Université de Montréal

Context-Aware Ranking: From Search to Dialogue

par

Yutao Zhu

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en Discipline

March 21, 2023

Université de Montréal

Faculté des arts et des sciences

Cette thèse intitulée

Context-Aware Ranking: From Search to Dialogue

présentée par

Yutao Zhu

a été évaluée par un jury composé des personnes suivantes :

Bang Liu
(président-rapporteur)

Jian-Yun Nie
(directeur de recherche)

Philippe Langlais
(membre du jury)

Xiaodong He
(examinateur externe)

Marie-Claude L'Homme
(représentant du doyen de la FESP)

Résumé

Les systèmes de recherche d'information (RI) ou moteurs de recherche ont été largement utilisés pour trouver rapidement les informations pour les utilisateurs. Le classement est la fonction centrale de la RI, qui vise à ordonner les documents candidats dans une liste classée en fonction de leur pertinence par rapport à une requête de l'utilisateur. Alors que IR n'a considéré qu'une seule requête au début, les systèmes plus récents prennent en compte les informations de contexte. Par exemple, dans une session de recherche, le contexte de recherche tel que le requêtes et interactions précédentes avec l'utilisateur, est largement utilisé pour comprendre l'intention de la recherche de l'utilisateur et pour aider au classement des documents. En plus de la recherche ad-hoc traditionnelle, la RI a été étendue aux systèmes de dialogue (c'est-à-dire, le dialogue basé sur la recherche, par exemple, XiaoIce), où on suppose avoir un grand référentiel de dialogues et le but est de trouver la réponse pertinente à l'énoncé courant d'un utilisateur. Encore une fois, le contexte du dialogue est un élément clé pour déterminer la pertinence d'une réponse.

L'utilisation des informations contextuelles a fait l'objet de nombreuses études, allant de l'extraction de mots-clés importants du contexte pour étendre la requête ou l'énoncé courant de dialogue, à la construction d'une représentation neuronale du contexte qui sera utilisée avec la requête ou l'énoncé de dialogue pour la recherche. Nous remarquons deux d'importantes insuffisances dans la littérature existante. (1) Pour apprendre à utiliser les informations contextuelles, on doit extraire des échantillons positifs et négatifs pour l'entraînement. On a généralement supposé qu'un échantillon positif est formé lorsqu'un utilisateur interagit avec (clique sur) un document dans un contexte, et un un échantillon négatif est formé lorsqu'aucune interaction n'est observée. En réalité, les interactions des utilisateurs sont éparses et bruitées, ce qui rend l'hypothèse ci-dessus irréaliste. Il est donc important de construire des exemples d'entraînement d'une manière plus appropriée. (2) Dans les systèmes de dialogue, en particulier les systèmes de bavardage (chitchat), on cherche à trouver ou générer les réponses sans faire référence à des connaissances externes, ce qui peut facilement provoquer des réponses non pertinentes ou des hallucinations. Une solution consiste à fonder le dialogue sur des documents ou graphe de connaissances externes, où les documents ou les graphes de connaissances peuvent être considérés comme de nouveaux types de contexte.

Le dialogue fondé sur les documents et les connaissances a été largement étudié, mais les approches restent simplistes dans la mesure où le contenu du document ou les connaissances sont généralement concaténés à l'énoncé courant. En réalité, seules certaines parties du document ou du graphe de connaissances sont pertinentes, ce qui justifie un modèle spécifique pour leur sélection.

Dans cette thèse, nous étudions le problème du classement de textes en tenant compte du contexte dans le cadre de RI ad-hoc et de dialogue basé sur la recherche. Nous nous concentrons sur les deux problèmes mentionnés ci-dessus. Spécifiquement, nous proposons des approches pour apprendre un modèle de classement pour la RI ad-hoc basée sur des exemples d'entraîenemt sélectionnés à partir d'interactions utilisateur bruitées (c'est-à-dire des logs de requêtes) et des approches à exploiter des connaissances externes pour la recherche de réponse pour le dialogue. La thèse est basée sur cinq articles publiés. Les deux premiers articles portent sur le classement contextuel des documents. Ils traitent le problème ovservé dans les études existantes, qui considèrent tous les clics dans les logs de recherche comme des échantillons positifs, et prélever des documents non cliqués comme échantillons négatifs. Dans ces deux articles, nous proposons d'abord une stratégie d'augmentation de données non supervisée pour simuler les variations potentielles du comportement de l'utilisateur pour tenir compte de la sparcité des comportements des utilisateurs. Ensuite, nous appliquons l'apprentissage contrastif pour identifier ces variations et à générer une représentation plus robuste du comportement de l'utilisateur. D'un autre côté, comprendre l'intention de recherche dans une session de recherche peut représentent différents niveaux de difficulté - certaines intentions sont faciles à comprendre tandis que d'autres sont plus difficiles et nuancées. Mélanger directement ces sessions dans le même batch d'entraînement perturbera l'optimisation du modèle. Par conséquent, nous proposons un cadre d'apprentissage par curriculum avec des examples allant de plus faciles à plus difficiles. Les deux méthodes proposées obtiennent de meilleurs résultats que les méthodes existantes sur deux jeux de données de logs de requêtes réels.

Les trois derniers articles se concentrent sur les systèmes de dialogue fondé les documents/connaissances. Nous proposons d'abord un mécanisme de sélection de contenu pour le dialogue fondé sur des documents. Les expérimentations confirment que la sélection de contenu de document pertinent en fonction du contexte du dialogue peut réduire le bruit dans le document et ainsi améliorer la qualité du dialogue. Deuxièmement, nous explorons une nouvelle tâche de dialogue qui vise à générer des dialogues selon une description narrative. Nous avons collecté un nouveau jeu de données dans le domaine du cinéma pour nos expérimentations. Les connaissances sont définies par une narration qui décrit une partie du scénario du film (similaire aux dialogues). Le but est de créer des dialogues correspondant à la narration. À cette fin, nous concevons un nouveau modèle qui tient l'état de la couverture de la narration le long des dialogues et déterminer la partie non couverte pour le prochain tour. Troisièmement, nous explorons un modèle de dialogue proactif qui peut diriger de manière proactive le dialogue dans une direction pour couvrir les sujets requis. Nous concevons un module de prédiction explicite des connaissances pour sélectionner les connaissances pertinentes à utiliser. Pour entraîner le processus de sélection, nous générons des signaux de supervision par une méthode heuristique. Les trois articles examinent comment divers types de connaissances peuvent être intégrés dans le dialogue.

Le contexte est un élément important dans la RI ad-hoc et le dialogue, mais nous soutenons que le contexte doit être compris au sens large. Dans cette thèse, nous incluons à la fois les interactions précédentes avec l'utilisateur, le document et les connaissances dans le contexte. Cette série d'études est un pas dans la direction de l'intégration d'informations contextuelles diverses dans la RI et le dialogue.

Mots clés: recherche d'information, classement de documents sensible au contexte, système de dialogue basé sur la recherche, dialogue fondé sur les connaissances

Abstract

Information retrieval (IR) or search systems have been widely used to quickly find desired information for users. Ranking is the central function of IR, which aims at ordering the candidate documents in a ranked list according to their relevance to a user query. While IR only considered a single query in the early stages, more recent systems take into account the context information. For example, in a search session, the search context, such as the previous queries and interactions with the user, is widely used to understand the user's search intent and to help document ranking. In addition to the traditional ad-hoc search, IR has been extended to dialogue systems (*i.e.*, retrieval-based dialogue, *e.g.*, XiaoIce), where one assumes a large repository of previous dialogues and the goal is to retrieve the most relevant response to a user's current utterance. Again, the dialogue context is a key element for determining the relevance of a response.

The utilization of context information has been investigated in many studies, which range from extracting important keywords from the context to expand the query or current utterance, to building a neural context representation used with the query or current utterance for search. We notice two important insufficiencies in the existing literature. (1) To learn to use context information, one has to extract positive and negative samples for training. It has been generally assumed that a positive sample is formed when a user interacts with a document in a context, and a negative sample is formed when no interaction is observed. In reality, user interactions are scarce and noisy, making the above assumption unrealistic. It is thus important to build more appropriate training examples. (2) In dialogue systems, especially chitchat systems, responses are typically retrieved or generated without referring to external knowledge. This may easily lead to hallucinations. A solution is to ground dialogue on external documents or knowledge graphs, where the grounding document or knowledge can be seen as new types of context. Document- and knowledge-grounded dialogue have been extensively studied, but the approaches remain simplistic in that the document content or knowledge is typically concatenated to the current utterance. In reality, only parts of the grounding document or knowledge are relevant, which warrant a specific model for their selection.

In this thesis, we study the problem of context-aware ranking for ad-hoc document ranking and retrieval-based dialogue. We focus on the two problems mentioned above. Specifically, we propose approaches to learning a ranking model for ad-hoc retrieval based on training examples selected from noisy user interactions (*i.e.*, query logs), and approaches to exploit external knowledge for response retrieval in retrieval-based dialogue. The thesis is based on five published articles.

The first two articles are about context-aware document ranking. They deal with the problem in the existing studies that consider all clicks in the search logs as positive samples, and sample unclicked documents as negative samples. In the first paper, we propose an unsupervised data augmentation strategy to simulate potential variations of user behavior sequences to take into account the scarcity of user behaviors. Then, we apply contrastive learning to identify these variations and generate a more robust representation for user behavior sequences. On the other hand, understanding the search intent of search sessions may represent different levels of difficulty – some are easy to understand while others are more difficult. Directly mixing these search sessions in the same training batch will disturb the model optimization. Therefore, in the second paper, we propose a curriculum learning framework to learn the training samples in an easy-to-hard manner. Both proposed methods achieve better performance than the existing methods on two real search log datasets.

The latter three articles focus on knowledge-grounded retrieval-based dialogue systems. We first propose a content selection mechanism for document-grounded dialogue and demonstrate that selecting relevant document content based on dialogue context can effectively reduce the noise in the document and increase dialogue quality. Second, we explore a new task of dialogue, which is required to generate dialogue according to a narrative description. We collect a new dataset in the movie domain to support our study. The knowledge is defined as a narrative that describes a part of a movie script (similar to dialogues). The goal is to create dialogues corresponding to the narrative. To this end, we design a new model that can track the coverage of the narrative along the dialogues and determine the uncovered part for the next turn. Third, we explore a proactive dialogue model that can proactively lead the dialogue to cover the required topics. We design an explicit knowledge prediction module to select relevant pieces of knowledge to use. To train the selection process, we generate weak-supervision signals using a heuristic method. All of the three papers investigate how various types of knowledge can be integrated into dialogue.

Context is an important element in ad-hoc search and dialogue, but we argue that context should be understood in a broad sense. In this thesis, we include both previous interactions

and the grounding document and knowledge as part of the context. This series of studies is one step in the direction of incorporating broad context information into search and dialogue.

Keywords: information retrieval, context-aware document ranking, retrieval-based dialogue system, knowledge-grounded dialogue

Contents

List of tables

List of figures

List of abbreviations

IR	Information Retrieval
NLP	Natural Language Processing
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long-Short Term Memory
GRU	Gate Recurrent Unit
BERT	Bidirectional Encoder Representation from Transformers
MLM	Masked Language Modeling
NSP	Next Sentence Prediction
LTR	Learning-To-Rank
SMT	Statistical Machine Translation

FC	Fully-Connected
MLP	Multi-Layer Perceptron
MAP	Mean Average Precision
MRR	Mean Reciprocal Rank
NDCG	Normalized Discounted Cumulative Gain

Acknowledgment

Getting to know Prof. Jian-Yun Nie is the most precious part of my experience in Montreal. He is an ultimate example of a supportive and compassionate person. Over the years, I have many thanks to express: thank him for spending a great deal of time on editing my papers and correcting all the typos and grammatical errors made by a non-native English speaker; thank him for inviting us to his home numerous times and taking us outside to relax; thank him for always being flexible with my schedules such as going back to China and visiting my family members. The opportunity of working with him is the most meaningful aspect of my time at the University of Montreal, and these experiences transcend my modest academic accomplishments in graduate school.

I want to thank Prof. Zhicheng Dou. It is hard to quantify how much help he has provided for me, including but not limited to writing advice for my paper, computing resources for my project, and internship recommendations. He often emphasized to me the significance of undertaking challenging tasks and resolving difficult research problems. However, to satisfy my utilitarian drive for publication, I continued to work on small incremental projects. The greatest regret of my PhD career is not taking his advice.

Over the time in graduate school, I have been blessed with the friendship of many amazing people. I would like to thank all friends from the University of Montreal: Zhaoliang Yang, Peng Lu, Shengchao Liu, Yifan Nie, Yabo Ling, Fengran Mo, Qianqian Xie, Yunhe Li, Bang Liu, and Suyuchen Wang, who have given me a sense of belonging. Pan Du helped a lot on my paper writing. Céline Bégin, Fabrizio Gotti, and Philippe Langlais assisted me greatly with department and lab-related concerns. Over the years, I have had the opportunity of befriending, working with, or learning from many excellent students, professors, and researchers at Renmin University of China, including Ji-Rong Wen, Ruihua Song, Xin Zhao, Kun Zhou, Yujia Zhou, Anwen Hu, Zhengyi Ma, Xubo Qin, Hongjin Qian, Kelong Mao, Qian Cao, Chuhao Jin, Hanxun Zhong, Jianwei Hou, and many others who I am grateful for but cannot list all their names.

Finally, I would like to thank my family. Thank you for always providing me with love and care.

Chapter 1

Introduction

Because it's there.

George Mallory
the pioneer of the Mount Everest expedition

Information access is one of the fundamental daily needs of human beings. To quickly obtain the desired information, various information retrieval (IR) systems have been developed. For example, search engines such as Google, Bing, and Baidu are IR systems on the Web that are used to retrieve relevant pages in response to user queries, providing a convenient and efficient way to access information from the Web. IR is not limited to retrieving texts. In dialogue systems (chatbots), such as Microsoft Xiaoice, Apple Siri, and Google Assistant, IR systems can also be used to retrieve the proper replies to user input utterances, thereby producing natural and fluent human-machine conversations. With the exponential growth of information, building effective IR systems has attracted more and more attention.

Practical IR systems often include two essential components: retrieval and ranking, both of which influence the performance of the system. To explain their differences, we consider a simplified IR system that takes the user-issued *query* as input and returns a list of relevant documents from a document repository.[1] In this case, the **retrieval** component seeks to retrieve numerous candidate documents from a large repository. This component is required to be efficient because the repository is often enormous. Okapi BM25 [**157**] is one of the most commonly used algorithms for relevance computing in this stage. The documents with the highest BM25 scores are returned and form a candidate document list. Even when the returned document list is significantly smaller than the size of the repository, it is still too large (*e.g.*, hundreds or thousands of documents) for human users to check. Moreover, BM25

[1] Henceforth, we will refer to the user input as a "query". It can be a search query for search engines, a user utterance for chatbots, a question for question answering systems, etc.

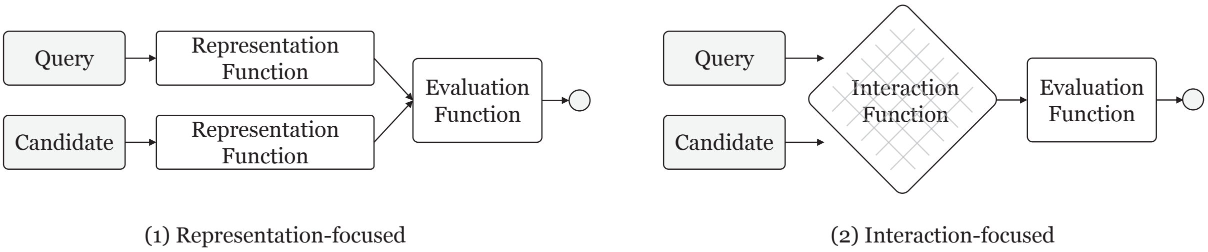


Fig. 1. Representation-focused and interaction-focused architectures of ad-hoc ranking models.

scores merely assess the word overlap between the query and document, which is insufficient for measuring the relevance accurately. Therefore, a **ranking** component is necessary for a practical IR system. This component is intended to compute more precise relevance scores between a query and a document and to rerank the candidate document list (hence, this process is also known as reranking). As the candidate document list is much smaller than the full repository, it is feasible to apply more complex ranking algorithms. In the early stages, researchers proposed many learning-to-rank algorithms [**15**] to learn a ranking function. They typically rely on manually extracted features (such as document length and keyword counts), which severely restricts their accuracy. Following the advancement of deep learning, many neural network-based methods have been proposed. Due to their capability of automatic feature extraction and their strong performance on various tasks, they have swiftly become the dominant methods. Consequently, the goal of this thesis is to enhance the ranking performance based on neural models.

