Finally, investigating the effect of knowledge injection on larger models than T5-3b could be another interesting direction of future work.

6.1.4 Relevance in the LLM Era

Large language models have accomplished impressive results on many NLP tasks, often rendering specialized models unnecessary [1, 120]. Fine-tuned LLMs are currently the predominant form of dialog systems in practice, especially as “support” tools on top of existing applications like e-commerce and customer service.

Currently, large language models are costly in terms of computation resources. There are several complementary directions of research to address this issue. There is some promising work [41, 45] on using sparse mixture-of-expert models to drive down the cost. Which expert to use is determined implicitly in these models, and explicit intent prediction models aren’t needed. If the cost can be brought down further, modular systems that use intent prediction might become obsolete.

Oh the other hand, small models are relevant in the *on-device* setting, where the model must

run with limited computational resources. On-device models are also an active area of research [32], and the methods proposed in this thesis could be useful in this setting. Multi-agent AI systems are also an active and promising area of research [44, 141]. Although explicit intent prediction has not been explored in this setting, it could be applied for more efficient communication between agents.

Another relevant question is, would knowledge-aware methods be necessary if the LLM were trained on all publicly available intent prediction data? This is hard to predict, but the answer might be yes, because the LLM would still not have access to private, application-specific data.

6.2 RQ2: User Modeling

6.2.1 RQ2a

I also addressed the research question, “In an open-domain conversational system, how can we represent the user to anticipate their interest in new topics of conversation?” At the time I did this work, coarse-grained topic suggestion and, therefore, user modeling for this purpose were unexplored.

The right topic to recommend depends on both prior user interests and the conversation. A topic could be aligned with the user’s interest and still not be appropriate to recommend if it does not fit the context. I explored adapting existing recommendation models to this setting. It remains an open problem, but many new methods have been proposed since then, which make this work much less relevant.

The main challenge in this setting was creating a user representation based on just the current conversation. I showed that contextual, sequence-based recommendation significantly outperforms a heavily tuned, popularity- and time-based baseline, and incorporating collaborative filtering signals further improves performance.

This work, however, has several limitations. The text representation models used in this work are very simple compared to the state-of-the-art models. The results about the relative performance of the models compared might not hold when more advanced models are used.

The findings about the importance of user-level features and the effectiveness of the hybrid model are likely to remain relevant. Even for the latest models, topic recommendation in mixed-initiative dialog systems is not a solved problem. However, newer representation methods would serve as a much better starting point for future research than the models used in this study.

6.3 Limitations

The overall experimental setting has several limitations. First, for every research question, I only use a conversational system of one domain and modality. I address RQ1a and RQ2a in the context of an open-domain voice-based conversational assistant, while I use a text-based conversational system for RQ1b. The findings may not generalize to conversational systems in other modalities.

Also, I assume a static set of intents, which is often not the case. Life-long Intent Detection is an active area of research. [72, 73, 121, 156] propose approaches to address the problem of catastrophic forgetting in lifelong intent detection. I do not perform online learning in this work. The approaches only work in settings where the set of intents is known in advance and does not change often.

For my work on user preference modeling, I conducted the experiments only on the data collected using our Alexa Prize conversational system. Although the methods I propose are generalizable to other conversational systems, I do not have experimental results to support this claim.

Fairness and transparency are also important considerations for any system that interacts with users. Although I have tried to be as inclusive as possible in the system design, recommender systems are prone to be biased to the types of users in their training data, and that is the case for my systems as well. Also, in Section 5.1, I use inferred gender as a feature for the user model, and the possible values for the feature are male and female, which is not inclusive of non-binary genders. This was because I relied on a name database that did not have non-binary names. However, this is a limitation that must be addressed before deploying the system in the real world.

6.4 Conclusions

This dissertation introduces novel methods for intent prediction and user modeling. Specifically, it makes the following contributions:

1. Methods for knowledge-aware intent prediction in three settings
2. Approach for user preference modeling in open domain conversational systems
3. Investigation of the response interpretation problem in information elicitation dialog

These contributions are relevant for improving information relevance and user experience in various conversational systems.

One of my main contributions is in the area of intent prediction. With more accurate intent prediction, conversational systems can provide more relevant and appropriate responses, resulting in a more natural and satisfying user experience. This contribution has the potential to improve user engagement and the overall effectiveness of conversational systems. The approach for intent classification using data augmentation and external knowledge can be used in any low-resource setting. This is critical for the functioning of conversational systems. I also introduce an approach for user preference modeling to enable personalization in conversational systems. I test this method for intent prediction in open-domain settings, but it is generalizable to domain-specific systems as well.

Thus, by addressing critical challenges in intent prediction and user modeling, this research lays the foundation for more contextually aware conversational systems. As conversational AI becomes increasingly prevalent in various applications, this can lead to greater user satisfaction in many applications.

Acknowledgements

This work was supported by Alexa Prize 2018, Procter & Gamble, and Kaiser Permanente.

Appendix A

Prompts for Experiments with LLMs

This chapter lists the prompts used for experiments with generative large language models. All prompts were written as chat completions in JSON format. The words surrounded by curly braces ({}) are not put in the prompt verbatim but replaced by the text they describe.