Early academic studies attempted to improve ad-hoc ranking performance by employing neural models. Ad-hoc ranking takes **a single query** into account and requires models to rank relevant texts as high as possible. Thus, the key challenge is determining how to compute the relevance between the query and candidate text. As illustrated in Figure 1, existing methods can be categorized into two groups according to their different assumptions for relevance evaluation. The first group is representation-focused architectures. It assumes that the relevance is contingent on the compositional meaning of the input texts. Therefore, representation-focused methods usually design a complex representation function for both the query and candidate text to obtain their representations. Then, the relevance score is calculated based on these representations through a matching function (*e.g.*, cosine similarity). For example, DSSM [**75**] utilizes two fully-connected networks to represent the query and candidate text and computes their relevance by cosine similarity. Based on a similar two-tower architecture, ARC-I [**72**] and MV-LSTM [**190**] respectively apply 1D convolutional neural networks (CNNs) and bi-directional LSTMs to enhance the representation learning. Nevertheless, researchers discovered that it is very challenging to represent the query or candidate text with a single vector, and the case is even worse when the text is

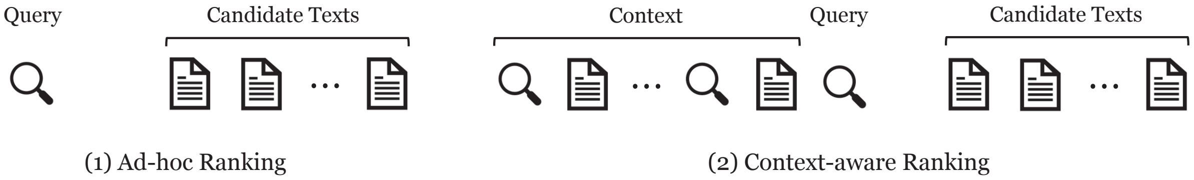


Fig. 2. Differences between ad-hoc ranking and context-aware ranking.

long. In contrast, another group of interaction-focused architectures assumes that the relevance is essentially about the relationship between the query and candidate text; so, it is more effective to directly learn the relevance relation from the basic interactions between document and query (*e.g.*, between tokens in them). These approaches emphasize the design of complex interaction and evaluation functions in order to compute the relevance score. For instance, KNRM [**211**] performs fine-grained interaction between the tokens in query and candidate text to generate a matching matrix, from which a kernel pooling method calculates the relevance score. MatchPyramid [**134**] directly constructs a matching matrix using the word embeddings of the query and candidate text. Then, a CNN is used to extract the matching features, and a feed-forward neural network is used to compute the relevance score. Lately, pre-trained language models, such as BERT [**34**], are also applied for ad-hoc ranking, where interaction can be captured through attention mechanisms. The interaction-focused architecture generally performs better than the representation-focused architecture because it performs interaction over low-level representations (*e.g.*, word embeddings) and preserves the original matching signals to a significant extent. However, these approaches are more time consuming than representation-based approaches.

In the utilization scenario, traditional IR focuses on one-shot search, in which a single query is used to retrieve documents. In more realistic search scenarios, users perform search in a session, *i.e.*, a query is submitted to the system within a **search context**. With the evolution of ranking models, researchers have found that the search context is also very helpful for capturing user search intents. The task that leverages both the context and query to rank candidate texts is called **context-aware ranking**, which is the focus of our thesis. We show the comparison between ad-hoc ranking and context-aware ranking in Figure 2. In the former, a single query is considered without context, while in the latter, the query is submitted after a series of interactions between the system and the user (queries and retrieved documents), so the intent can be influenced by the latter.

Context-aware ranking has been studied in many scenarios, whereas in this thesis, we focus on the two most prevalent ones: **search** and **dialogue**. In the search scenario, user behavior has progressively shifted from a one-shot query to multiple interactions with the search engines, particularly for complicated and ambiguous information needs. A typical scenario involves initiating a search with general queries and then refining them after browsing documents. During this process, search intents may be modified and refined. It is essential to model the search context because it reflects how the search intents have gradually evolved and how the users' information needs are satisfied, which is helpful for understanding the users' real search intents underlying the current query. In the dialogue scenario, the dialogue context is also critical for understanding the query (*i.e.*, user input utterance), as the context may provide background knowledge (such as topic) of the dialogue and users in a dialogue frequently omit information that has already been mentioned in context. In recent years, researchers have found that search and dialogue are not two independent scenarios, but can be naturally combined. Conversational search is the result of this combination [**32, 117, 143**], where users tend to use natural language texts as queries and the entire search behaves like a conversation between users and search engines. Intuitively, the strategy for conversational search should account for both the search requirement and the dialogue's characteristics. We will study the problem of context-aware document ranking and context-aware dialogue ranking in relation to the search and dialogue scenarios.

This thesis is composed of five published articles that address the problems of context-aware IR and context-aware dialogue. In the following sections, we will describe the motivations of these studies and summarize the main ideas used in them.

1.1. Search: Context-Aware Document Ranking

As aforementioned, the search context is critical for understanding the potential search intent of the current query. The task is defined as ranking a list of candidate documents based on the current query and previous search behaviors (including previous queries and clicked documents). The challenge lies in inferring user intents from search context. Earlier studies sought to extract statistical or rule-based features from the search context and used them to expand the query [**198, 208**]. These methods require a lot of human expertise in feature extraction, which greatly restricts their applicability and performance. Afterwards, neural networks were introduced to model the search context. A typical structure is based on hierarchical recurrent neural networks (RNNs) [**3**]. Concretely, each of the queries in the search context is first represented as a vector by a lower-level RNN. Then, a higher-level RNN receives the vectors as input, aggregates the information in all queries, and outputs a vector for document ranking. This structure is extended by incorporating the previously clicked documents into the context modeling [**4**]. Recent years have witnessed great success of pre-trained language models on various IR and NLP tasks [**48, 54, 85, 130, 146, 212, 245**]. Some researchers concatenated historical queries and clicked documents as a long sequence and applied BERT [**34**] to perform document ranking [**149**]. It yields encouraging results.

All existing methods train new models for context-aware document ranking following a standard paradigm: they first treat each search session in the search log as a positive sample, while sampling unclicked documents for the current query to form negative samples; then, all training samples are mixed together and uniformly sampled as batches of data to train the model. This paradigm is straightforward and effective, but it has two limitations.

(1) All search sessions (sequences) are treated as definite and accurate. That is, an observed sequence from search logs is considered a positive sample, whereas any unseen sequence is either neglected or used as a negative sample. This strict view disregards the flexibility and sparseness of user activities during a search session. When using a search engine, it is common for users to have diverse interaction patterns or submit different queries for the same information need. These interactions vary from user to user and search context to search context. The same holds true for user click behaviors: one can click on different documents for the same information need. This flexibility has not been explicitly addressed in previous research when extracting the training examples. In the same way, user behaviors are sparse in the sense that not all the relevant documents are selected by users. It is thus wrong to assume that all the unselected documents are irrelevant. When we have a large amount of log data and are only interested in common user patterns, existing methods can still extract emerging strong patterns and discard accidental variations. Unfortunately, when we train models on log data using this assumption, the models are incapable of capturing the nuances in user behaviors and coping with the variance. A better approach is to view the data as they are: they are just samples of possible query formulations and interactions, but much more are not shown in the logs.

To tackle this problem, in our first article [**245**], we propose **a contrastive learning-based method to enhance the representation of the user behavior sequence and improve the performance of context-aware document ranking**. We first design three data augmentation strategies to generate possible variations for user search behavior sequences in a search log. Concretely, we mask some terms in a query or document, delete some queries or documents from the sequence, or reorder the sequence. These strategies aim to simulate typical variations in user behaviors. The generated sequences can be considered similar to the observed ones. Then, we employ contrastive learning to distinguish similar and dissimilar sequence pairs. It attempts to bring similar sequences (generated variants) together and separate dissimilar sequences (other randomly sampled sequences). In this way, models can better cope with the variations and become more robust to unseen user behavior sequences. Our experiments on two real search log datasets demonstrate the effectiveness of contrastive learning for context-aware document ranking.

(2) The second limitation of the existing approaches lies in the fact that all training samples are uniformly sampled to form a training batch, with the assumption that all positive samples are of similar importance to reflect relevance. It goes in a similar way for negative samples that reflect irrelevance. While it is typically true that both positive and negative examples are beneficial for training a ranking model, they are not equally useful at the training

stage. In the context of human learning, it is common to follow a curriculum that regulates the ordering and content of the educational materials [**92, 137, 167**]. With this strategy, students can leverage previously acquired knowledge to facilitate the acquisition of new and more complex concepts. Things are similar for the training of a context-aware document ranking model. Various training samples have varying levels of difficulty for learning. For example, given a search context, positive documents are easily identifiable if they are close to the current query and notably distinct from negative documents. In contrast, if the positive and negative documents are quite similar, they are hard to distinguish. When human learners are faced with samples of mixed difficulty, they may become confused because signals from different samples may appear inconsistent or because they lack the necessary information to comprehend difficult samples. Recent research in machine learning has demonstrated that learning with a batch of samples of mixed difficulties may disturb the optimization, in particular when the network is deep [**7**]. In this case, curriculum learning, *i.e.*, learning from easy samples before hard samples, is especially useful.

Motivated by these observations, in our second article [**246**], we propose **a dual curriculum learning framework for context-aware document ranking**. This new training framework takes the difficulty of different training samples into account. We develop two complementary curricula, one for positive (search context, document) pairs and one for negative pairs. In the curriculum for positive pairs, sampling is restricted to easy pairs in the early steps, and then gradually expanded to the entire sample set, so that hard pairs can also be learned. In the curriculum for negative pairs, we do the opposite: sampling from all pairs in the early steps, then restricting gradually to hard pairs in the late steps. This curriculum strategy is applicable to any existing learning approach, and we have integrated it into three state-of-the-art approaches for context-aware document ranking. Experiments on two search log datasets show that our curriculum learning method significantly improves the three strong baselines, including the one that we previously proposed in the first article.

1.2. Dialogue: Context-Aware Dialogue Ranking

Many search engines now support the use of natural language queries, transforming the search process into a “conversation” between humans and search engines. Hence, researchers have utilized techniques of dialogue systems to enhance the performance of search engines.

Dialogue systems aim at providing a response to a human input. Existing approaches can be roughly categorized into generation-based and retrieval-based, depending on whether the response is generated from scratch or retrieved from a repository. ChatGPT [**132**] is a recently proposed generation-based dialogue system. With the great power of its backbone large language models and instruction tuning, it can help users with various tasks, such as writing emails, proofreading texts, and summarizing documents. Nevertheless, since all responses are generated by the model, their factualness and accuracy cannot be guaranteed. We concentrate on the retrieval-based methods, which can also be treated as an application of IR systems. The responses are retrieved from a large repository, which can provide good quality and fluency. Besides, these responses can be controlled by constraining the repository, which are suitable for some domain-specific applications, such as legal consulting based on legal provisions or policy analyzing from government websites. It is important to note that all of these applications rely on a rich repository.

Similar to the search scenario, studies on retrieval-based dialogue systems have progressed from modeling a single user input utterance [**111**] to modeling the entire dialogue context [**178, 205, 223, 230**]. The dialogue context can provide background knowledge (such as topic) of the dialogue, which is crucial for selecting an appropriate response. This task is referred to as context-aware dialogue ranking. It is defined as selecting a proper response from a candidate list for a dialogue context. The task is similar to context-aware document ranking, but with the following replacement: "search context" → "dialogue context", "query" → "user input utterance", and "candidate document list" → "candidate response list".

In the early years, researchers utilized hierarchical RNNs to represent each utterance within the context and aggregate its information as a vector. Then, another RNN was utilized to represent the response candidate as a vector. Finally, the context-response matching score can be computed using cosine similarity [**111**]. This method can be categorized into the representation-focused architecture mentioned earlier. Similarly, various approaches based on the interaction-focused architecture are proposed and quickly became the mainstream due to their superior performance. A typical framework is "representation-matching-aggregation" [**178, 205, 223, 230**]. Specifically, all utterances in the dialogue context and the response candidate are first represented as word-level vectors by a group of neural networks, such as CNNs or RNNs. Then, an interaction matrix is computed based on the word-level representations, and the second group of neural networks is used to extract matching features. Finally, a third group of neural networks is employed to aggregate the matching signals and determine the final matching score. Recently, pre-trained language models have been introduced to the dialogue response ranking task with excellent success [**54, 67**] (we will describe them in Section 2.3.1.2).

Thanks to recently proposed powerful models, existing retrieval-based dialogue systems have been capable of selecting more suitable responses for dialogue contexts. Nonetheless, one can still find a big gap between human-machine dialogues and real human dialogues. The lack of proper knowledge is the main obstacle for dialogue models to delve deeply into a topic. Whereas pre-trained language models have learned a lot of knowledge from large-scale text corpora, it is infeasible to store all world-knowledge in their parameters. Consequently, providing external knowledge as additional input to the system is an effective way to improve dialogue performance. Knowledge-grounded dialogue models have attracted increasing interest. In addition to the dialogue context, external knowledge is incorporated as additional "context" to facilitate dialogue modeling. Various knowledge can be introduced to achieve different purposes. For instance, endowing dialogue models with pre-defined personas is an effective way to produce more interesting and engaging dialogues [**100, 145, 226**], which are referred to as personalized dialogue systems. Document-grounded dialogues have also been investigated to simulate the scenario that people discuss about a document, *e.g.*, a piece of news or a Wikipedia page. Domain-specific knowledge graphs are also a common way to provide external knowledge, which has shown to be effective for boosting domain-specific dialogues [**105, 127, 242, 234**].

Although external knowledge has been introduced to human-machine dialogues in an effort to increase their quality, the relationship between the knowledge and dialogue remains weak, as evidenced by several factors: (a) The external knowledge is incorporated into the dialogue process without careful selection. One often hopes that the use of knowledge can be automatically learned by the signals back-propagated from the final response selection. Sadly, this is impracticable due to the complexity and diversity of the knowledge integration process. (b) The knowledge plays just a supplementary role in the dialogue process, such as providing some background information. Under this circumstance, models often employ little to no knowledge. Things are even worse when there is a lot of noise in the knowledge (*e.g.*, in a knowledge graph that is automatically extracted). Motivated by these observations, we investigate methods for **bridging knowledge and dialogue more tightly** so as to enhance dialogue performance.

(1) Our first exploration in this direction is to enable the model to do some knowledge selection prior to incorporating it into the dialogue. It is conducted under the setting of document-grounded dialogues, wherein the document contains a great deal of information (perhaps containing noise) awaiting selection. The task is formally defined by selecting a proper response that should be not only coherent with the dialogue context, but also consistent with the provided document. The common strategy in existing approaches is ranking responses according to a combination of context-response matching and document-response matching [**58, 226, 231**]. The latter encourages responses to be pertinent to the document's content. Yet, based on our observations, a good response need not be tied to the entire content of the external knowledge (*i.e.*, document), but rather to a small part of it. For example, given a Wikipedia article about a movie, when discussing the actors of the movie, the model would only need to access the information related to these actors, while information about the movie's producer, publisher, or plot is irrelevant. Intuitively, it is necessary and effective to perform a selection on the knowledge before feeding it into the dialogue.

Table 1. An example of movie script extracted from the movie *Forrest Gump*.

Narrative	Jenny doesn't like to go home. To accompany Jenny, Gump decides to go home later. Gump is Jenny's best friend.
Script line 1	Mama's going to worry about me.
Script line 2	Just stay a little longer.
Script line 3	OK, Jenny, I'll stay.
Script line 4	You are my most special friend.

To address this problem, in our third article [**247**], we propose **a content selection network for document-grounded retrieval-based dialogue systems**. On the one hand, we design a new gate mechanism to achieve a hard selection on the document content. Content relevant to the current dialogue step will be assigned a higher weight and pass the selection gate, while the irrelevant parts will be rejected. This allows our model to automatically filter out noise from the document. On the other hand, as the dialogue topic evolves, before performing the content selection, we design a decay mechanism and determine the current dialogue context by focusing on the most recent utterances, rather than on the entire dialogue history. This further improves the accuracy of the content selection. Overall, we refine the document prior to feeding it into the model, effectively narrowing the gap between the document and dialogue, thereby improving the dialogue performance. Our experiments on two public datasets demonstrate the effectiveness of our proposed method.

(2) Following the previous article, we investigate ways to further promote the proximity of knowledge and dialogue. We consider the following research question: Can the dialogue fully cover the given knowledge? Since there is no existing dataset supporting our study, we first construct a new dataset based on movie scripts. An example is illustrated in Table 1. In this dataset, each piece of script is associated with a narrative, which describes what happens in the piece of script. This is the most relevant dataset we are able to collect for the research, in which the piece of script corresponds to the dialogue, while the narrative represents the knowledge. The task is characterized as selecting a proper script line that is both coherent with the context and consistent with the given narrative. Despite having a similar form, the narrative in this research problem plays a totally different role: its content should be completely covered by the script session. Therefore, it is inappropriate to take the narrative as a general context and do narrative-response matching. Generally, there are two challenges in using the narrative: (a) When determining the next script line, one should keep track of which part of the narrative has already been covered by previous script lines. This role of the narrative is different from that of the context. In general dialogue modeling, typical methods of context-aware response ranking focus on measuring the matching degree between the context and the response. If the same matching mechanism is performed for a narrative, many lines will be redundant. So, tracking the narrative is an essential step for

the task. (b) When determining the next script line, it is also necessary to decide which part of the narrative should be expressed. The narrative is often organized in a specific chronological order. Thus, the model for narrative-guided script ranking needs to be able to learn how to utilize the remaining information in the narrative.

To address these issues, in our fourth article [**250**], we propose **a new model that can leverage the narrative to generate movie script**. Our model is able to keep track of what has been stated in the narrative and what is still remaining to select the next line by an updating mechanism. This is an example of a more general dialogue system that aims to accomplish a set or sequence of tasks, for which the system should keep track of what has been done, and what should be done next. We also design a content prediction module that can assist the model in learning to predict the narrative content that should be covered by the next line. Experimental results on our newly collected dataset indicate that the narrative does indeed play a different role than a general context, and our model can control the script selection process so that the whole script is close to the pre-defined narrative. Our study answers the research question that the dialogue can be forced to cover all of the required knowledge with proper model architectures. It paves the way for more controllable dialogue systems.

(3) The previous article demonstrates the possibility of utilizing external knowledge to influence the dialogue process. Going one step further in the same direction, we explore designing a dialogue model that may proactively adjust the dialogue topic and lead the dialogue using the provided knowledge. At the time we analyzed this problem, most approaches for dialogue systems worked passively, *i.e.*, they could only respond to human input but fail to lead a dialogue proactively. This kind of system may quickly become boring for the user. We start our study by following a preliminary work on proactive dialogue [**204**], where models are trained to proactively lead the dialogue by covering some pre-defined topics (called a goal) with supplementary knowledge. This study treats both the goal and supplementary knowledge as additional context and perform attention to integrate them into the dialogue response ranking process. The modeling of goals and knowledge are all optimized through the final ranking loss, making it difficult to tell if an error is the result of suboptimal goal modeling, poor knowledge prediction, or bad response selection.

To tackle this problem, in our fifth article [**248**], we propose **a multi-task learning framework for knowledge-grounded proactive human-machine conversation**. We design an explicit knowledge prediction module to select the relevant piece of knowledge to use. This module is combined with a response selection module to form a multi-task learning framework. The knowledge prediction module first tracks the state of goal achievement, *i.e.*, which part of the goal has been achieved, and then leverages the dialogue context to predict which knowledge should be used in the current turn. The response selection module then relies on the selected knowledge to determine the final answer. Different from existing methods, we explicitly optimize knowledge prediction using automatically generated weak-supervision signals to assist in learning to predict the relevant knowledge more accurately. The experimental results demonstrate that the explicitly trained knowledge prediction process can indeed identify the most relevant piece of knowledge to use, resulting in superior performance over the state-of-the-art methods.

In summary, in this thesis, we leverage context information in a document reranking model and use contrastive learning to train the model more effectively; and for retrieval-based conversation, we explore several directions to leverage external knowledge so that the conversations can be more knowledgeable and proactive. This thesis is organized as follows: we will first introduce some background knowledge and related work from our studies. Then, we will present our two published articles on context-aware document ranking. Both are dedicated to enhancing the existing training paradigm and improving performance. Next, we will present three other articles on knowledge-grounded retrieval-based dialogue systems, all of which improve dialogue performance by strengthening the interaction between knowledge and dialogue. Finally, we will give some general conclusions about the thesis.

Chapter 2

Background

In this chapter, we will provide background knowledge about our work. Namely, we will describe basic neural networks, document ranking, dialogue systems, commonly used evaluation metrics, and statistical significance tests.

2.1. Basic Neural Networks

In modern natural language processing, an essential step is to create vector representations for words and texts. A good word representation can lead to better performance on downstream tasks. In this section, we briefly introduce some neural architectures that are commonly used to compute text representations.

2.1.1. Traditional Models

2.1.1.1. Convolutional Neural Network (CNN). Convolutional neural networks are based on the shared-weight architecture of the convolutional kernels that slide along input features and provide feature maps [**93**]. They are first applied to image processing tasks. There are also many applications to NLP tasks, for example, capturing n-gram features from word embeddings [**205, 230**]. According to different shapes of input, CNNs can be implemented by kernels with different dimensions. Here, we use the most commonly applied 2D-CNN as an example for illustration. Given an input matrix $\mathbf{M}$, the k-th kernel $\mathbf{W}^k$ slides over $\mathbf{M}$ and generates a feature map $\mathbf{z}^k$ as follows:

$$\mathbf{z}^k_{i,j} = \sigma(\sum_{s=0}^{r_k-1}\sum_{t=0}^{r_k-1} \mathbf{W}^k_{s,t} \cdot \mathbf{M}_{i+s,j+t} + b^k), \tag{2.1.1}$$

where r_k denotes the size of the k-th kernel; b^k is a bias parameter; $\mathbf{M}_{i,j}$ is the element at the i-th row and j-th column of $\mathbf{M}$; and $\sigma(\cdot)$ is a non linear function, often implemented by ReLU. Overall, with k kernels, CNNs can output k feature maps. A pooling layer is often used after a convolutional layer to reduce the dimensions of data by combining the elements in feature maps. For example, a max-pooling layer can be defined as:

$$\mathbf{p}_{i,j}^{k} = \max_{0 \leq s < d_k} \max_{0 \leq t < d_k} \mathbf{z}_{i \cdot d_k + s, j \cdot d_k + t}^{k}, \tag{2.1.2}$$

where d_k is the size of the pooling kernel.

CNNs can be stacked into multiple layers so as to extract features in different granularities. They simulate the *receptive field* in biological process, thus naturally fitting to extract image features.

2.1.1.2. Recurrent Neural Network (RNN). Recurrent neural networks are a kind of neural network architecture specifically designed to deal with sequential data. It was originally proposed for modeling time series data [**36, 44**]. Later, researchers found its great potential in NLP tasks [**123, 124**], as a sentence or a text is naturally a sequence of words. Formally, given a sequence of words $[y_1,y_2,\cdots,y_N]$, where y_t is associated with a k-dimensional vector representation $\mathbf{x}_t$, RNN computes a hidden vector $\mathbf{h}_t$ at each time step t, which is often considered as a representation that embeds all information in previous words, *i.e.*, $[y_1, y_2, \cdots, y_t]$. $\mathbf{h}_t$ is obtained by a function g, which can be defined as:

$$\mathbf{h}_t = g(\mathbf{h}_{t-1},\mathbf{x}_t). \tag{2.1.3}$$

The function g can have different forms, for example:

$$g(\mathbf{h}_{t-1},\mathbf{x}_t) = \sigma(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{x}_t + \mathbf{b}), \tag{2.1.4}$$

where $\mathbf{W}$, $\mathbf{U} \in \mathbb{R}^{k \times k}$ are parameters that aggregate the previous state and the current input, and $\mathbf{b} \in \mathbb{R}^k$ are biases. $\sigma(\cdot)$ is a non-linear function, such as sigmoid, tanh, or ReLU.

The naive RNNs have two well-known problems: gradient exploding and gradient vanishing [**8**]. The gradient will be extremely large or approach zero when the training error is backpropagated over time. These problems limit RNNs' capability of capturing the long-term dependency in long sequences. To tackle these problems, several new architectures are proposed, where the most famous two are the Long Short-Term Memory model (LSTM) [**71**] and the Gate Recurrent Unit model (GRU) [**27**]. They both apply gate mechanisms to control the information flow between recurrent units and improve the modeling of long-term dependencies. RNNs typically model sequential data in one direction (*e.g.*, from left to right), and researchers also found that applying two RNNs in bi-directions (called Bi-RNN) and combining their hidden states can further improve their capability [**71**].

2.1.1.3. Transformer. Transformer [**186**] is a neural architecture proposed later, which has quickly become the mainstream method for various tasks. The vanilla Transformer consists of an encoder and a decoder. They have similar structures but the decoder includes an additional attention mechanism to aggregate information from the encoder. In this thesis, we mainly use the encoder, so we briefly introduce it here.

A Transformer encoder contains a stack of N identical layers, each of which has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple fully-connected feed-forward network. Specifically, the attention mechanism can be formalized as a scaled dot-product attention with three inputs as follows:

$$\mathrm{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}, \tag{2.1.5}$$

where $\mathbf{Q} \in \mathbb{R}^{N_q \times d}$, $\mathbf{K} \in \mathbb{R}^{N_k \times d}$, and $\mathbf{V} \in \mathbb{R}^{N_k \times d}$ denote the query, key, and value matrices, respectively. N_q and N_k denote the number of query and key/value vectors. d is the dimension of the representation. The attention mechanism can be explained as follows: for each query vector in $\mathbf{Q}$, it first computes the dot-products of the query with all keys, aiming to evaluate the similarity between the query and each key. Then, it is divided by $\sqrt{d}$ for scaling and applies a softmax function to compute the weights on the values. Finally, the new representation of the query is calculated as weighted sum of values. To make the attention function more flexible, a multi-head attention mechanism is often applied. Instead of performing a single attention, multi-head attention first projects the input $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ into h different spaces, each of which has a dimension of $d_m = d/h$. Then, it computes the attention with the function defined in Equation (2.1.5) as follows:

$$\mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{concat}\left(\left[\mathrm{Attn}(\mathbf{Q}\mathbf{W}_i^Q,\mathbf{K}\mathbf{W}_i^K,\mathbf{V}\mathbf{W}_i^V)\right]_{i=1}^{h}\right)\mathbf{W}^O, \tag{2.1.6}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V \in \mathbb{R}^{d \times d_m}$, and $\mathbf{W}^O \in \mathbb{R}^{d \times d}$ are parameters to project the input matrices, and the output $\mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V})$ has the same shape as $\mathbf{Q}$.

Based on the multi-head attention mechanism, a Transformer encoder layer can be defined as:

$$\mathrm{encoder}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{LayerNorm}(\mathbf{A} + \mathrm{FFN}(\mathbf{A})), \tag{2.1.7}$$

$$\mathbf{A} = \mathrm{LayerNorm}(\mathbf{Q} + \mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V})), \tag{2.1.8}$$

$$\mathrm{FFN}(\mathbf{X}) = \mathrm{ReLU}(\mathbf{X}\mathbf{W}^{F_1} + \mathbf{b}^{F_1})\mathbf{W}^{F_2} + \mathbf{b}^{F_2}, \tag{2.1.9}$$

where $\mathbf{W}^{F_1}$, $\mathbf{W}^{F_2}$, $\mathbf{b}^{F_1}$, and $\mathbf{b}^{F_2}$ are parameters. $\mathrm{LayerNorm}(\cdot)$ stands for layer normalization [**6**]. When using Transformer encoder for representation, a commonly used variant is self-attention, which can be defined as:

$$\mathrm{SelfAttn}(\mathbf{X}) = \mathrm{encoder}(\mathbf{X},\mathbf{X},\mathbf{X}). \tag{2.1.10}$$

By performing attention to the input $\mathbf{X}$ itself, we can obtain a more effective representation for $\mathbf{X}$, allowing the tokens in $\mathbf{X}$ to connect among themselves.

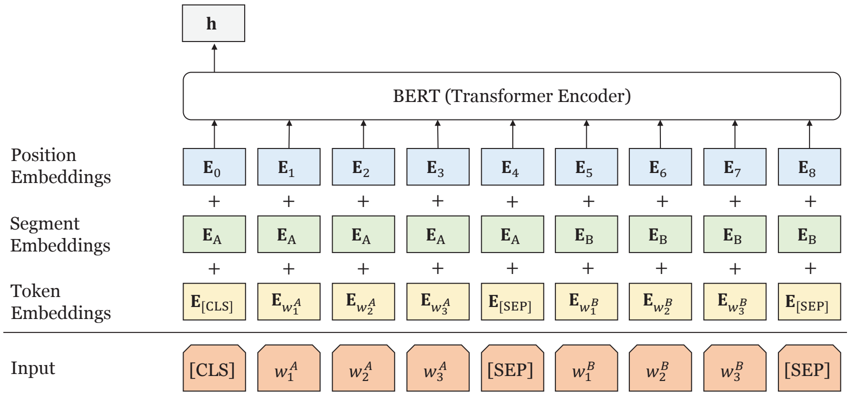

h
BERT (Transformer Encoder)
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8
Segment Embeddings
E_A E_A E_A E_A E_A E_B E_B E_B E_B
Token Embeddings
$E_{[CLS]}$ $E_{w_1^A}$ $E_{w_2^A}$ $E_{w_3^A}$ $E_{[SEP]}$ $E_{w_1^B}$ $E_{w_2^B}$ $E_{w_3^B}$ $E_{[SEP]}$
Input
[CLS] w_1^A w_2^A w_3^A [SEP] w_1^B w_2^B w_3^B [SEP]

Fig. 1. BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings, and the position embeddings.

2.1.2. Pre-trained Language Models

With the development of hardware resources, training large models on large-scale datasets have gradually become possible. Researchers have found that training models (such as Transformer) on large-scale datasets by general language modeling objectives can greatly improve the models' text representation capability. This technique is called pre-trained language models.

Once proposed, pre-trained language models have been widely applied to various natural language processing (NLP) tasks and achieved outstanding performance. In general, these models are first *pre-trained* on large-scale corpora, and then *fine-tuned* on downstream datasets for specific tasks. By pre-training, the model can learn effective language representations, thus further improving the performance on downstream tasks. In the early stage, researchers explored pre-training Transformer [**186**] on natural language corpus for language modeling. The great performance reveals the potential of pre-training. Typical works include GPT, GPT-2 [**151**], and the later GPT-3/3.5 [**12, 133**]. Thereafter, researchers found that the standard conditional language models can only capture the information in a left-to-right or right-to-left manner, while the semantic information of the context is neglected, thus leading to sub-optimal performance on language understanding tasks. To tackle this problem, BERT [**34**], *i.e.*, the bidirectional Transformer encoder, is proposed. By training with the Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives, the model can better capture contextual information from both sides, thus achieving excellent performance on many tasks. Following BERT, many studies designs various pre-training objectives for different tasks [**96, 107, 152**].

Since BERT is employed in our studies, we introduce its structure and general usage in more detail.

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a recently proposed language representation model [**34**]. BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right contexts. BERT's architecture is a multi-layer Transformer encoder. With the self-attention mechanism, the representation of a token can attend to all other tokens, thus achieving information aggregation from context in both sides (left and right). BERT is pre-trained on BooksCorpus [**243**] and English Wikipedia with the training objectives of Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In the MLM task, BERT is trained to predict the randomly masked token `[MASK]` based on its context tokens. By this means, BERT can learn the relationship between tokens. The NSP task asks the model to predict whether two sentences are consecutive in a document, which can improve the model's capability of modeling relationships between sentences.

The input representation of BERT is shown in Figure 1. For each token, its representation is the sum of three embeddings: the token embedding, the segmentation embedding, and the position embedding. The token embedding maps each token into a vector through an embedding table (randomly initialized). The segmentation embedding indicates the token belongs to the first or the second sentence (explained later). The position embedding reflects the token's position (*e.g.*, the position of the `[CLS]` token is zero).