A.1 Prompts for Synthetic Data Generation

A.1.1 Baseline

- TripClick

```
[{"role": "system",
  "content": "You are an AI assistant that helps generate realistic
  training data for query classification models."},
{"role": "user",
  "content": "Your task is to generate plausible web search queries
  from the healthcare domain for training a query classification
  model. The queries are from the TripClick dataset, which contains
  click data from a web search engine. The clicked documents have
  topics associated with them. Given a topic, you need to generate
  10 queries that are likely to retrieve the documents on that
  topic. The queries should be stylistically similar to the given
  examples. Respond with just the queries, with each query in a new
  line. The topic is {topic}. Here are some example queries with
  that topic:{list of examples, each in a new line}"]]
```

- Private Health Search

```
[{"role": "system",
  "content": "You are an AI assistant that helps generate
  realistic training data for query classification models."},
{"role": "user",
  "content": "The queries are from a health insurance website.
  Users typed these queries looking for different types of
  information. They clicked URLs that were relevant to their
  queries. The URLs have intents associated with them, and the
  queries are classified into the same intents. Given an intent
  , you need to generate a query that would be issued by a user
  with that intent. Given an intent, you need to generate 10
  queries that are likely to be issued by a user with that
```



```
intent. The queries should be stylistically similar to the
given examples. Respond with just the queries, with each
query in a new line. The intent is {intent}. Here are some
example queries with that intent:{list of examples, each in a
new line}"]]
```

A.1.2 Entity-based

- TripClick

```
[{"role": "system",
  "content": "You are an AI assistant that helps generate realistic
  training data for query classification models."},
 {"role": "user",
  "content": "Your task is to generate plausible web search queries
  from the healthcare domain for training a query classification
  model. The queries are from the TripClick dataset, which contains
  click data from a web search engine. The clicked documents have
  topics associated with them. You will be given a query and the
  associated topics. The query will mention one or more named
  entities. You need to generate 10 similar queries that mention a
  different named entity of the same type as the original query,
  that would still result in clicks on documents of with the same
  topics as the original query. The new query should be
  stylistically similar to the original query. Respond with just
  the queries, with each query in a new line. Do not number the
  queries. The query is "{query}" and the topic is {topic}. Please
  rewrite the query with another {type of detected entity}."}]
```

A.2 Prompts for In-Context Learning

- TripClick

```
[{"role": "system",
  "content": "You are a web search assistant. Your job is to help
  understand which topics the user is interested in so that the
  most relevant documents can be shown to the user. "},
 {"role": "user",
  "content": "Your task is to classify web search queries into a set
  of \"topics\". The queries are from the TripClick dataset, which
  contains click data from a web search engine. The clicked
  documents have topics associated with them, and the queries are
  classified into the same topics. Multiple topics can be
  associated with a query. Respond with just the topics separated
  by commas. \n\nHere is the full list of topics:\n {List of all
  topics in the dataset}\n\nDO NOT respond with topics that are not
  in the list. Here are some examples. The format is <query> | <
  topics>.\n\n {List of examples, each in a new line.}"]]
```

- Private Health Search

```
[{"role": "system",
  "content": "You are a web search assistant. Your job is to help
  understand which topics the user is interested in so that the
  most relevant documents can be shown to the user. "},
```

```
{"role": "user",  
  "content": "Your task is to classify web search queries into a set  
of \"intents\". The queries are from a health insurance website.  
Users typed these queries looking for different types of  
information. They clicked URLs that were relevant to their  
queries. The URLs have intents associated with them, and the  
queries are classified into the same intents. Multiple intents  
can be associated with a query. Respond with just the intents  
separated by commas. \n\nHere is the full list of intents:\n {  
List of all intents in the dataset}\n\nDO NOT respond with  
intents that are not in the list. Here are some examples. The  
format is <query> | <intents>.\n\n{List of examples, each in a  
new line.}"}]
```

Appendix B

Additional Results

Table B.1: Classwise results for intent classification on TripClick dataset. Bold classes are those augmented

label	F1	Class Frequency
dentistry	0.8934	799
ophthalmology	0.8654	408
psychiatry	0.8585	2310
cardiology	0.8576	3896
pulmonology	0.8576	2216
urology	0.8559	1227
endocrinology	0.8536	2132
obgyn	0.8400	3828
oncology	0.8375	2396
neurological surgery	0.8311	2292
rheumatology	0.8247	1766
geriatrics	0.8209	1313
gastroenterology	0.8168	2475
infectious disease	0.8123	3447
dermatology	0.7955	953
orthopaedics	0.7720	1288
otolaryngology-ent	0.7565	1054
pediatrics	0.7335	4244
critical care	0.6919	672
allergies and immunology	0.6762	369
hematology	0.6723	1111
anesthesiology	0.6566	727
surgery	0.6401	1974
radiology	0.6157	413
emergency medicine	0.6035	659
physical medicine	0.5293	524
pain_management/palliative_medicine	0.4000	4
hospice and palliative care	0.3259	58
womens health	0.2704	1272
primary care	0.2010	272

Bibliography

- [1] ACHIAM, J., ADLER, S., AGARWAL, S., AHMAD, L., AKKAYA, I., ALEMAN, F. L., ALMEIDA, D., ALTENSCHMIDT, J., ALTMAN, S., ANADKAT, S., ET AL. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [2] AGARWAL, R., SINGH, A., ZHANG, L. M., BOHNET, B., CHAN, S., ANAND, A., ABBAS, Z., NOVA, A., CO-REYES, J. D., CHU, E., ET AL. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018* (2024).
- [3] AHMADVAND, A., CHOI, I., SAHIJWANI, H., SCHMIDT, J., SUN, M., VOLOKHIN, S., WANG, Z., AND AGICHTTEIN, E. Emory irisbot: An open-domain conversational bot for personalized information access. *Alexa Prize Proceedings* (2018).
- [4] AHMADVAND, A., CHOI, I. J., SAHIJWANI, H., SCHMIDT, J., SUN, M., VOLOKHIN, S., WANG, Z., AND AGICHTTEIN, E. Emory irisbot: An open-domain conversational bot for personalized information access. *Alexa Prize Proceedings* (2018).
- [5] AHMADVAND, A., SAHIJWANI, H., CHOI, J. I., AND AGICHTTEIN, E. Concet: Entity-aware topic classification for open-domain conversational agents. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019), pp. 1371–1380.
- [6] ALIANNEJADI, M., AZZOPARDI, L., ZAMANI, H., KANOULAS, E., THOMAS, P., AND CRASWELL, N. Analysing mixed initiatives and search strategies during conversational search. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), pp. 16–26.
- [7] ALIANNEJADI, M., KISELEVA, J., CHUKLIN, A., DALTON, J., AND BURTSEV, M. Building and evaluating open-domain dialogue corpora with clarifying questions. *arXiv preprint arXiv:2109.05794* (2021).
- [8] ANDREA, M., RAGANATO, A., AND NAVIGLI, R. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics* 2 (2014), 231–244.
- [9] ANDRIY, M., AND SALAKHUTDINOV, R. R. Probabilistic matrix factorization. In *Advances in neural information processing systems* (2008), pp. 1257–1264.
- [10] ANNERVAZ, K., CHOWDHURY, S. B. R., AND DUKKIPATI, A. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. *arXiv preprint arXiv:1802.05930* (2018).

- [11] AUER, S., BIZER, C., KOBILAROV, G., LEHMANN, J., CYGANIAK, R., AND IVES, Z. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 2007, pp. 722–735.
- [12] BELTAGY, I., LO, K., AND COHAN, A. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676* (2019).
- [13] BENGIO, Y., DUCHARME, R., AND VINCENT, P. A neural probabilistic language model. *Advances in neural information processing systems 13* (2000).
- [14] BODENREIDER, O. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research 32*, suppl_1 (2004), D267–D270.
- [15] BOJANOWSKI, P., GRAVE, E., JOULIN, A., AND MIKOLOV, T. Enriching word vectors with subword information. 135–146.
- [16] BORDES, A., USUNIER, N., GARCIA-DURAN, A., WESTON, J., AND YAKHNENKO, O. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems 26* (2013).
- [17] BORGEAUD, S., MENSCH, A., HOFFMANN, J., CAI, T., RUTHERFORD, E., MILLICAN, K., VAN DEN DRIESSCHE, G. B., LESPIAU, J.-B., DAMOC, B., CLARK, A., ET AL. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning* (2022), PMLR, pp. 2206–2240.
- [18] BOUMA, G. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL* (2009), 31–40.
- [19] BRODER, A. Z., FONTOURA, M., GABRILOVICH, E., JOSHI, A., JOSIFOVSKI, V., AND ZHANG, T. Robust classification of rare queries using web knowledge. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2007), SIGIR '07, Association for Computing Machinery, p. 231–238.
- [20] BRYMAN, A. *Social research methods*. Oxford university press, 2016.
- [21] CER, D., YANG, Y., KONG, S.-Y., HUA, N., LIMTIACO, N., JOHN, R. S., CONSTANT, N., GUAJARDO-CESPEDES, M., YUAN, S., TAR, C., ET AL. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).
- [22] CHEN, J., NAIRN, R., NELSON, L., BERNSTEIN, M., AND CHI, E. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the SIGCHI conference on human factors in computing systems* (2010), pp. 1185–1194.
- [23] CHEN, Q., ZHUO, Z., AND WANG, W. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909* (2019).
- [24] CHIU, J. P., AND NICHOLS, E. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics 4* (2016), 357–370.
- [25] CHOI, J. I., AHMADVAND, A., AND AGICHTEIN, E. Offline and online satisfaction prediction in open-domain conversational systems. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019), pp. 1281–1290.