BERT is often used to compute a representation for a single sentence or a sentence pair. Formally, when representing a single sentence $A = [w_1^A, w_2^A, \cdots, w_m^A]$ with m tokens, it first constructs the input sequence as:

$$S_{\text{A}} = \texttt{[CLS]}\, w_1^A w_2^A \cdots w_m^A \,\texttt{[SEP]}, \tag{2.1.11}$$

where `[CLS]` and `[SEP]` are two special tokens. Then, the output representation of the `[CLS]` token is used as the sentence representation. Since only a single sentence is represented, the segmentation embeddings of all tokens are the same. When representing a sentence pair (A, B), where $A = [w_1^A, w_2^A, \cdots, w_m^A]$ and $B = [w_1^B, w_2^B, \cdots, w_n^B]$, it constructs the input sequence as:

$$S_{\text{AB}} = \texttt{[CLS]}\, w_1^A w_2^A \cdots w_m^A \,\texttt{[SEP]}\, w_1^B w_2^B \cdots w_n^B \,\texttt{[SEP]}, \tag{2.1.12}$$

where the `[SEP]` token is used to separate two sentences. As shown in Figure 1, to indicate different sentences, the segment embeddings are set as $\mathbf{E}_{\text{A}}$ and $\mathbf{E}_{\text{B}}$ respectively for sentence A and sentence B. Similarly, the `[CLS]` token's representation is used as the representation of the sentence pair. Intuitively, each token in the sequence can attend to all other tokens in both sentences, so the interaction between the two sentences can be captured.

2.2. Document Ranking

A Modern information retrieval system usually contains two essential components: document retrieval and document (re-)ranking. The former aims at retrieving many relevant documents as a candidate list from a large repository, while the latter focuses on (re-)ranking the documents in the candidate list according to their relevance. In this thesis, we concentrate on the document ranking task, where the candidate documents are provided by upstream retrieval systems (such as BM25 [**157**] or dense retrievers [**85, 212**]).

In document ranking, according to whether search context information is used, we can categorize the task into ad-hoc ranking (no search context is used) and context-aware ranking. Below, we review some representative studies in both categories. All of them are based on neural networks, which are the most relevant to our study.

2.2.1. Ad-hoc Ranking

Based on different assumptions for relevance evaluation, existing neural ranking models can be divided into two categories, namely representation-focused architecture and interaction-focused architecture [**62**]. Besides these two categories, there are also some hybrid models that combine both architectures in learning relevance features.

The assumption of representation-focused architecture is that relevance relies on compositional meaning of the input texts. Therefore, models based on this architecture usually design a complex representation function for both queries and documents to obtain their high-level representations. The final relevance score is computed based on these representations through an evaluation function. Typical methods include DSSM [**75**], ARC-I [**72**], and MV-LSTM [**190**]. DSSM applied two fully-connected networks to represent queries and documents; ARC-I used 1D convolutional layers and max-pooling layers to produce high-level representation for queries and documents; and MV-LSTM employed a bi-directional LSTM for encoding queries and documents. This architecture is more suitable for short texts (since it is difficult to obtain good representation for long texts) and more efficient for online computing (because it can pre-calculate representations of the texts offline).

In contrast, the assumption of interaction-focused architecture is that relevance is essentially about the relation between queries and documents, so it is more effective to learn the interactions directly. Models based on this architecture often focus on designing complex interaction function and evaluation function to produce the relevance score. For example, KNRM [**211**] constructed a matching matrix by computing word-level cosine similarity between the query and document. Then, a group of kernels was applied to convert the word-level interactions to query-document ranking features. These ranking features were combined by a ranking layer to produce the final ranking score. ARC-II [**72**] applied 1D convolutional layers to compute the interactions between the query and document. Then, 2D convolutional layers and max-pooling layers were employed to extract features and a multi-layer perceptron was used to compute the final matching score. MatchPyramid [**134**] directly constructed an interaction matrix based on word embeddings and also applied 2D convolutional and max-pooling layers to extract matching features. Some BERT-based models [**218**] can also be treated as interaction-focused because the query and document interaction is achieved by the self-attention. This architecture generally performs better than the representation-focused architecture since more detailed interaction signals can be captured. However, the interaction-focused architecture is inefficient for online computing as the interaction cannot be pre-computed.

In order to take advantage of both representation-focused methods and interaction-focused methods, some hybrid models are proposed. For example, DUET [**126**] employed both representation-focused architecture and interaction-focused architecture as two sub-models, and used a sum operation to combine the scores from both networks to produce the final relevance score. ILM [**129**] further combined the deep learning-based neural features and traditional features. It obtained representation-focused features by encoding the query and document respectively and concatenating them together; it computed interaction-focused features by applying CNNs over the interaction matrix; and it also extracted some non-neural features (such as BM25 score and TF-IDF score). All three kinds of features were finally combined together to compute the final ranking score.

2.2.2. Context-aware Ranking

With the recent development, search engines can support more and more complex information needs. Users' search behaviors have evolved from one-shot query to multiple interactions with the search engine. For example, when users have ambiguous information needs, they will first search for some general information and gradually refine their queries to be more specific. These consecutive queries form a search session. Intuitively, the evolution of user queries reflect the change in user search intents, and the corresponding clicking behaviors reveal how their information needs might be satisfied. Therefore, to model the user intent of the current query, the context information in search sessions (including previous queries and the corresponding clicked documents) has shown to be very useful [**9, 50, 83, 239**]. Compared to ad-hoc ranking that only uses the current query, context-aware ranking should also exploit the search context, which is a more complex problem. Researchers have explored leveraging context and search activities in different forms to build better models for search query or document ranking.

Early studies focused on deriving contextual features from users' search activities to characterize their search intent and rank documents. For example, some keywords were extracted from users' historical queries and clicked documents and used to expand the current query [**165**]. Statistical features and rule-based features were also introduced to quantify context information from user search behavior [**198, 208**]. However, these methods often rely on manually extracted features or handcrafted rules, which heavily limits their applications to different datasets. Later, researchers started to build predictive models for users' search intent or future behavior. For example, a hidden Markov model was employed to model the evolution of users' search intent, based on which the document ranking is conducted [**14**].

Encouraged by the development of neural networks, various approaches have been proposed for context-aware document ranking. For example, Ahmad et al. [**3**] proposed a hierarchical neural structure with RNNs to model historical queries, and the output was combined with the current query's representation for document ranking. They have also found that jointly learning query suggestion and document ranking can boost the model's performance on both tasks. Based on this structure, in addition to leveraging historical queries, researchers introduced historical clicked documents (modeled together with the queries by the hierarchical RNNs) and found they are also helpful for document ranking [**4**].

More recently, large-scale pre-trained language models, such as BERT [**34**], have achieved great success in many NLP and IR tasks [**86, 87, 106, 113**]. Qu et al. [**149**] proposed to concatenate all historical queries, clicked documents, and unclicked documents as a long sequence and leveraged BERT as an encoder to compute their term-level representations. These representations were further combined with relative position embeddings and human behavior embeddings through another transformer-based structure to get the final representations. The ranking score is computed based on the representation of the special `[CLS]` token. Chen et al. [**21**] designed three generative tasks, *i.e.*, predicting future queries, clicked documents, and a supplementary query, to enhance the user behavior sequence modeling. They applied BART [**96**] that has a BERT-like Transformer encoder for document ranking, and a Transformer decoder for those generative tasks.

Different from existing studies, we do not devise new model architectures, but focus on improving the training paradigm for context-aware document ranking. In our first article, we propose a contrastive learning based-method to simulate potential variations in human search behavior. It greatly enhances the robustness of user behavior sequence representation. Then, in our second article, we design a curriculum learning framework to regulate the order of the training samples into an easy-to-hard manner, with which existing models can be improved significantly.

2.3. Dialogue Systems

According to the application domains, dialogue systems can be generally categorized into domain-specific systems and open-domain systems. The former aims at handling specific tasks, such as booking tickets, thus being also known as task-oriented systems. This kind of systems have been widely applied in real scenarios to automate the completion of the tasks, thus reduce the cost of human resources. The later seeks to communicate with people in a chit-chat style, so it is often referred to as a chatbot. As the purpose of the two kinds of systems are different, their implementations are different as well. For task-oriented systems, the domain and task are determined, so the overall process of the conversation can be predefined. The system only needs to fulfill the given steps to complete a specific task. The core problem in such a system is to recognize the information from user input and track the state of the conversation by filling the slots, while the response itself can be directly generated from a set of templates. In contrast, open-domain conversation systems cannot identify user intent in advance, so the key problem is understanding the user's input and constructing a proper response. In this thesis, we focus on retrieval-based dialogue, which aims at retrieving an appropriate response from a large dialogue repository. This approach is widely applied in some open-domain dialogue systems such as XiaoIce. We will study the problem of response selection. Besides, as we consider external knowledge as additional context and investigate their impact on response selection performance, we also review some knowledge-grounded dialogue systems in this section.

2.3.1. Open-domain Dialogue Systems

Inspired by the Turing test proposed in 1950 [**184**], researchers and engineers have developed many dialogue systems for chitchat [**29, 195**]. ELIZA, created in 1966, is perhaps the first chatbot known publicly [**195**]. It can chat with people in a specific domain based on many hand-crafted scripts and limited domain knowledge. Recently, with large-scale human dialogue data, researchers begin to explore data-driven approaches to build a chatbot. Existing methods can be categorized into two groups. The first is generation-based methods. The model is trained to learn how to generate a response from scratch (introduced in Section 2.3.1.1). After training, it can generate a response to any user input. However, because all responses are generated by neural networks, grammatical correctness and fluency cannot be guaranteed.[1] The second group of methods are retrieval-based. Several response candidates are retrieved by an IR system from a large repository, and the model is designed to rerank them by their relevance with user input (introduced in Section 2.3.1.2). Since all responses are human-written, they are usually fluent and correct in grammar. However, the system may not make a suitable response when the size of the repository is limited. So, the retrieval-based dialogue is applicable only when a large repository of dialogues is available. This thesis assumes that we have such a large repository. We focus on context-aware ranking problem, which is widely studied in retrieval-based approaches.

We will introduce some representative studies in both categories in following sections.

[1] This was generally the case before the arrival of ChatGPT, which demonstrates a great capacity to generate fluent sentences.

2.3.1.1. Generation-based Methods. Some early studies treat the response generation task as a statistical machine translation (SMT) problem, which enjoys the advantage of its end-to-end and data-driven features [**156, 187**]. Compared to earlier work that heavily relied on hand-crafted rules [**95, 222**] or generated rules from data [**131**], the SMT model resulted in a paradigm change [**156**]. Thereafter, Sordoni et al. [**170**] extended the neural language model in SMT architecture and proposed a sequence-to-sequence (seq2seq) model to rescore the generated outputs. However, SMT-based models do not use a proper representation of the dialogue context. It only concatenates one or a few previous utterances with the current one as the input. More recently, more sophisticated seq2seq methods have been applied to response generation and achieve better performance [**98, 162, 163, 187**]. In these seq2seq models, an encoder encodes the dialogue context (user input) into a vector representation, based on which a decoder generates the response.

Formally, given an input context $X = (x_1, \cdots, x_{L_X})$, the encoder represents it as a vector. Then, the decoder decodes the response $Y = (y_1, \cdots, y_{L_Y})$ based on the representation. The generation probability of the response Y can be defined as:

$$p(Y|X) = \prod_{t=1}^{L_Y} p(y_t|y_{<t}, X), \tag{2.3.1}$$

where y_t denotes the word generated at the t-th step, and $y_{<t}$ denotes the previous generated words $(y_1, \cdots, y_{t-1})$. Both the encoder and decoder are often implemented as recurrent neural networks (RNNs). The definition we give above is often known as single-turn response generation, because the previous turns between two interlocutors are not considered. Indeed, by concatenating previous utterances in the dialogue context with the current one as input, the methods proposed for single-turn response generation can be easily extended to multi-turn response generation.

Based on the seq2seq structure with attention mechanism [**164, 187**], multiple extensions have been made to tackle the "safe response" problem [**98**] (*i.e.*, to respond by general utterances such as "OK"); to incorporate external knowledge [**209, 234**]; to generate responses with emotions or personas [**100, 145, 233**]; to model the hierarchical structure of the dialogue context [**162, 210**]; and to reduce the cost in response decoding [**206**].

Recently, pre-trained language models have shown their power in many NLP tasks. They are also applicable for building open-domain dialogue systems. In response generation, DialoGPT [**229**] applied the multi-layer Transformer and is trained on large-scale dialogue pairs extracted from Reddit. With a large amount of training data to fine-tune the parameters, it outperformed traditional non-pre-trained methods significantly.

2.3.1.2. Retrieval-based Methods. The key issue of retrieval-based methods is how to measure the suitability of a candidate response to a user input. Formally, suppose that we have a dataset $\mathcal{D}$, in which each sample is represented as (c, r, y), where $c = \{u_1, \ldots, u_n\}$

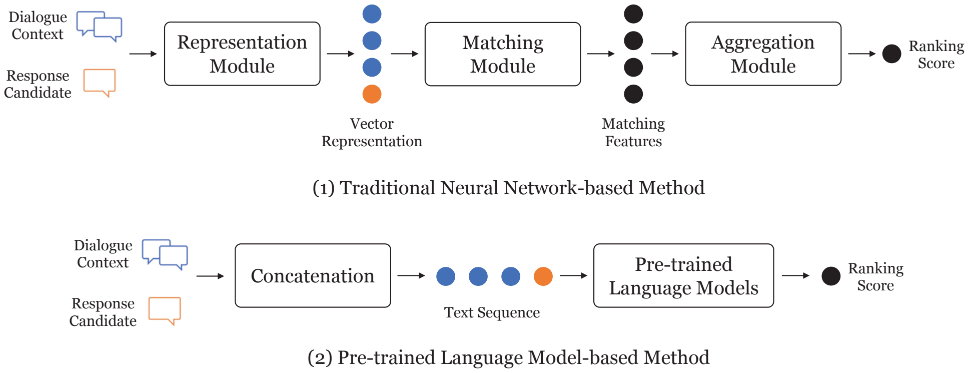


Fig. 2. Illustration of two typical methods for building retrieval-based dialogue systems.

represents a conversation context with $\{u_i\}_{i=1}^n$ as utterances; r is a response candidate; and $y \in \{0,1\}$ is a binary label, indicating whether r is a proper response. The goal is to learn a matching model g from $\mathcal{D}$, such that for a new context-response pair (c,r), $g(c,r)$ measures the degree of suitability of a response r to the given context c. Early work studied single-turn response selection where only the last utterance u_n is used, while recent work paid more attention to context-response matching for multi-turn response selection.

Sequential matching network (SMN) [**205**] is the first model for multi-turn response selection. It proposed a *representation-matching-aggregation* framework (as shown in the upper side of Figure 2), which has been further extended by many studies. In this framework, the context and response were initially encoded in vector space through a pre-trained embedding table (such as word2vec [**122**] or GloVe [**140**]). Then, a matching matrix was computed between each utterance and the response candidate by cosine similarity or dot product. Next, CNNs or RNNs were applied to extract matching features from the matching matrices. Finally, all matching features were aggregated by another RNN. Several studies extended this framework by introducing new neural architectures. For instance, deep attention matching network (DAM) [**237**] applied self-attention and cross-attention mechanism to facilitate the representation of the context and response, and allow for interactions between them. By stacking multiple attention layers, more fine-grained features can be extracted so that the final performance is improved. Multi-representation fusion network (MRFN) [**177**] proposed to represent the context and response in multiple ways, including character-based word embedding, word2vec, sequential representation based on RNNs, and local representation based on CNNs, etc. Then, matching features were extracted separately based on these different representations and then aggregated together. As different representations are expected to capture information about various aspects, the model can rely on multiple matching features for response selection. Multi-hop selector network (MSN) [**223**] devised a selection mechanism to filter out irrelevant context. The refined context is shown to be more effective in response selection.

Later, similar to generation-based methods, pre-trained language models, such as BERT [**34**] and its variants (*e.g.*, RoBERTa [**108**]), also boosted the performance on response selection over several public datasets (as shown in the lower side of Figure 2). For example, SA-BERT [**55**] demonstrated that separately modeling the utterance from different speakers can improve the performance. Different speakers were marked by type embeddings, and there was a special separation token between utterances. Such clear indicators can help pre-trained models to learn coherency within and between speakers.