- [26] CHOI, Y., MONSERRAT, T.-J. K. P., PARK, J., SHIN, H., LEE, N., AND KIM, J. Protochat: Supporting the conversation design process with crowd feedback. *Proceedings of the ACM on Human-Computer Interaction 4*, CSCW3 (2021), 1–27.
- [27] CHRISTAKOPOULOU, K., RADLINSKI, F., AND HOFMANN, K. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (2016), ACM, pp. 815–824.
- [28] CHU, Z., WANG, Z., LIU, Y., HUANG, Y., ZHANG, M., AND MA, S. Convsearch: A open-domain conversational search behavior dataset. *arXiv preprint arXiv:2204.02659* (2022).
- [29] CONNEAU, A., SCHWENK, H., BARRAULT, L., AND LECUN, Y. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781* (2016).
- [30] CORNOLTI, M., FERRAGINA, P., CIARAMITA, M., RÜD, S., AND SCHÜTZE, H. A piggyback system for joint entity mention detection and linking in web queries. In *Proceedings of the 25th International Conference on World Wide Web* (2016), pp. 567–578.
- [31] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, Minnesota, June 2019), Association for Computational Linguistics, pp. 4171–4186.
- [32] DUBIEL, M., BARGHOUTI, Y., KUDRYAVTSEVA, K., AND LEIVA, L. A. On-device query intent prediction with lightweight llms to support ubiquitous conversations. *Scientific Reports 14*, 1 (2024), 12731.
- [33] EMELIN, D., BONADIMAN, D., ALQAHTANI, S., ZHANG, Y., AND MANSOUR, S. Injecting domain knowledge in language models for task-oriented dialogue systems. *arXiv preprint arXiv:2212.08120* (2022).
- [34] FLEISS, J. L. Measuring nominal scale agreement among many raters. *Psychological bulletin 76*, 5 (1971), 378.
- [35] GABRILOVICH, E., BRODER, A., FONTOURA, M., JOSHI, A., JOSIFOVSKI, V., RIEDEL, L., AND ZHANG, T. Classifying search queries using the web as a source of knowledge. *ACM Transactions on the Web (TWEB) 3*, 2 (2009), 1–28.
- [36] GAO, T., HAN, X., QIU, K., BAI, Y., XIE, Z., LIN, Y., LIU, Z., LI, P., SUN, M., AND ZHOU, J. Manual evaluation matters: Reviewing test protocols of distantly supervised relation extraction. *arXiv preprint arXiv:2105.09543* (2021).
- [37] GAO, T., YAO, X., AND CHEN, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).
- [38] GUO, F., METALLINO, A., KHATRI, C., RAJU, A., VENKATESH, A., AND RAM, A. Topic-based evaluation for conversational bots. In *NIPS* (2018).

- [39] HASHEMI, H., ZAMANI, H., AND CROFT, W. B. Guided transformer: Leveraging multiple external sources for representation learning in conversational search. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval* (2020), pp. 1131–1140.
- [40] HASHEMI, H., ZAMANI, H., AND CROFT, W. B. Learning multiple intent representations for search queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), pp. 669–679.
- [41] HE, X. O. Mixture of a million experts. *arXiv preprint arXiv:2407.04153* (2024).
- [42] HU, J., WANG, G., LOCHOVSKY, F., SUN, J.-T., AND CHEN, Z. Understanding user’s query intent with wikipedia. In *Proceedings of the 18th International Conference on World Wide Web* (New York, NY, USA, 2009), WWW ’09, Association for Computing Machinery, p. 471–480.
- [43] HUANG, H.-Y., CHOI, E., AND YIH, W.-T. Flowqa: Grasping flow in history for conversational machine comprehension. *arXiv preprint arXiv:1810.06683* (2018).
- [44] HUANG, Q., WAKE, N., SARKAR, B., DURANTE, Z., GONG, R., TAORI, R., NODA, Y., TERZOPOULOS, D., KUNO, N., FAMOTI, A., ET AL. Position paper: Agent ai towards a holistic intelligence. *arXiv preprint arXiv:2403.00833* (2024).
- [45] JIANG, A. Q., SABLAYROLLES, A., ROUX, A., MENSCH, A., SAVARY, B., BAMFORD, C., CHAPLOT, D. S., CASAS, D. D. L., HANNA, E. B., BRESSAND, F., ET AL. Mixtral of experts. *arXiv preprint arXiv:2401.04088* (2024).
- [46] JIANG, Z., XU, F., GAO, L., SUN, Z., LIU, Q., DWIVEDI-YU, J., YANG, Y., CALLAN, J., AND NEUBIG, G. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (Singapore, Dec. 2023), H. Bouamor, J. Pino, and K. Bali, Eds., Association for Computational Linguistics, pp. 7969–7992.
- [47] JOTY, S., MOHIUDDIN, M. T., AND NGUYEN., D. T. Coherence modeling of asynchronous conversations: A neural entity grid approach. In *proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (2018), pp. 558–568.
- [48] KALCHBRENNER, N., AND BLUNSOM, P. Recurrent convolutional neural networks for discourse compositionality. *arXiv preprint arXiv:1306.3584* (2013).
- [49] KEYVAN, K., AND HUANG, J. X. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Computing Surveys* 55, 6 (2022), 1–40.
- [50] KHATRI, C., GOEL, R., HEDAYATNIA, B., METANILLOU, A., VENKATESH, A., GABRIEL, R., AND MANDAL, A. Contextual topic modeling for dialog systems. In *IEEE 2018 Spoken Language Technology (SLT)* (2018), IEEE.
- [51] KHATRI, C., HEDAYATNIA, B., VENKATESH, A., NUNN, J., PAN, Y., LIU, Q., SONG, H., GOTTARDI, A., KWATRA, S., PANCHOLI, S., ET AL. Advancing the state of the art in open domain dialog systems through the alexa prize. *arXiv preprint arXiv:1812.10757* (2018).