2.3.2. Knowledge-Grounded Conversation

Existing open-domain dialogue systems can provide fluent responses based on dialogue context, but there is still a gap between human-machine dialogues and real human dialogues. The lack of proper knowledge is the primary reason that makes it difficult for dialogue methods to delve deeply into a topic. Large pre-trained language models have exhibited the capability of extracting implicit knowledge from a large amount of texts used to pre-train them [**12, 227, 160, 181**]. In a dialogue on a topic well covered by the texts used in the pre-training, correct answers can be provided. However, as one can also see from the recent ChatGPT model, wrong answers can be generated on a topic that is not well covered [**132**]. For example, the answer can be factually wrong, or an answer can be invented without grounding. Explicit knowledge grounding in dialogue can be an effective way to enhance the current dialogue systems.

Various approaches to knowledge-grounded conversation have been proposed. We select and introduce two categories of approaches, which are the most relevant to our study.

2.3.2.1. Document-grounded Dialogue. Leveraging grounding document is reported to be an effective way to enhance human-machine conversation [**52, 118, 226, 231**]. Zhou et al. [**235**] published a dataset in which conversations are grounded on movie-related articles from Wikipedia.[2] The interlocutors are shown with these Wikipedia articles and asked to talk about them, so that the collected conversations are grounded on these external documents (knowledge). Based on this dataset, Zhao et al. [**231**] proposed to model the context and document interactions. They devised a document-grounded matching network that lets the document and the context attend to each other so as to generate better representations for response selection. Through the attention mechanism, different parts of the document contents are assigned different weights and further participate in response selection to different extents. Gu et al. [**58**] argued that the dialogue context and document should interact with the response candidate in parallel, as opposite to being fused together. They designed a dually interactive matching network, where the document can interact with the response candidate directly.

[2] Wikipedia, `https://www.wikipedia.org`

Different from previous methods, we find that documents often contain a lot of noise. Even though one may expect the noise contents (for the current step) to be assigned with lower weights, they can still disorient response selection, especially when there is a direct keyword match. To alleviate this problem, in our third article, we propose to perform a selection on the document content before integrating it into the response selection process. Two concurrent works with ours have adopted similar ideas [**57, 73**].

2.3.2.2. Methods for Knowledge Selection. Selecting proper knowledge is an essential step for knowledge-grounded dialogue systems. For example, Ghazvininejad et al. [**52**] applied a memory network to store and fetch the knowledge. The knowledge is selected according to its similarity with the dialogue turn. Lian et at. [**104**] proposed a model with the knowledge selection mechanism which leverages both prior and posterior distributions over the knowledge to facilitate knowledge selection. The basic idea is that a relevant knowledge should lead to similar prior and posterior distributions, and the posterior distribution can act as weak labels for learning the prior distribution. Kim et al. [**90**] extended this method to a sequential knowledge Transformer that employs a sequential latent variable model to better leverage the response information for the proper choice of the knowledge. Pre-trained language models have been applied for automatic knowledge selection. For example, Zhao et al. [**232**] proposed a model based on GPT-2. The knowledge is sequentially selected and concatenated to the context sequence so as to generate responses. Essentially, the knowledge selection is achieved by the attention mechanisms in pre-trained language models.

In our two last articles, we perform knowledge selection for different purposes. In the fourth article which tries to simulate dialogue in movie following a narrative, we conduct selection on a given narrative to determine which part of narrative should be covered by the current dialogue (script line). The model should also pay attention to which part of the narrative has been covered by previous script, which is not required in problems studied in the existing work. In the fifth article focusing on proactive dialogue, we use knowledge to drive dialogue. The selected knowledge should be not only consistent with the dialogue context, but also relevant to the current dialogue target (*i.e.*entity). We design a new heuristic to generate weak labels for pieces of knowledge according to whether they are involved in the gold standard answer. The weak labels are used to train a knowledge selection model, which greatly improves the selection accuracy.

2.4. Evaluation Metrics

Evaluation is essential to assess the quality of an approach and to compare different approaches. In this section, we introduce some metrics that are used in our experiments.

Precision@k (P@k) It measures the proportion of relevant documents up to the k-th position in the ranking list, which is defined as:

$$\text{P@}k = \sum_{i=1}^{k} \frac{r(d_i)}{k}, \tag{2.4.1}$$

where $r(d_i) \in \{0{,}1\}$ indicates the binary relevance score of the document d_i in the ranked list.

Recall@k (R@k) It measures the proportion of relevant documents that are ranked in the top k of the ranking list, which is defined as:

$$\text{R@}k = \sum_{i=1}^{k} \frac{r(d_i)}{N}, \tag{2.4.2}$$

where N is the total numbers of relevant documents.

Average Precision (AP) It considers both precision and recall by computing an average of the precision at k, k being chosen as the positions of relevant documents. The definition is as follows:

$$\text{AP} = \sum_{i:r(d_i)=1} \frac{\text{P@}i}{N}, \tag{2.4.3}$$

where N is the total number of relevant documents.

Mean Average Precision (MAP) It is defined as the mean AP over the set of evaluation topics T:

$$\text{MAP} = \sum_{t \in T} \frac{\text{AP}(t)}{|T|}. \tag{2.4.4}$$

Mean Reciprocal Rank (MRR) It considers the first relevant document's position in the ranked list, which is defined as:

$$\text{MRR} = \frac{1}{|T|} \sum_{t \in T} \frac{1}{i_t}, \tag{2.4.5}$$

where i_t is the rank of the first relevant document in the list for the topic t.

Normalized Discounted Cumulative Gain@k (NDCG@k) This metric is often used for graded relevance (vs binary relevance in the metrics presented above). The gain of a document in the ranked list is obtained by discounting its relevance score by the logarithm of its rank. The discounted gain of each document in the ranked list is accumulated to compute the discounted cumulative gain (DCG) as follows:

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{2^{r(d_i)-1}}{\log_2(i+1)}, \tag{2.4.6}$$

where $r(d_i)$ is the relevance score of the document d_i. NDCG normalize the DCG scores across topics: it divides the DCG score by the DCG obtained by the ideal ranking of the documents in the ranked list as:

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k}, \tag{2.4.7}$$

where IDCG@k is the DCG score of the ideal ranking list.

2.5. Statistical Significance

Statistical significance refers to the claim that a set of observed data are attributed to a specific cause rather than the result of chance. Given two systems A and B, we can use statistical significance testing to check whether their performance difference is due to the result of chance. There are several different significance tests, such as randomization test [**28**], Wilcoxon signed rank test [**189**], sign test [**189**], bootstrap test [**28**], and student's paired t-test [**43**]. Each of them has its own criterion and null hypothesis. Among them, t-test is the most commonly used and appropriate one [**168**].

Concretely, in a t-test, we first have a null hypothesis that the mean of the distribution of the performance difference between two systems A and B is zero, *i.e.*, the two runs are identical. Then, we perform the statistical significance test as follows (taking the case that the two runs have similar variances as an example):

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \tag{2.5.1}$$

where t is the t-value, x_1 and x_2 are the means of the two runs being compared, s^2 is the pooled standard error of the two runs, and n_1 and n_2 are the number of observations in each of the runs. Then, we can compare the t-value against the values in a critical value chart (*e.g.*, a Student's t table) to obtain a p-value, which is the probability of obtaining the observed difference in performance, under the assumption that the null hypothesis is true. A low p-value indicates a high confidence of rejecting the null hypothesis, *i.e.*, the difference between the two systems is not a result of chance.

First Article.

Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking

by

Yutao Zhu[1], Jian-Yun Nie[1], Zhicheng Dou[2], Zhengyi Ma[2],
Xinyu Zhang[3], Pan Du[1], Xiaochen Zuo[2], and Hao Jiang[3]

([1]) University of Montreal, Quebec, Canada

([2]) Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

([3]) Distributed and Parallel Software Lab, Huawei, Hangzhou, Zhejiang, China

This article was published in Proceedings of the 30th ACM International Conference on Information and Knowledge Management, pp. 2780-2791. 2021 .

The main contributions of Yutao Zhu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Jian-Yun Nie and Zhicheng Dou contributed to the model design and paper writing; Zhengyi Ma and Xiaochen Zuo helped conduct the experiments; Xinyu Zhang and Hao Jiang provided the computation resources for experiments; Pan Du contributed to the paper writing.

Résumé. Les informations contextuelles dans les sessions de recherche se sont avérées utiles pour saisir l'intention de recherche des utilisateurs. Les études existantes ont exploré les séquences de comportement des utilisateurs dans les sessions de différentes manières afin d'améliorer la suggestion de requête ou le classement des documents. Cependant, une séquence de comportement de l'utilisateur a souvent été considérée comme un signal précise et exacte reflétant le comportement de l'utilisateur. En réalité, elle est très variable: les requêtes de l'utilisateur pour la même intention peuvent varier, et il peut cliquer sur différents documents. Pour apprendre une représentation plus robuste de la séquence de comportement de l'utilisateur, nous proposons une méthode basée sur l'apprentissage contrastif, qui prend en compte les variations possibles dans les séquences de comportement de l'utilisateur. Plus précisément, nous proposons trois stratégies d'augmentation des données pour générer des variantes similaires des séquences de comportement de l'utilisateur et les contraster avec d'autres séquences. Ce faisant, le modèle est forcé d'être plus robuste en ce qui concerne les variations possibles. La représentation optimisée de la séquence est incorporée dans le classement des documents. Des expériences sur deux ensembles de données réelles de logs de requêtes montrent que le modèle proposé surpasse de manière significative les méthodes de l'état de l'art, ce qui démontre l'efficacité de notre méthode pour le classement des documents en fonction du contexte.
Mots clés : Classement des documents en fonction du contexte, apprentissage contrastif, séquence de comportement de l'utilisateur, augmentation des données

Abstract. Context information in search sessions has proven to be useful for capturing user search intent. Existing studies explored user behavior sequences in sessions in different ways to enhance query suggestion or document ranking. However, a user behavior sequence has often been viewed as a definite and exact signal reflecting a user's behavior. In reality, it is highly variable: user's queries for the same intent can vary, and different documents can be clicked. To learn a more robust representation of the user behavior sequence, we propose a method based on contrastive learning, which takes into account the possible variations in user's behavior sequences. Specifically, we propose three data augmentation strategies to generate similar variants of user behavior sequences and contrast them with other sequences. In so doing, the model is forced to be more robust regarding the possible variations. The optimized sequence representation is incorporated into document ranking. Experiments on two real query log datasets show that our proposed model outperforms the state-of-the-art methods significantly, which demonstrates the effectiveness of our method for context-aware document ranking.
Keywords: Context-aware Document Ranking, Contrastive Learning, User Behavior Sequence, Data Augmentation

Prologue

At the time we wrote this article, existing studies focused on either deigning new structures to model user search behavior or applying multi-task learning to jointly optimize context-aware document ranking and other related tasks (such as query suggestion). All of these studies directly trained models on search log data, where the observed search sessions are used as positive samples, while unobserved ones are disregarded or used as negative samples. They overlooked the fact that there are no fixed search patterns for specific information needs. All queries and clicked documents are just possible options for satisfying a search intent, while a vast number of options have not yet been recorded in the search log. Under this circumstance, models cannot learn sufficient knowledge to deal with potential variations in user behavior sequences. To deal with this problem, we are inspired by recent studies on contrastive learning, and propose several data augmentation strategies to mimic the potential variations. Each user behavior sequence is augmented by introducing minor changes, so the produced sequences are similar to the original one. Through contrastive learning to distinguish these similar sequence pairs from randomly sampled dissimilar ones, the models can generate more robust representation for user behavior sequences and further improve the performance on context-aware document ranking. It is worth noting that, though our proposed method is designed for context-aware document ranking, where the previous search behaviors of a search session serve as the context, it is a general learning method that can also be applied to other context-aware ranking problems, as long as the context can be represented by a text sequence.

1. Introduction

Search engines have evolved from one-shot searches to consecutive search interactions with users [**2**]. To fulfil complex information needs, users will issue a sequence of queries, examine and interact with some of the results. User historical behavior or interaction history in a session is known to be very useful for understanding the user's information needs and to rank documents [**9, 50, 83, 238**].

Various studies exploited user behavior data for different purposes. For example, by analyzing search logs, researchers found that a user's search history provides useful information for understanding user intent during the search sessions [**9**]. To utilize the historical user behavior in document ranking, some early work explored query expansion and learning to rank techniques [**9, 17, 64, 165**]. More recently, various neural structures have been used to model the user behavior sequence. For example, a recurrent neural network (RNN) is proposed to model the historical queries and suggest the next query [**169**]. This structure has been extended to model both the historical queries and clicked documents, leading to further improvement on document ranking [**3, 4**]. Pre-trained language models have also been exploited to encode contextual information from user behavior sequences, and they achieved promising results [**149**].

All these studies tried to learn a prediction or representation model to capture the information hidden in the sequences. However, user behavior sequences have been viewed as definite and exact sequences. That is, an observed sequence is used as a positive sample and any unseen sequence is not used or is viewed as a negative sample. This strict view does not reflect the flexible nature of user's behavior in a session. Indeed, when interacting with a search engine, users do not have a definitive interaction pattern, nor a fixed query for an information need. All these are flexible and change greatly from a user to another, and from a search context to another. Similarly, user's click behaviors are also not definitive: one can click on different documents for the same information need, and can also click on irrelevant documents. The high variation is inherent in the user's interactions with a search engine. This characteristic has not been explicitly addressed in previous studies. One typically relied on a large amount of log data, hoping that strong patterns can emerge, while accidental variations (or noise) can be discarded. This is true to some extent when we have a large amount of log data and when we are only interested in the common patterns shared by users. However, the models strictly relying on the log data cannot fully capture the nuances in user behaviors and cope with the variations. A better approach is to view the data as they are, *i.e.*, they are just samples of possible query formulations and interactions, but much more are not shown in the logs.

To tackle this problem, in this work, we propose a **data augmentation** approach to generate possible variations from a search log. More specifically, we use three strategies to mask some terms in a query or document, delete some queries or documents, or reorder the sequence. These strategies reflect some typical variations in user's behavior sequences. The generated behavior sequences can be considered similar to the observed ones. We have, therefore, automatically tagged user behavior sequences in terms of similarity, which are precious for model training. In addition, we can generate more training data from search logs, which has always been a critical issue for research in this area. Based on the augmented data, we utilize **contrastive learning** to extract what is similar and dissimilar. More specifically, the contrastive model tries to pull the similar sequences (generated variants) closer and to distinguish them from semantically unrelated ones. Compared to the existing approaches based on search logs, we expect that contrastive learning can better cope with the inherent variations and generate more robust models to deal with new behavior sequences.

Contrastive learning is implemented with a pre-trained language model BERT [**34**] through encoding a sequence and its variants into a contextualized representation with a contrastive loss. The document ranking is then learned by a linear projection on top of the optimized sequence representation. With both the original sequences and corresponding variants modeled in the representation, the final ranking function can not only address the context information thoroughly, but also learn to cope with the inherent variations, hence generating better ranking results during prediction.

We conduct experiments on two large-scale real-world search log datasets (AOL and Tiangong-ST). Experimental results show that our proposed method outperforms the existing methods (including those exploiting search logs) significantly, which demonstrates the effectiveness of our approach.

Our contributions are three-fold:

(1) We design three different data augmentation strategies to construct similar sequences of observed user behavior sequences, which modify the original sequence at term, query/document, and behavior levels.

(2) We propose a self-supervised task with a contrastive learning objective based on the augmented behavior sequences to capture what is hidden behind the sequences and their variants, and to distinguish them from other unrelated sequences.

(3) Experiments on two large-scale real-world search log datasets confirm the effectiveness of our method. This study shows that contrastive learning with automatically augmented search logs is an effective way to alleviate the shortage of log data in IR research.

2. Related Work

2.1. Exploiting Historical Log Data

Context information in sessions has shown to be useful in modeling user intent in search tasks [**9, 50, 83, 239**]. Early studies focused on extracting contextual features from users' search activities so as to characterize their search intent. For example, some keywords were extracted from users' historical queries and clicked documents and used to rerank the documents for the current query [**165**]. Statistical features and rule-based features were also introduced to quantify or characterize context information [**198, 208**]. However, these methods often rely on manually extracted features or handcrafted rules, which limits their application in different retrieval tasks.

Later, researchers started to build predictive models for users' search intent or future behavior. For example, a hidden Markov model was employed to model the evolution of users' search intent. Then, both document ranking and query suggestion were conducted based on the predicted user intent [**14**]. Reinforcement learning has also been applied to model user interactions in search tasks [**59, 112**]. Unfortunately, the predefined model space or state transition structure limits the learning of rich user-system interactions.

The development of neural networks generated various solutions for context-aware document ranking. Some researchers proposed a hierarchical neural structure with RNNs to model historical queries and suggest the next query [**169**]. This model is further extended with the attention mechanism to better represent sessions and capture user-level search behavior [**24**]. Recently, researchers found that jointly learning query suggestion and document ranking can boost the model's performance on both tasks [**3**]. In addition to leveraging historical queries, the historical clicked documents are also reported to be helpful in both query suggestion and document ranking [**4**].

More recently, large-scale pretrained language models, such as BERT [**34**], have achieved great performance on many NLP and IR tasks [**86, 87, 106, 113**]. Qu et al. [**149**] proposed to concatenate all historical queries, clicked documents, and unclicked documents as a long sequence and leveraged BERT as an encoder to compute their term-level representations. These representations were further combined with relative position embeddings and human behavior embeddings through another transformer-based structure to get the final representations. The ranking score is computed based on the representation of the special "[CLS]" token.

Our framework is also based on BERT, but we use contrastive learning to pretrain the model in a self-supervised manner. Theoretically, this strategy better leverages the available training data, which can also be applied to existing methods.

2.2. Contrastive Learning for IR

Contrastive learning aims to learn effective representation of data by pulling semantically close neighbors together and pushing apart other non-neighbors [**66, 193**]. It has been widely applied in computer vision [**23, 179, 252**] and NLP tasks [**40, 49, 61, 207**] and has proven its high efficiency in leveraging the training data without the need of annotation. What is required in contrastive learning is to identify semantically close neighbors. In visual representation, neighbors are commonly generated by two random transformations of the same image (such as flipping, cropping, rotation, and distortion) [**23, 35**]. Similarly, in text representation, data augmentation techniques such as word deletion, reordering, and substitution are applied to derive similar texts from a given text sequence [**121, 207**]. Although the principle of contrastive learning is well accepted, the ways to implement it are still under exploration, with the general guiding principles of *alignment* and *uniformity* [**193**].

As for pre-training, Chang et al. [**19**] designed several paragraph-level pre-training tasks and the Transformer models can improve over the widely-used BM25 [**157**]. Ma et al. [**113**] constructed a representative word prediction (ROP) task for pre-training BERT. Experimental results showed that the BERT model pre-trained with ROP and masked language model (MLM) tasks achieves great performance on ad-hoc retrieval. Our proposed sequence representation optimization stage can be treated as a pre-training stage because it is trained before document ranking (our main task). However, as we do not use external datasets, we do not categorize our method as a pre-training approach.

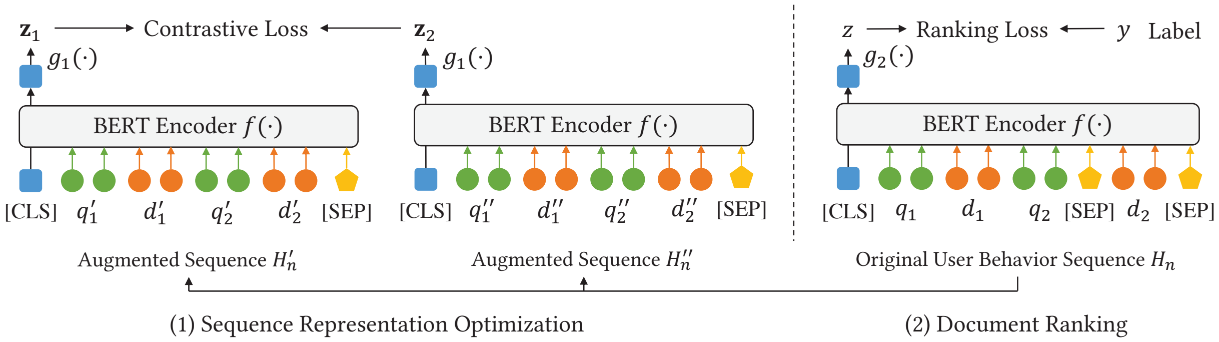

$\mathbf{z}_1$
Contrastive Loss
$\mathbf{z}_2$
$g_1(\cdot)$
$g_1(\cdot)$
BERT Encoder $f(\cdot)$
BERT Encoder $f(\cdot)$
[CLS] q_1' d_1' q_2' d_2' [SEP]
[CLS] q_1'' d_1'' q_2'' d_2'' [SEP]
Augmented Sequence H_n'
Augmented Sequence H_n''
z
Ranking Loss
y Label
$g_2(\cdot)$
BERT Encoder $f(\cdot)$
[CLS] q_1 d_1 q_2 [SEP] d_2 [SEP]
Original User Behavior Sequence H_n
(1) Sequence Representation Optimization
(2) Document Ranking

Fig. 3. The illustration of COCA. The query-document sequence (H_n) is augmented with two different strategies, and the processed sequences are treated as a positive pair (H_n' and H_n''). Other augmented sequences in the same minibatch are used to construct negative pairs for them (not shown here). The contrastive objective is to pull close the representation of the two sequences in positive pairs and push apart the representation of others.

In this work, we propose a contrastive learning objective for optimizing the sequence representation in order to improve the downstream document ranking task. This first attempt paves the way for future research on the use of contrastive learning to IR.

3. Methodology

Context-aware document ranking aims at using the historical user behavior sequence and the current query to rank a set of candidate documents. In this work, we design a new framework for this task. Our framework aims at optimizing the representation of the user behavior sequence before learning document ranking. As shown in Figure 3, our framework can be divided into two stages: (1) *sequence representation optimization* and (2) *document ranking*. In the first stage, we design a self-supervised task with contrastive learning objective to optimize the sequence representation. In the second stage, our model uses the optimized sequence representation and further learns the ranking model. We call our framework COCA – **CO**ntrastive learning for **C**ontext-**A**ware document ranking.

3.1. Notations

Before introducing the task and the model, we first provide the definitions of important concepts and notations. We present a user's search history as a sequence of M queries $Q = \{q_1, \cdots, q_M\}$, where each query q_i is associated with a submission timestamp t_i and the corresponding list of returned documents $D_i = \{d_{i,1}, \cdots, d_{i,M}\}$. Each query q_i is represented as the original text string that users submitted to the search engine. Q is ordered according to query timestamp t_i. Each document $d_{i,j}$ has two attributes: its text content and click label $y_{i,j}$ ($y_{i,j} = 1$ if it is clicked). In general, user clicks serve as a good proxy of relevance feedback [**4, 80, 81, 149**]. Given all available historical queries and clicked documents up to n turns, we denote the user behavior sequence as $H_n = \{q_1,d_1, \cdots ,q_n,d_n\}$.[3] As reported in [**4, 149**], the unclicked documents are less helpful and may even introduce noise, so they are not considered in the user behavior sequence.

3.2. Overview

With the above concepts and notations, we briefly introduce the two stages in COCA as follows.

(1) **Sequence Representation Optimization**. As shown in the left side of Figure 3, our target is to obtain a better representation of the user behavior sequence H_n in this stage. To achieve this, we first construct two augmented sequences H'_n and H''_n from H_n with randomly selected augmentation strategies (Section 3.3.1). Such a pair of sequences are considered to be similar. Then a BERT encoder is applied to get the representations of these two sequences (Section 3.3.2). With the contrastive loss, the model learns to pull them close and push them away from other sequences in the same minibatch (Section 3.3.3). By comparing the two augmented sequences, the BERT encoder is forced to learn a more generalized and robust representation for sequences.

(2) **Document Ranking**. As shown in the right side of Figure 3, we aim to rank the relevant documents as high as possible in this stage. Given the current query q_i and the historical behavior sequence H_{i-1}, we treat $H_{i-1} \cup \{q_i\}$ as a sequence and the candidate document $d_{i,j}$ as another sequence. Then, we concatenate them together and use the BERT encoder trained in the first stage to generate a representation. The final ranking score is obtained by a linear projection on the representation. A cross-entropy loss is applied between the predicted ranking score and the click label $y_{i,j}$.

3.3. Sequence Representation Optimization

The user behavior sequence contains abundant information about the user intent. To optimize the representation of the user behavior sequence, we propose a self-supervised approach. Specifically, we apply a contrastive learning objective to pull close the representation of similar sequences and push apart different ones. The similar sequences are created by the three augmentation strategies described below.

3.3.1. Augmentation Strategy. Inspired by the existing data augmentation strategies in NLP and image processing, we propose three strategies to construct similar sequences, namely term mask, query/document deletion, and behavior reordering (shown in Figure 4). These strategies correspond to three levels of variation in user behaviors, *i.e.*, term level, query/document level, and user behavior level.

[3] Following previous studies [**149**], we only use one clicked document to construct the sequence.

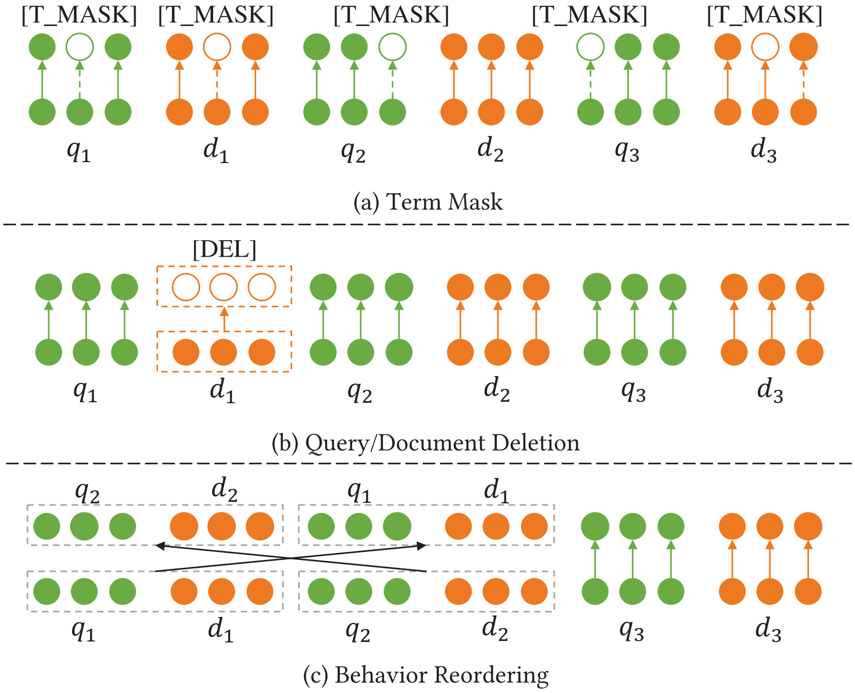


Fig. 4. Three augmentation strategies used in COCA. We use the user behavior sequence with three query-document pairs as an example.

(a) **Term Mask**. In natural language processing, the "word mask" or "word dropout" technique has been widely applied to avoid overfitting. It has been shown to improve the robustness of the sentence representation, *e.g.*, in sentence generation [**10**], sentiment analysis [**30**], and question answering [**45**]. Inspired by this, we propose to apply a random term mask operation over the user behavior sequence (including query terms and document terms) as one of the augmentation strategies for contrastive learning.

With the term-level augmentation strategy, we can obtain various user behavior sequences similar to the original one. The similar sequences only have minor differences in some terms. This aims to simulate the real search situations where users may issue slightly different queries for searching the same target, and a document may satisfy similar information needs. By contrasting similar sequences with others, the models can learn the importance of different terms in both queries and documents. Besides, it can also help the model to learn more generalized sequence representation by avoiding relying too much on specific terms.

Specifically, for a user behavior sequence $H_n = \{q_1,d_1,\cdots,q_n,d_n\}$, we first represent it as a term sequence $H_n = \{w_1,\cdots,w_{N_T}\}$, where N_T is the total number of terms. Then, we randomly mask a proportion γ of terms $T_n = \{t_1,\cdots,t_L\}$, where $L_{\text{tm}} = \lfloor N \cdot \gamma \rfloor$, and t_i is the index of term to be masked. If a term is masked, it is replaced by a special token "[T_MASK]", which is similar to the token "[MASK]" used in BERT [**34**]. Therefore, we formulate this augmentation strategy as a function f^{tm} over the user behavior sequence H_n

as:

$$f^{\text{tm}}(H_n) = \{\widehat{w}_1, \cdots, \widehat{w}_N\}, \tag{3.1}$$

$$\widehat{w}_t = \begin{cases} w_t, & t \notin T_n, \\ [\text{T\_MASK}], & t \in T_n. \end{cases} \tag{3.2}$$

(b) **Query/Document Deletion**. Random crop (deletion) is a common data augmentation strategy in computer vision to increase the variety of images [**23, 179**]. This operation can create a random subset of an original image and help the model generalize better. Inspired by this, we propose a query/document deletion augmentation operation for contrastive learning.

The query/document deletion strategy can improve the learning of sequence representation in two respects. First, after deletion, the resulting user behavior sequence becomes a similar one with the difference on some queries or documents. This reflects a type of variation in real query logs. By contrasting these similar sequences with others, the models are trained to learn the influence of the deleted queries or documents. Second, the generated incomplete sequence provides a partial view of the original sequence, which forces the model to learn a more robust representation without relying on complete information.

Specifically, for a user behavior sequence $H_n = \{q_1, d_1, \cdots, q_n, d_n\}$, we treat each query and document as a sub-sequence s and represent the sequence as $H_n = \{s_1, s_2, \cdots, s_{2n-1}, s_{2n}\}$. Then, we randomly delete a proportion μ of sub-sequences $R_n = \{r_1, \cdots, r_L\}$, where $L_{\text{del}} = \lfloor 2n \cdot \mu \rfloor$, and r_i is the index of the sub-sequence to be deleted. Different from the term mask strategy, if a query or document is deleted, the whole sub-sequence is replaced by a special token "[DEL]". This augmentation strategy is formulated as a function f^{del} on H_n and defined as:

$$f^{\text{del}}(H_n) = \{\widehat{s}_1, \cdots, \widehat{s}_{2n}\}, \tag{3.3}$$

$$\widehat{s}_r = \begin{cases} s_r, & r \notin R_n, \\ [\text{DEL}], & r \in R_n. \end{cases} \tag{3.4}$$

(c) **Behavior Reordering**. Many tasks assume the strict order of the sequence, *e.g.*, natural language generation [**89, 186**] and text coherence modeling [**101, 102, 251**]. However, we observe that the user search behavior sequence is much more flexible. For example, when users only have a vague search intent, they will issue several queries in a random order to obtain related information before making their real intent clear [**59**]. Besides, sometimes users may issue a repeated query when they miss some information, which is called re-finding behavior [**114, 240**]. Under this circumstance, we cannot assume the order of the queries is strict. To prevent the model from relying too much on the order information and make the model more robust to the newly issued query, we propose a behavior reordering strategy for contrastive learning. Different from the former two strategies, user behavior reordering does not reduce the information contained in the sequence. Models can focus on learning content representation in queries and documents rather than merely "remembering" their relative order.

For a user behavior sequence $H_n = \{q_1,d_1,\cdots,q_n,d_n\}$, we treat each query and its corresponding document as a behavior sub-sequence and denote it as $H_n = \{b_1,\cdots,b_n\}$, where $b_i = \{q_i,d_i\}$. Then, we randomly select two behavior sub-sequences and switch their positions, and this operation is conducted $L_{\text{br}} = \lfloor n \cdot \eta \rfloor$ times, where η is the reordering ratio. Considering the randomly selected i-th pairwise position as (u_i, v_i), we switch b_{u_i} and b_{v_i}, which can be formulated as a function f^{br} on H_n:

$$f^{br}(H_n) = \{\widehat{b}_1,\cdots,\widehat{b}_n\} \tag{3.5}$$

$$\widehat{b}_j = \begin{cases} b_j, & j \neq u_i \text{ and } j \neq v_i, \\ b_{v_i}, & j = u_i, \\ b_{u_i}, & j = v_i. \end{cases} \tag{3.6}$$

3.3.2. Representation. Previous work has shown the effectiveness of applying BERT [**34**] for sequence representation [**31, 130, 149, 175, 251**]. In our framework, we also use the pre-trained BERT as an encoder to represent the augmented user behavior sequences (shown in the left side of Figure 3). For a user behavior sequence $H_n = \{q_1,d_1,\cdots,q_n,d_n\}$, following the design of vanilla BERT, we add special tokens "[CLS]" and "[SEP]" at the head and tail of the sequence, respectively. Besides, to further indicate the end of each query/document, we append a special token "[EOS]" at the end of it. Therefore, the input sequence X is represented as:

$$X = \text{[CLS]}q_1\text{[EOS]}d_1\text{[EOS]}\cdots q_n\text{[EOS]}d_n\text{[EOS][SEP]}. \tag{3.7}$$

Then, the embedding of each token, the positional embedding, and the segment embedding are added together and input to BERT to obtain the contextualized representation.[4] The output of BERT is a sequence of representations for all tokens, and we use the representation of "[CLS]" token as the sequence representation:

$$\mathbf{z} = g_1\Big(\text{BERT}(X)_{\text{[CLS]}}\Big), \tag{3.8}$$

where $\mathbf{z} \in \mathbb{R}^{768}$, and $g_1(\cdot)$ is a linear projection.