- [52] KIM, J.-K., WANG, G., LEE, S., AND KIM, Y.-B. Deciding whether to ask clarifying questions in large-scale spoken language understanding. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (2021), IEEE, pp. 869–876.
- [53] KIM, S., JANG, J. Y., JUNG, M., AND SHIN, S. A model of cross-lingual knowledge-grounded response generation for open-domain dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2021* (2021), pp. 352–365.
- [54] KIM, Y. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014).
- [55] KOCIELNIK, R., XIAO, L., AVRAHAMI, D., AND HSIEH, G. Reflection companion: a conversational system for engaging users in reflection on physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–26.
- [56] KOREN, Y., BELL, R., AND VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer*, 8 (2009), 30–37.
- [57] KRAUSE, B., DAMONTE, M., DOBRE, M., DUMA, D., FAINBERG, J., FANCELLO, F., KAHEMBWE, E., CHENG, J., AND WEBBER., B. Edina: Building an open domain socialbot with self-dialogues. In *1st Proceedings of the Alexa Prize* (2017), Amazon.
- [58] LAFFERTY, J., MCCALLUM, A., AND PEREIRA, F. C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- [59] LANDIS, J. R., AND KOCH, G. G. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.
- [60] LEE, J., YOON, W., KIM, S., KIM, D., KIM, S., SO, C. H., AND KANG, J. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.
- [61] LEE, J. Y., AND DERNONCOURT, F. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827* (2016).
- [62] LEE, Y.-C., YAMASHITA, N., AND HUANG, Y. Designing a chatbot as a mediator for promoting deep self-disclosure to a real mental health professional. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–27.
- [63] LEI, W., ZHANG, G., HE, X., MIAO, Y., WANG, X., CHEN, L., AND CHUA, T.-S. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 2073–2083.
- [64] LEWIS, P., PEREZ, E., PIKTUS, A., PETRONI, F., KARPUKHIN, V., GOYAL, N., KÜTTLER, H., LEWIS, M., YIH, W.-T., ROCKTÄSCHEL, T., ET AL. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [65] LI, G., ZHAI, Y., CHEN, Q., GAO, X., ZHANG, J., AND ZHANG, Y. Continual few-shot intent detection. In *Proceedings of the 29th International Conference on Computational Linguistics* (2022), pp. 333–343.

- [66] LI, L., CHU, W., LANGFORD, J., AND SCHAPIRE, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web* (2010), pp. 661–670.
- [67] LI, R., KAHOU, S. E., SCHULZ, H., MICHALSKI, V., CHARLIN, L., AND PAL, C. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems* (2018), pp. 9748–9758.
- [68] LI, S., LEI, W., WU, Q., HE, X., JIANG, P., AND CHUA, T.-S. Seamlessly unifying attributes and items: Conversational recommendation for cold-start users. *ACM Transactions on Information Systems (TOIS)* 39, 4 (2021), 1–29.
- [69] LIN, B. Y., CHEN, X., CHEN, J., AND REN, X. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151* (2019).
- [70] LIN, T.-E., WU, Y., HUANG, F., SI, L., SUN, J., AND LI, Y. Duplex conversation: Towards human-like interaction in spoken dialogue systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), pp. 3299–3308.
- [71] LIN, X. V., CHEN, X., CHEN, M., SHI, W., LOMELI, M., JAMES, R., RODRIGUEZ, P., KAHN, J., SZILVASY, G., LEWIS, M., ET AL. Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352* (2023).
- [72] LIU, Q., HAO, Y., LIU, X., LI, B., SUI, D., HE, S., LIU, K., ZHAO, J., CHEN, X., ZHANG, N., ET AL. Class lifelong learning for intent detection via structure consolidation networks. In *Findings of the Association for Computational Linguistics: ACL 2023* (2023), pp. 293–306.
- [73] LIU, Q., YU, X., HE, S., LIU, K., AND ZHAO, J. Lifelong intent detection via multi-strategy rebalancing. *arXiv preprint arXiv:2108.04445* (2021).
- [74] LIU, W., TANG, J., LIANG, X., AND CAI, Q. Heterogeneous graph reasoning for knowledge-grounded medical dialogue system. *Neurocomputing* 442 (2021), 260–268.
- [75] LOPS, P., DE GEMMIS, M., AND SEMERARO, G. Content-based recommender systems: State of the art and trends. *Recommender systems handbook* (2011), 73–105.
- [76] LU, Z., DOU, Z., LIAN, J., XIE, X., AND YANG, Q. Content-based collaborative filtering for news topic recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2015), vol. 29.
- [77] LUO, S.-B., KUO, C.-C., AND CHEN, K.-Y. Spoken multiple-choice question answering using multi-turn audio-extractor bert. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (2020), IEEE, pp. 386–392.
- [78] MAO, K., DOU, Z., CHEN, H., MO, F., AND QIAN, H. Large language models know your contextual search intent: A prompting framework for conversational search. *arXiv preprint arXiv:2303.06573* (2023).
- [79] MAO, K., QIAN, H., MO, F., DOU, Z., LIU, B., CHENG, X., AND CAO, Z. Learning denoised and interpretable session representation for conversational search. In *Proceedings of the ACM Web Conference 2023* (2023), pp. 3193–3202.

- [80] MECKLENBURG, N., LIN, Y., LI, X., HOLSTEIN, D., NUNES, L., MALVAR, S., SILVA, B., CHANDRA, R., ASKI, V., YANNAM, P. K. R., ET AL. Injecting new knowledge into large language models via supervised fine-tuning. *arXiv preprint arXiv:2404.00213* (2024).
- [81] MEHRI, S., RAZUMOVSKAIA, E., ZHAO, T., AND ESKENAZI, M. Pretraining methods for dialog context representation learning. *arXiv preprint arXiv:1906.00414* (2019).
- [82] MENDES, P. N., JAKOB, M., GARCÍA-SILVA, A., AND BIZER, C. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems* (2011), pp. 1–8.
- [83] MESGAR, M., TRAN, T. T., GLAVAS, G., AND GUREVYCH, I. The devil is in the details: On models and training regimes for few-shot intent classification. *arXiv preprint arXiv:2210.06440* (2022).
- [84] MI, F., WANG, Y., AND LI, Y. Cins: Comprehensive instruction for few-shot learning in task-oriented dialog systems. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2022), vol. 36, pp. 11076–11084.
- [85] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781* (2013).
- [86] MITCHELL, E., ELHADAD, N., AND MAMYKINA, L. Examining ai methods for micro-coaching dialogs. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (2022), pp. 1–24.
- [87] MITCHELL, T. M. *Machine learning*. No. 35. Burr Ridge, IL: McGraw Hill 45, 1997.
- [88] MO, F., MAO, K., ZHU, Y., WU, Y., HUANG, K., AND NIE, J.-Y. Convgqr: Generative query reformulation for conversational search. *arXiv preprint arXiv:2305.15645* (2023).
- [89] MOHAMMAD, A. Convai3: Generating clarifying questions for open-domain dialogue systems (clariq). *arXiv: 2009.11352* (2020).
- [90] MOHIUDDIN, T., JOTY, S., AND NGUYEN, D. T. Coherence modeling of asynchronous conversations: A neural entity grid approach. *arXiv preprint arXiv:1805.02275* (2018).
- [91] NEUMANN, M., KING, D., BELTAGY, I., AND AMMAR, W. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task* (Florence, Italy, Aug. 2019), D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds., Association for Computational Linguistics, pp. 319–327.
- [92] OHSUGI, Y., SAITO, I., NISHIDA, K., ASANO, H., AND TOMITA, J. A simple but effective method to incorporate multi-turn context with bert for conversational machine comprehension. *arXiv preprint arXiv:1905.12848* (2019).
- [93] ORTEGA, D., AND VU, N. T. Neural-based context representation learning for dialog act classification. *arXiv preprint arXiv:1708.02561* (2017).
- [94] OVADIA, O., BRIEF, M., MISHAELI, M., AND ELISHA, O. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934* (2023).

- [95] OWOICHO, P., SEKULIĆ, I., ALIANNEJADI, M., DALTON, J., AND CRESTANI, F. Exploiting simulated user feedback for conversational search: Ranking, rewriting, and beyond. *arXiv preprint arXiv:2304.13874* (2023).
- [96] PARK, H., SON, J., MIN, J., AND CHOI, J. Selective umls knowledge infusion for biomedical question answering. *Scientific Reports* 13, 1 (2023), 14214.
- [97] PHELAN, O., MCCARTHY, K., AND SMYTH, B. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems* (2009), pp. 385–388.
- [98] POERNER, N., WALTINGER, U., AND SCHÜTZE, H. E-bert: Efficient-yet-effective entity embeddings for bert. *arXiv preprint arXiv:1911.03681* (2019).
- [99] POST, M., AND BERGSMA, S. Explicit and implicit syntactic features for text classification. In *proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (2013), pp. 866–872.
- [100] QIN, L., LIU, T., CHE, W., KANG, B., ZHAO, S., AND LIU, T. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 8193–8197.
- [101] QIU, Z., TAO, Y., PAN, S., AND LIEW, A. W.-C. Knowledge graphs and pretrained language models enhanced representation learning for conversational recommender systems. *IEEE Transactions on Neural Networks and Learning Systems* (2024).
- [102] QU, C., YANG, L., CROFT, W. B., TRIPPAS, J. R., ZHANG, Y., AND QIU, M. Analyzing and characterizing user intent in information-seeking conversations. In *The 41st international acm sigir conference on research & development in information retrieval* (2018), pp. 989–992.
- [103] RAM, A., PRASAD, R., KHATRI, C., VENKATESH, A., GABRIEL, R., LIU, Q., NUNN, J., HEDAYATNIA, B., CHENG, M., NAGAR, A., ET AL. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604* (2018).
- [104] REKABSAZ, N., LESOTA, O., SCHEDL, M., BRASSEY, J., AND EICKHOFF, C. Tripclick: the log files of a large health web search engine. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021), pp. 2507–2513.
- [105] SAHIJWANI, H., CHOI, J. I., AND AGICHTTEIN, E. Would you like to hear the news? investigating voice-based suggestions for conversational news recommendation. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval (CHIIR'20)* (2020).
- [106] SAHIJWANI, H., DHOLE, K., PURWAR, A., VASUDEVAN, V., AND AGICHTTEIN, E. Contextual response interpretation for automated structured interviews: A case study in market research. In *Companion Proceedings of the ACM Web Conference 2023* (New York, NY, USA, 2023), WWW '23 Companion, Association for Computing Machinery, p. 886–891.
- [107] SALLE, A., MALMASI, S., ROKHLENKO, O., AND AGICHTTEIN, E. Cosearcher: studying the effectiveness of conversational search refinement and clarification through user simulation. *Information Retrieval Journal* 25, 2 (2022), 209–238.