3.3.3. Training Objective. We apply a contrastive learning objective to optimize the user behavior sequence representation. A contrastive learning loss function is defined for the contrastive prediction task, *i.e.*, trying to predict the positive augmentation pair (H_i, H_j) in set $\{\mathcal{H}\}$. We construct the set $\{\mathcal{H}\}$ by randomly augmenting twice for all sequences in a

[4] Please refer to the original paper of BERT [**34**] for more details about these embeddings.

minibatch. The strategy of each augmentation is randomly selected from our proposed three ones. Assuming a minibatch with size N, we can obtain a set $\{\mathcal{H}\}$ with size $2N$. The two augmented sequences from the *same* user behavior sequence form the positive pair, while all other sequences from the same minibatch are regarded as negative samples for them. Following previous work [**23, 49, 207, 40**], the contrastive learning loss for a positive pair is defined as:

$$l(i,j) = -\log \frac{\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_k)/\tau)}, \tag{3.9}$$

where $\mathbb{1}_{k \neq i}$ is the indicator function to judge whether $k \neq i$ and τ is a hyperparameter representing temperature. The overall contrastive learning loss is defined as all positive pairs' losses in a minibatch:

$$\mathcal{L}_{\text{CL}} = \sum_{i=1}^{2N} \sum_{j=1}^{2N} m(i,j) l(i,j), \tag{3.10}$$

where $m(i,j) = 1$ when (H_i, H_j) is a positive pair, and $m(i,j) = 0$ otherwise.

From another perspective, the contrastive learning stage can be viewed as a kind of domain-specific post-training for pre-trained language models. As these contextualized language models are usually pre-trained on general corpora, such as the Toronto Books Corpus and Wikipedia, it is less effective to directly fine-tune these models on our downstream ranking task if there is a domain shift. Our contrastive learning stage can help the model on domain adaptation to further improve the ranking task. This strategy has shown to be effective in various tasks including reading comprehension [**214**] and dialogue generation [**54, 196**].

3.4. Context-aware Document Ranking

In the previous step, the BERT encoder has been optimized with the contrastive learning objective. We now incorporate this BERT encoding to learn the context-aware document ranking task.

3.4.1. Representation. Previous studies have applied BERT for ranking in a manner of sequence pair classification [**31, 130, 149**]. Different from the first stage, the ranking stage aims at measuring the relationship between the historical user behavior sequence $H_{n-1} = \{q_1,d_1, \cdots ,q_{n-1},d_{n-1}\}$, the current query q_n, and a candidate document $d_{n,i}$. Therefore, we treat $H_{n-1} \cup q_n$ as one sequence and $d_{n,i}$ as another sequence, and the input sequence Y is represented as:

$$Y = [\text{CLS}]q_1[\text{EOS}]d_1[\text{EOS}] \cdots q_n[\text{EOS}][\text{SEP}]d_{n,i}[\text{EOS}][\text{SEP}]. \tag{3.11}$$

Afterwards, the embedding of each token, the positional embedding, and the segment embedding are added together and input to BERT. Note that Y contains two sequences, so we set their segment embeddings respectively as 0 and 1 to distinguish them. The output representation of "[CLS]" is used as the sequence representation to calculate the ranking score z as:

$$\mathbf{h} = \mathrm{BERT}(Y)_{[\mathrm{CLS}]}, \quad z = g_2(\mathbf{h}), \tag{3.12}$$

where $\mathbf{h} \in \mathbb{R}^{768}$, and $g_2(\cdot)$ is a linear projection to map the representation into a (scalar) score.

3.4.2. Optimization. Following previous studies [**4, 149**], we use the following cross-entropy loss to optimize the model:

$$\mathcal{L}_{\mathrm{rank}} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log z_i + (1 - y_i)\log(1 - z_i), \tag{3.13}$$

where N is the number of samples in the training set.

4. Experiments

4.1. Datasets and Evaluation Metrics

We conduct experiments on two public datasets: AOL search log data [**135**] and Tiangong-ST query log data [**22**].[5]

For AOL search log, we use the one provided by Ahmad et al. [**4**]. The dataset contains a large number of sessions, and each session consists of several queries. In training and validation sets, there are five candidate documents for each query in the session. In the test set, 50 documents retrieved by BM25 [**157**] are used as candidates for each query in the session. All queries have at least one satisfied click in this dataset, and if there are more than one clicked documents, we use the first one in the list to construct the user behavior sequence.

Tiangong-ST dataset is collected from a Chinese commercial search engine. It contains web search session data extracted from an 18-day search log. Each query in the dataset has 10 candidate documents. In the training and validation sets, we use the clicked documents as the satisfied clicks. Some queries may have no satisfied click, we use a special token "[Empty]" for padding. For the test, the last query of each session is manually annotated with relevance scores, while other (previous) queries in the session have only click labels. Therefore, we construct two test sets based on the original test data as follows:

(1) Tiangong-ST-Click: In this test set, we only use the previous queries (*i.e.*, without the last query) and their candidate documents. Similar to AOL dataset, in this test scenario, all

[5] We understand that the AOL dataset should normally not be used in experiments. We choose to use it here because it contains real human clicks, which fits our experiments well. MS MARCO Conversational Search dataset may be another possible dataset, but the sessions in it are artificially constructed rather than real search logs. So, we do not use the MS MARCO dataset in experiments.

Table 1. The statistics of the datasets used in the paper.

AOL	**Training**	**Validation**	**Test**
# Sessions	219,748	34,090	29,369
# Queries	566,967	88,021	76,159
Avg. # Query per Session	2.58	2.58	2.59
Avg. # Document per Query	5	5	50
Avg. Query Len	2.86	2.85	2.9
Avg. Document Len	7.27	7.29	7.08
Avg. # Clicks per Query	1.08	1.08	1.11
Tiangong-ST	**Training**	**Validation**	**Test**
# Sessions	143,155	2,000	2,000
# Queries	344,806	5,026	6,420
Avg. # Query per Session	2.41	2.51	3.21
Avg. # Document per Query	10	10	10
Avg. Query Len	2.89	1.83	3.46
Avg. Document Len	8.25	6.99	9.18
Avg. # Clicks per Query	0.94	0.53	(3.65)

documents are labeled with “click” or “unclick”, and the model is asked to rank the clicked documents as high as possible. Note that the query with no click document is not used for testing.

(2) Tiangong-ST-Human: In this test set, only the last query with human annotated relevance score is used. The score ranges from 0 to 4. More details can be found in [**22**].

The statistics of both datasets are shown in Table 1. Following previous studies [**4, 47, 74, 76**], to reduce memory requirements and speed up training, we only use the document title as its content.

Evaluation Metrics We use Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at position k (NDCG@k, $k = \{1,3,5,10\}$) as evaluation metrics. For Tiangong-ST-Human, since the candidate documents are provided by a commercial search engine, the irrelevant documents are expected to be limited. Hence, as suggested by the authors of [**22**], we only evaluate the results with NDCG@k. All evaluation results are calculated by TREC’s evaluation tool (trec_eval) [**63**].

4.2. Baseline

We compare our method with several baseline methods, including those for (1) ad-hoc ranking and (2) context-aware ranking.

(1) **Ad-hoc ranking methods**. These methods do not use context information (historical queries and documents), and only current query is used for ranking documents.

Table 2. Experimental results on all datasets. All baseline models are based on the code released in the original paper. The best performance and the second best performance are in **bold** and underlined, respectively. The improvement of `COCA` over the best baseline is given. $\dagger$ indicates `COCA` achieves significant improvements over all existing methods in paired t-test with p-value < 0.01.

Model	**MAP**	**MRR**	**NDCG@1**	**NDCG@3**	**NDCG@5**	**NDCG@10**
			AOL			
`ARC-I`	0.3361	0.3475	0.1988	0.3108	0.3489	0.3953
`ARC-II`	0.3834	0.3951	0.2428	0.3564	0.4026	0.4486
`KNRM`	0.4038	0.4133	0.2397	0.3868	0.4322	0.4761
`Duet`	0.4008	0.4111	0.2492	0.3822	0.4246	0.4675
`M-NSRF`	0.4217	0.4326	0.2737	0.4025	0.4458	0.4886
`M-Match`	0.4459	0.4572	0.3020	0.4301	0.4697	0.5103
`CARS`	0.4297	0.4408	0.2816	0.4117	0.4542	0.4971
`HBA`	0.5281	0.5384	0.3773	0.5241	0.5624	0.5951
`COCA`	**0.5500**†	**0.5601**†	**0.4024**†	**0.5478**†	**0.5849**†	**0.6160**†
Improv.	4.15%	4.03%	6.65%	4.52%	4.00%	3.51%
			Tiangong-ST-Click			
`ARC-I`	0.6597	0.6826	0.5315	0.6383	0.6946	0.7509
`ARC-II`	0.6729	0.6954	0.5458	0.6553	0.7086	0.7608
`KNRM`	0.6551	0.6748	0.5104	0.6415	0.6949	0.7469
`Duet`	0.6745	0.7026	0.5738	0.6511	0.6955	0.7621
`M-NSRF`	0.6836	0.7065	0.5609	0.6698	0.7188	0.7691
`M-Match`	0.6778	0.6993	0.5499	0.6636	0.7199	0.7646
`CARS`	0.6909	0.7134	0.5677	0.6764	0.7271	0.7746
`HBA`	0.6957	0.7171	0.5726	0.6807	0.7292	0.7781
`COCA`	**0.7481**†	**0.7696**†	**0.6386**†	**0.7445**†	**0.7858**†	**0.8180**†
Improv.	7.53%	7.32%	11.53%	9.37%	7.76%	5.13%
			Tiangong-ST-Human			
`ARC-I`	-	-	0.7088	0.7087	0.7317	0.8691
`ARC-II`	-	-	0.7131	0.7237	0.7379	0.8732
`KNRM`	-	-	0.7473	0.7505	0.7624	0.8891
`Duet`	-	-	0.7577	0.7354	0.7548	0.8829
`M-NSRF`	-	-	0.7124	0.7308	0.7489	0.8795
`M-Match`	-	-	0.7311	0.7233	0.7427	0.8801
`CARS`	-	-	0.7385	0.7386	0.7512	0.8837
`HBA`	-	-	0.7612	0.7518	0.7639	0.8896
`COCA`	-	-	**0.7769**	**0.7576**	**0.7703**	**0.8932**
Improv.	-	-	2.06%	0.77%	0.84%	0.40%

`KNRM` [**211**] performs fine-grained interaction between current query and candidate documents and obtain a matching matrix. The ranking features and scores are then calculated by a kernel pooling method.

ARC-I [**72**] is a representation-based method. The query and document are represented by convolutional neural networks (CNNs), respectively. The score is calculated by a multi-layer perceptron (MLP).

ARC-II [**72**] is an interaction-based method. A matching map is constructed from the query and document, based on which the matching features are extracted by CNNs. The score is also computed by an MLP.

Duet [**126**] computes local and distributed representations of the query and document by several layers of CNNs and MLPs. Then, it integrates both interaction-based features and representation-based features to compute ranking scores.

(2) **Context-aware ranking methods**. These methods can leverage both context information and current query to rank candidate documents.

M-NSRF [**3**] is a multi-task model, which jointly predicts the next query and ranks corresponding documents. The historical queries in a session are encoded by a recurrent neural network (RNN). The ranking score is computed based on the query representation, history representation, and document representation.

M-Match-Tensor [**3**] is similar to M-NSRF but learns a contextual representation for each word in the queries and documents. The computation of ranking score is based on the word-level representation.

CARS [**4**] is also a multi-task model, which learns query suggestion and document ranking simultaneously. Different from M-NSRF, this method also models the click documents in the history through an RNN. An attention mechanism is applied to compute representations for each query and document. The final ranking score is computed based on the representation of historical queries, clicked documents, current query, and candidate documents.[6]

HBA-Transformer [**149**] (henceforth denoted as HBA) concatenates historical queries, clicked documents, and unclick documents into a long sequence and applies BERT [**34**] to encode them into representations. Then, a higher-level transformer structure with behavior embedding and relative position embedding is employed to further enhance the representation. Finally, the representation of the first token ("[CLS]") is used to calculate the ranking score. This is the state-of-the-art method in context-aware document ranking task. It is the most similar to our approach, but without contrastive learning.

4.3. Implement Details

We use PyTorch [**136**] and Transformers [**200**] to implement our model. The pre-trained BERT is provided by Huggingface.[7] The maximum number of tokens in the two stages

[6] We will notice some slight discrepancies between our results and those of the original paper of CARS. This is due to different tie-breaking strategies in evaluation. Following [**149**], we use trec_eval while the authors of CARS uses an author-implemented evaluation.

[7] https://huggingface.co/bert-base-uncased

are set as 128. Sequences with more than 128 tokens are truncated by popping query-document pairs from the head. We use AdamW [**110**] optimizer in both stages. In the sequence representation optimization stage, both the term mask ratio and query/document deletion ratio are tuned from 0.1 to 0.9 and set as 0.6. As for behavior reordering, only one pair of positions are switched because the session is not long (on average 2.5 queries per sessions). The three strategies are randomly selected. Note that the reordering strategy can only be applied to sessions with more than one query. The batch size is set as 128, and the temperature is set as 0.1. We train the model for four epochs. The learning rate is set as 5e-5. In the document ranking stage, we apply a dropout layer on the sequence representation with the rate of 0.1. The learning rate is set as 5e-5 and linearly decayed during the training. We train the model for three epochs. All hyperparameters are tuned based on the performance on the validation set. Our code is released on GitHub at `https://github.com/DaoD/COCA`.

4.4. Experimental Results and Analysis

The experimental results are shown in Table 2. We can find that `COCA` outperforms all existing methods. This result clearly demonstrates the superiority of our method. Based on the results, we can make the following observations.

(1) Among all models, `COCA` performs the best, demonstrating its effectiveness on modeling user behavior sequence through contrastive learning. In general, context-aware document ranking models outperform ad-hoc ranking models. For example, on the AOL dataset, the weak contextualized model `M-NSRF` can nevertheless outperform the strong ad-hoc ranking model `KNRM`. This indicates that modeling user behavior sequence is beneficial for understanding user intent and improving ranking results.

(2) Compared with RNN-based multi-task learning models (`M-NSRF`, `M-Match-Tensor`, and `CARS`), BERT-based methods (`HBA` and `COCA`) achieve better performance. In particular, on the AOL dataset, `HBA` and `COCA` improve the results by more than 15% in terms of all metrics. Notably, `HBA` and `COCA` learn document ranking independently without supervision signals from the query suggestion task. This result reflects the obvious benefit of using pre-trained language models (such as BERT) to document ranking.

(3) `HBA` is the state-of-the-art method on context-aware document ranking task. It designs sophisticated structures over a BERT encoder to evaluate user behavior from various perspectives, including an intra-behavior attention on clicked documents and skipped documents; an inter-behavior attention on all turns; and an embedding that indicates their relative positions. In comparison, our `COCA` only applies a standard BERT encoder and achieves significantly better performance (paired t-test with p-value < 0.01). Both MAP and MRR are improved by around 4%. The key difference between them is the contrastive

Table 3. Performance of COCA on the AOL dataset with different data augmentation strategies.

	MAP	MRR	NDCG@1	NDCG@3	NDCG@10
COCA (Full)	**0.5500**	**0.5601**	**0.4024**	**0.5478**	**0.6160**
None	0.5341	0.5445	0.3867	0.5296	0.5999
TM	0.5472	0.5576	0.4009	0.5444	0.6121
QDD	0.5452	0.5554	0.3969	0.5422	0.6110
TM + QDD	0.5492	0.5592	0.4005	0.5467	0.6155
TM + BR	0.5448	0.5550	0.3963	0.5414	0.6115
QDD + BR	0.5473	0.5576	0.3995	0.5444	0.6132

learning we use. The improvements of COCA over HBA demonstrates the advantage of using contrastive learning for behavior sequence representation.