- [108] SAMARINAS, C., DHARAWAT, A., AND ZAMANI, H. Revisiting open domain query facet extraction and generation. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval* (2022), pp. 43–50.
- [109] SAPPELLI, M., VERBERNE, S., HEIJDEN, M. V. D., HINNE, M., AND KRAAIJ, W. Collection and analysis of ground truth data for query intent.
- [110] SCHICK, T., DWIVEDI-YU, J., DESSÌ, R., RAILEANU, R., LOMELI, M., HAMBRO, E., ZETTLEMOYER, L., CANCEDDA, N., AND SCIALOM, T. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems 36* (2024).
- [111] SCHUURMANS, J., AND FRASINCAR, F. Intent classification for dialogue utterances. *IEEE Intelligent Systems 35*, 1 (2019), 82–88.
- [112] SEKULIĆ, I., ALIANNEJADI, M., AND CRESTANI, F. User engagement prediction for clarification in search. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43* (2021), Springer, pp. 619–633.
- [113] SELTZER, J., CHENG, K., SUN, Y., KOLAGATI, S., LIN, J., ZONG, S., ET AL. *smartprobe*: A virtual moderator for market research surveys. *arXiv preprint arXiv:2305.08271* (2023).
- [114] SONG, S., CHEN, X., WANG, C., YU, X., WANG, J., AND HE, X. A two-stage user intent detection model on complicated utterances with multi-task learning. In *Companion Proceedings of the Web Conference 2022* (2022), pp. 197–200.
- [115] SRINIVASAN, K., RAMAN, K., SAMANTA, A., LIAO, L., BERTELLI, L., AND BENDERSKY, M. Quill: Query intent with large language models using retrieval augmentation and multi-stage distillation. *arXiv preprint arXiv:2210.15718* (2022).
- [116] SUN, W., YAN, L., MA, X., REN, P., YIN, D., AND REN, Z. Is chatgpt good at search? investigating large language models as re-ranking agent. *ArXiv abs/2304.09542* (2023).
- [117] SUN, Y., AND ZHANG, Y. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval* (2018), pp. 235–244.
- [118] TAVAKOLI, L., ZAMANI, H., SCHOLER, F., CROFT, W. B., AND SANDERSON, M. Analyzing clarification in asynchronous information-seeking conversations. *Journal of the Association for Information Science and Technology 73*, 3 (2022), 449–471.
- [119] TOM AYOOLA, JOSEPH FISHER, A. P. Improving entity disambiguation by reasoning over a knowledge base. In *NAACL* (2022).
- [120] TOUVRON, H., MARTIN, L., STONE, K., ALBERT, P., ALMAHAIRI, A., BABAEI, Y., BASHLYKOV, N., BATRA, S., BHARGAVA, P., BHOSALE, S., ET AL. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
- [121] VARSHNEY, V., PATIDAR, M., KUMAR, R., VIG, L., AND SHROFF, G. Prompt augmented generative replay via supervised contrastive learning for lifelong intent detection. In *Findings of the Association for Computational Linguistics: NAACL 2022* (2022), pp. 1113–1127.

- [122] VÄTH, D., VANDERLYN, L., AND VU, N. T. Conversational tree search: A new hybrid dialog task. *arXiv preprint arXiv:2303.10227* (2023).
- [123] VENKATESH, A., KHATRI, C., RAM, A., GUO, F., GABRIEL, R., NAGAR, A., PRASAD, R., CHENG, M., HEDAYATNIA, B., METALLINO, A., ET AL. On evaluating and comparing conversational agents.
- [124] VRANDEČIĆ, D., AND KRÖTZSCH, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM* 57, 10 (2014), 78–85.
- [125] WANG, C., PAN, H., LIU, Y., CHEN, K., QIU, M., ZHOU, W., HUANG, J., CHEN, H., LIN, W., AND CAI, D. Mell: Large-scale extensible user intent classification for dialogue systems with meta lifelong learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2021), pp. 3649–3659.
- [126] WANG, H., ZHANG, F., XIE, X., AND GUO, M. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference* (2018), pp. 1835–1844.
- [127] WANG, J., AND LI, W. Template-guided clarifying question generation for web search clarification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), pp. 3468–3472.
- [128] WANG, J., WANG, Z., ZHANG, D., AND YAN., J. Combining knowledge with deep convolutional neural networks for short text classification. In *IJCAI* (2017), pp. 2915–2921.
- [129] WANG, R., TANG, D., DUAN, N., WEI, Z., HUANG, X., CAO, G., JIANG, D., ZHOU, M., ET AL. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808* (2020).
- [130] WANG, X., GAO, T., ZHU, Z., ZHANG, Z., LIU, Z., LI, J., AND TANG, J. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics* 9 (2021), 176–194.
- [131] WANG, Y., REDDY, R. G., MUJAHID, Z. M., ARORA, A., RUBASHEVSKII, A., GENG, J., AFZAL, O. M., PAN, L., BORENSTEIN, N., PILLAI, A., AUGENSTEIN, I., GUREVYCH, I., AND NAKOV, P. Factcheck-gpt: End-to-end fine-grained document-level fact-checking and correction of llm output. *ArXiv abs/2311.09000* (2023).
- [132] WANG, Z., AND AI, Q. Controlling the risk of conversational search via reinforcement learning. In *Proceedings of the web conference 2021* (2021), pp. 1968–1977.
- [133] WANG, Z., TU, Y., ROSSET, C., CRASWELL, N., WU, M., AND AI, Q. Zero-shot clarifying question generation for conversational search. In *Proceedings of the ACM Web Conference 2023* (2023), pp. 3288–3298.
- [134] WANG, Z., AND WANG., H. Understanding short texts(tutorial). In ACL.
- [135] WANG, Z., XU, Z., AND AI, Q. Reward-free policy imitation learning for conversational search. *arXiv preprint arXiv:2304.07988* (2023).

- [136] WEI, J., KIM, S., JUNG, H., AND KIM, Y.-H. Leveraging large language models to power chatbots for collecting user self-reported data. *arXiv preprint arXiv:2301.05843* (2023).
- [137] WEISCHEDEL, R., PALMER, M., MARCUS, M., HOVY, E., PRADHAN, S., RAMSHAW, L., XUE, N., ET AL. "ontonotes release 5.0". In *Linguistic Data Consortium* (2013), Philadelphia, PA.
- [138] WELD, H., HUANG, X., LONG, S., POON, J., AND HAN, S. C. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Computing Surveys (CSUR)* (2021).
- [139] WEN, L., WANG, X., DONG, Z., AND CHEN, H. Jointly modeling intent identification and slot filling with contextual and hierarchical information. In *National CCF Conference on Natural Language Processing and Chinese Computing* (2017), Springer, pp. 3–15.
- [140] WU, C.-S., HOI, S. C., SOCHER, R., AND XIONG, C. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 917–929.
- [141] WU, Q., BANSAL, G., ZHANG, J., WU, Y., ZHANG, S., ZHU, E., LI, B., JIANG, L., ZHANG, X., AND WANG, C. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155* (2023).
- [142] XIAO, L., MA, J., DONG, X. L., MARTINEZ-GOMEZ, P., ZALMOUT, N., CHEN, W., ZHAO, T., HE, H., AND JIN, Y. End-to-end conversational search for online shopping with utterance transfer. *arXiv preprint arXiv:2109.05460* (2021).
- [143] XIONG, C., CALLAN, J., AND LIU., T.-Y. Bag-of-entities representation for ranking. ACM, pp. 181–184.
- [144] YAMADA, I., SHINDO, H., TAKEDA, H., AND TAKEFUJI., Y. Learning distributed representations of texts and entities from knowledge base. MIT Press, pp. 397–411.
- [145] YAN, R., ZHAO, D., AND E, W. Joint learning of response ranking and next utterance suggestion in human-computer conversation system. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (2017), pp. 685–694.
- [146] YASUNAGA, M., REN, H., BOSSELUT, A., LIANG, P., AND LESKOVEC, J. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Online, June 2021), Association for Computational Linguistics, pp. 535–546.
- [147] YE, D., LIN, Y., LI, P., SUN, M., AND LIU, Z. A simple but effective pluggable entity lookup table for pre-trained language models. *arXiv preprint arXiv:2202.13392* (2022).
- [148] YE, Z.-X., AND LING, Z.-H. Hybrid semi-markov crf for neural sequence labeling. *arXiv preprint arXiv:1805.03838* (2018).

- [149] ZAMANI, H., LUECK, G., CHEN, E., QUISPE, R., LUU, F., AND CRASWELL, N. Mimics: A large-scale data collection for search clarification. In *Proceedings of the 29th ACM international conference on information & knowledge management* (2020), pp. 3189–3196.
- [150] ZHANG, W., CHEN, J., WU, H., WAN, S., AND LI, G. A knowledge-grounded dialog system based on pre-trained language models. *arXiv preprint arXiv:2106.14444* (2021).
- [151] ZHANG, X., AND LECUN, Y. Text understanding from scratch. *arXiv preprint arXiv:1502.01710* (2015).
- [152] ZHANG, Y., CHEN, X., AI, Q., YANG, L., AND CROFT, W. B. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management* (2018), pp. 177–186.
- [153] ZHANG, Y., SUN, S., GALLEY, M., CHEN, Y.-C., BROCKETT, C., GAO, X., GAO, J., LIU, J., AND DOLAN, B. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (Online, July 2020), Association for Computational Linguistics, pp. 270–278.
- [154] ZHANG, Z., ZENG, Z., LIN, Y., WANG, H., YE, D., XIAO, C., HAN, X., LIU, Z., LI, P., SUN, M., AND ZHOU, J. Plug-and-play knowledge injection for pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Toronto, Canada, July 2023), A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Association for Computational Linguistics, pp. 10641–10658.
- [155] ZHAO, C., YU, T., XIE, Z., AND LI, S. Knowledge-aware conversational preference elicitation with bandit feedback. In *Proceedings of the ACM Web Conference 2022* (2022), pp. 483–492.
- [156] ZHAO, Y., ZHENG, Y., TIAN, Z., GAO, C., YU, B., YU, H., LI, Y., SUN, J., AND ZHANG, N. L. Prompt conditioned vae: Enhancing generative replay for lifelong learning in task-oriented dialogue. *arXiv preprint arXiv:2210.07783* (2022).
- [157] ZHAO, Z., DOU, Z., MAO, J., AND WEN, J.-R. Generating clarifying questions with web search results. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2022), pp. 234–244.
- [158] ZHOU, Y., DI EUGENIO, B., ZIEBART, B., SHARP, L., LIU, B., GERBER, B., AGADAKOS, N., AND YADAV, S. Towards enhancing health coaching dialogue in low-resource settings. In *Proceedings of the 29th International Conference on Computational Linguistics* (2022), pp. 694–706.
- [159] ZORRILLA, A. L., AND TORRES, M. I. A multilingual neural coaching model with enhanced long-term dialogue structure. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 12, 2 (2022), 1–47.
- [160] ZOU, J., CHEN, Y., AND KANOULAS, E. Towards question-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020), pp. 881–890.

- [161] ZOU, J., HUANG, J., REN, Z., AND KANOULAS, E. Learning to ask: Conversational product search via representation learning. *ACM Transactions on Information Systems* 41, 2 (2022), 1–27.
- [162] ZOU, J., SUN, A., LONG, C., ALIANNEJADI, M., AND KANOULAS, E. Asking clarifying questions: To benefit or to disturb users in web search? *Information Processing & Management* 60, 2 (2023), 103176.