(4) Intriguingly, the improvements of COCA on the AOL dataset are much more significant than those on the Tiangong-ST dataset. There are two potential reasons: (a) COCA is trained on data using click labels rather than relevance labels, and the user behavior sequence is also constructed based on click labels. Therefore, the model is better at predicting click-based scores than relevance scores. (b) According to our statistics, more than 77.4% documents are labeled as relevant (*i.e.*, their annotated relevance scores are larger than 1), hence the base score is quite high. Even the basic model ARC-I can obtain NDCG@1 and NDCG@10 values of 0.7088 and 0.8691. Without more precise relevance labels for training, it is more challenging for our model to further improve relevance ranking.

4.5. Discussion

We further investigate the following research questions.

4.5.1. Influence of Data Augmentation Strategy. To study the effectiveness of our proposed sequence augmentation strategy, we test the performance on AOL with different combinations of strategies. The results are shown in Table 3. “None” means that we use the original BERT parameters for document ranking without our proposed sequence optimization stage. We denote the term mask strategy as “TM”, query/document deletion as “QDD”, and behavior reordering as “BR”. Note that the reordering strategy can only apply to sequences with more than two query-document pairs, thus cannot work independently.

First, compared with no sequence optimization stage, optimizing sequence representation with any combination of our proposed strategies is helpful. This clearly demonstrates that our proposed method is effective in building a more robust representation. Second, the term mask works best and this single strategy can improve around 2.5% in MAP. This implies that learning user behavior sequences with similar queries and documents are very useful for document ranking. Finally, it is interesting to see that combining term mask and behavior

Table 4. Performance of `COCA` on the AOL dataset with different hyperparameters.

		CE (↓)	**Acc.**	**MAP**	**MRR**	**NDCG@1**	**NDCG@10**
Tempera. τ	0.05	**0.5662**	79.62	0.5417	0.5521	0.3947	0.6078
	0.1	0.5823	83.56	**0.5500**	**0.5601**	**0.4024**	**0.6160**
	0.3	1.6240	**84.03**	0.5451	0.5552	0.3972	0.6116
	0.5	4.3226	69.85	0.5433	0.5536	0.3950	0.6031
	1.0	5.2148	62.33	0.5417	0.5522	0.3951	0.6073
Batch Size	16	0.7289	81.14	0.5380	0.5482	0.3897	0.6044
	32	0.7226	80.92	0.5447	0.5547	0.3972	0.6108
	64	0.7210	81.20	0.5432	0.5534	0.3951	0.6089
	128	**0.5823**	**83.56**	**0.5500**	**0.5601**	**0.4024**	**0.6160**

reordering strategy (*i.e.*, "`TM + BR`") leads to a performance degradation compared with only using the term mask strategy. After checking the sequence representation optimization process, we find that the contrastive learning loss in this case is very low and the prediction accuracy is very high, which indicates that this combination is easy to overfit and cannot learn a good sequence representation.

4.5.2. Performance with Different Hyperparameters. As reported in recent work [**23**], the temperature and batch size are two important hyperparameters in contrastive learning. To investigate the impact of them, we train our model with different settings and test their performance. In addition to evaluating the performance of ranking, we also compute the loss value (cross-entropy, CE) and prediction accuracy in contrastive prediction. The results are shown in Table 4.

Considering temperature, according to Equation (3.9), a higher temperature will cause a higher loss, which are consistent with our results. However, a lower contrastive loss cannot always lead to a better performance. Indeed, $\tau = 0.1$ is the best choice for the document ranking task. Therefore, it is important to select a proper temperature for contrastive learning. Similar observations are also reported in other recent studies [**23, 49**].

As for batch size, we can see that contrastive learning benefits from larger batch sizes. According to a recent study [**23**], larger batch sizes can provide more negative examples, so that the convergence can be facilitated. Due to our limited hardware resources, the largest batch size we can handle is 128. We speculate that a larger batch size can bring more improvements.

4.5.3. Performance on Sessions with Different Lengths. To understand the impact of the session length on the final ranking performance, we categorize the sessions in the test set into three bins:

(1) Short sessions (with 1-2 queries) - 77.13% of the test set;

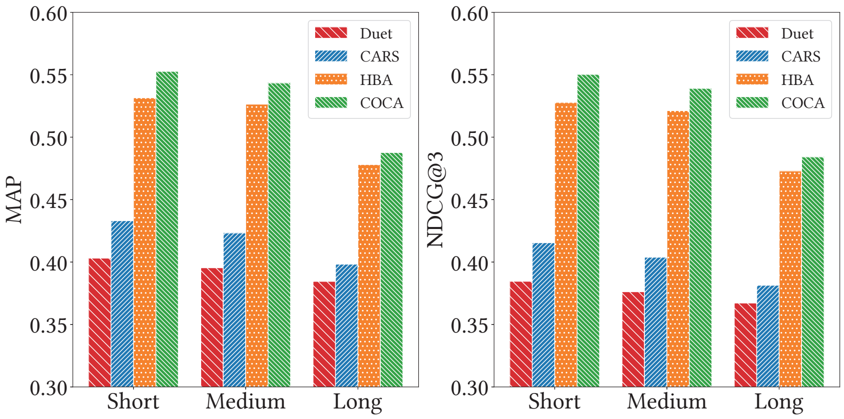


Fig. 5. Performance on different lengths of sessions.

(2) Medium sessions (with 3-4 queries) - 18.19% of the test set;

(3) Long sessions (with 5+ queries) - 4.69% of the test set.

As we also consider sessions with only one query, the short sessions have a higher proportion than that provided in [**4**].

We compare `COCA` with `Duet`, `CARS`, `HBA` on AOL dataset and show the results regarding MAP and NDCG@3 in Figure 5. First, it is evident that `COCA` outperforms all context-aware baseline methods on all three bins of sessions. This suggests `COCA`'s advantages in learning search context. Second, we can see the ad-hoc ranking method `Duet` performs worse than other context-aware ranking methods. This demonstrates once again that modeling the historical user behavior is essential for improving the document ranking performance. Third, we can observe that `COCA` performs relatively worse in long sessions than in short sessions. We hypothesize that those longer sessions are intrinsically more difficult, and similar trend in baseline methods can support this. This can be due to the fact that a long session may contain more noise or exploratory search. This is also shown by a larger improvement in the short sessions from `COCA` to the ad-hoc baseline ranker `Duet` than that in the long sessions (37.10% v.s. 26.83% in terms of MAP). This result implies that it may be useful to model the immediate search context rather than the whole context.

4.5.4. Effect of Modeling User Behavior Progression. It is important to study how the modeled search context helps document ranking when a search session progresses. We compare `COCA` with `CARS` and `HBA` at individual query positions in short (S), medium (M), and long (L) sessions. The results are reported in Figure 6. Due to the limited space, long sessions with more than seven queries are not presented.

It is noticeable that the ranking performance is improved steadily as a search session progresses, *i.e.*, more search context becomes available for predicting the next click. Both `COCA` and `HBA` benefit from it, while `COCA` improves faster by better exploiting the context. In contrast, the performance of `CARS` is unstable. This implies that BERT-based methods are

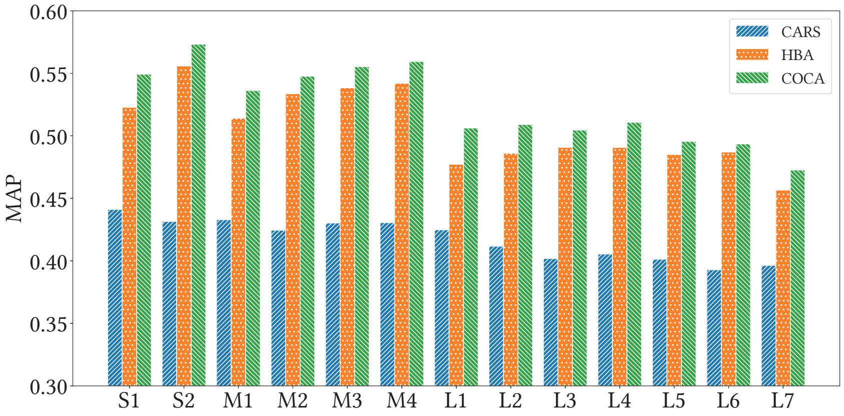


Fig. 6. Performance at different query positions in short (S1-S2), medium (M1-M4), and long sessions (L1-L7). The number after "S", "M", or "L" indicates the query index in the session.

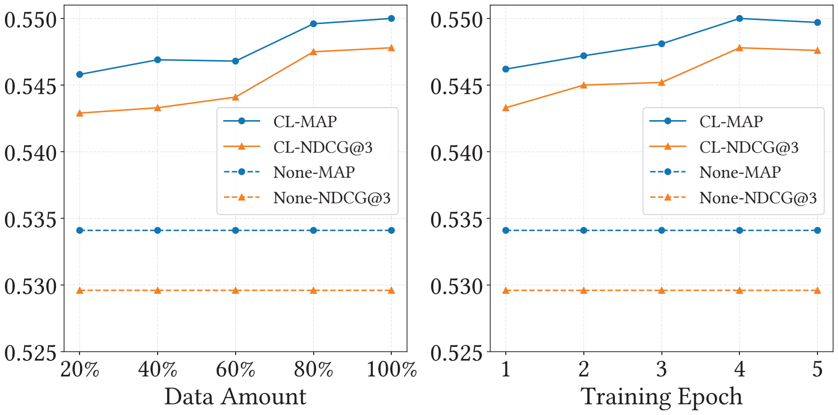


Fig. 7. The performance with different training data amount and training epochs.

much more effective in modeling search context. One interesting finding is that, when the search sessions get longer (*e.g.*, from L4 to L7), the gain of `COCA` diminishes. We attribute this to the more noisy nature of long sessions.

4.5.5. Influence of Amount of Training Data. As reported by recent studies [**23, 49**], the amount of data for contrastive learning has a great impact on downstream task (*e.g.*, document ranking in our case). We investigate such influence by training the model with different proportions of data and different epochs. As a comparison, we also illustrate the performance of `COCA` without sequence representation optimization stage (denoted as "`None`").

We first reduce the number of training data used for contrastive learning.[8] It is clear that contrastive learning benefits from a larger amount of data. Surprisingly, our proposed sequence representation optimization stage can still work with only 20% of training data.

[8] Note that all models are trained four epochs with different number of data.

This demonstrates the potential and effectiveness of learning better sequence representation for context-aware document ranking. We also train `COCA` with different number of epochs in the sequence optimization stage. The performance on document ranking is shown in the right side of Figure 7. The results suggest that the contrastive learning also benefits from larger training epochs. In our implementation, the data augmentation strategies are randomly selected in different epochs. Therefore, the sequence representation can be more fully learned. When training more than four epochs, the performance is stable without further improvement. Therefore, four epochs is the best choice in our experiments.

5. Conclusion and Future Work

In this work, we aimed at learning better representation of user behavior sequence for context-aware document ranking. A self-supervised task with contrastive learning objective is introduced for optimizing sequence representation before learning document ranking. To construct positive pairs in contrastive learning, we proposed three data augmentation strategies at term, query/document, and user behavior level. These strategies can improve the generalization and robustness of sequence representation. The optimized sequence representation is used in document ranking task. We conducted comprehensive experiments on two large-scale search log datasets. The results clearly showed that our proposed method is very effective. In particular, our method with contrastive learning was shown to outperform the close competitor `HBA` without it.

This is the first attempt to utilize contrastive learning in IR and much remains to be explored. For example, it may be more appropriate to exploit recent history instead of the whole history. Query and document weighting in the history could also be a promising avenue.

Second Article.

From Easy to Hard: A Dual Curriculum Learning Framework for Context-Aware Document Ranking

by

Yutao Zhu[1], Jian-Yun Nie[1], Yixuan Su[2], Haonan Chen[3], Xinyu Zhang[4], and Zhicheng Dou[3]

([1]) University of Montreal, Quebec, Canada

([2]) Language Technology Lab, University of Cambridge, Cambridge, United Kingdom

([3]) Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

([4]) Huawei Poisson Lab, Hangzhou, Zhejiang, China

This article was published in Proceedings of the 31st ACM International Conference on Information and Knowledge Management, pp. 2784-2794. 2022 .

The main contributions of Yutao Zhu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Jian-Yun Nie and Zhicheng Dou contributed to the model design and paper writing; Yixuan Su contributed to refining the idea; Haonan Chen helped conduct the experiments; Xinyu Zhang provided the computation resources for experiments.

Résumé. Les informations contextuelles dans les sessions de recherche sont importantes pour saisir les intentions de recherche des utilisateurs. Diverses approches ont été proposées pour modéliser les séquences de comportement des utilisateurs afin d'améliorer le classement des documents dans une session. Généralement, les échantillons d'entraînement des paires (contexte de recherche, document) sont échantillonnés de manière aléatoire dans chaque époque d'entraînement. En réalité, la difficulté de comprendre l'intention de recherche de l'utilisateur et de juger de la pertinence du document varie considérablement d'un contexte de recherche à l'autre. Le mélange d'échantillons d'apprentissage de difficultés différentes peut perturber le processus d'optimisation du modèle. Dans ce travail, nous proposons un cadre d'apprentissage par curriculum pour le classement de documents en fonction du contexte, dans lequel le modèle de classement apprend des signaux de correspondance entre la paire de contexte de recherche et de document candidat de plus facile à plus difficile. Ce faisant, nous visons à guider le modèle progressivement vers un optimum global. Pour tirer parti des exemples positifs et négatifs, deux curriculums sont conçus. Les expériences sur deux ensembles de données réelles de log de requête montrent que notre cadre proposé peut améliorer la performance de plusieurs méthodes existantes de manière significative, démontrant ainsi l'efficacité de l'apprentissage par curriculum pour le classement des documents en fonction du contexte.
Mots clés : Apprentissage du programme scolaire, difficulté de l'échantillon, classement des documents en fonction du contexte

Abstract. Contextual information in search sessions is important for capturing users' search intents. Various approaches have been proposed to model user behavior sequences to improve document ranking in a session. Typically, training samples of (search context, document) pairs are sampled randomly in each training epoch. In reality, the difficulty to understand user's search intent and to judge document's relevance varies greatly from one search context to another. Mixing up training samples of different difficulties may confuse the model's optimization process. In this work, we propose a curriculum learning framework for context-aware document ranking, in which the ranking model learns matching signals between the search context and the candidate document in an easy-to-hard manner. In so doing, we aim to guide the model gradually toward a global optimum. To leverage both positive and negative examples, two curricula are designed. Experiments on two real query log datasets show that our proposed framework can improve the performance of several existing methods significantly, demonstrating the effectiveness of curriculum learning for context-aware document ranking.
Keywords: Curriculum Learning, Sample Difficulty, Context-aware Document Ranking

Prologue

Our previous study based on contrastive learning achieved the state-of-the-art performance on context-aware document ranking. During the research process, we noticed that understanding the user intents of search sessions represents different difficulties. Some queries are short and accurate, for which it is easy to retrieve relevant documents, while others are complex and ambiguous, which are hard to understand. However, all existing methods (including the one we proposed in the previous article) assumed that all positive and negative samples are equally important for model optimization, *i.e.*, all samples are uniformly sampled to form a training batch. Imagine that when human learners are presented with samples of mixed difficulties, they may be confused because signals from different samples may be inconsistent. Things are similar in model training: when models are trained with a batch of samples with mixed difficulties, their optimization process is easily disturbed. To tackle this problem, we introduce curriculum learning that organizes training samples in an easy-to-hard manner for optimization. By this means, models can first learn some basic matching signals from easy samples, and then leverage such knowledge in the learning with hard samples.

After we finish this article, there are several new methods for context-aware document ranking tasks. One study followed the idea of multi-task learning, and proposed several generative tasks as a supplement for learning document ranking [**21**]. It applied generative pre-trained language models (*i.e.*, BART [**96**]) and achieved good performance. Two studies aimed to enhance the current search session's representation by other similar sessions in the log data [**192, 220**]. Current session and extracted similar sessions construct a graph structure and is modeled by graph neural networks. Our work can be considered as the first attempt of using curriculum learning for context-aware document ranking.

1. Introduction

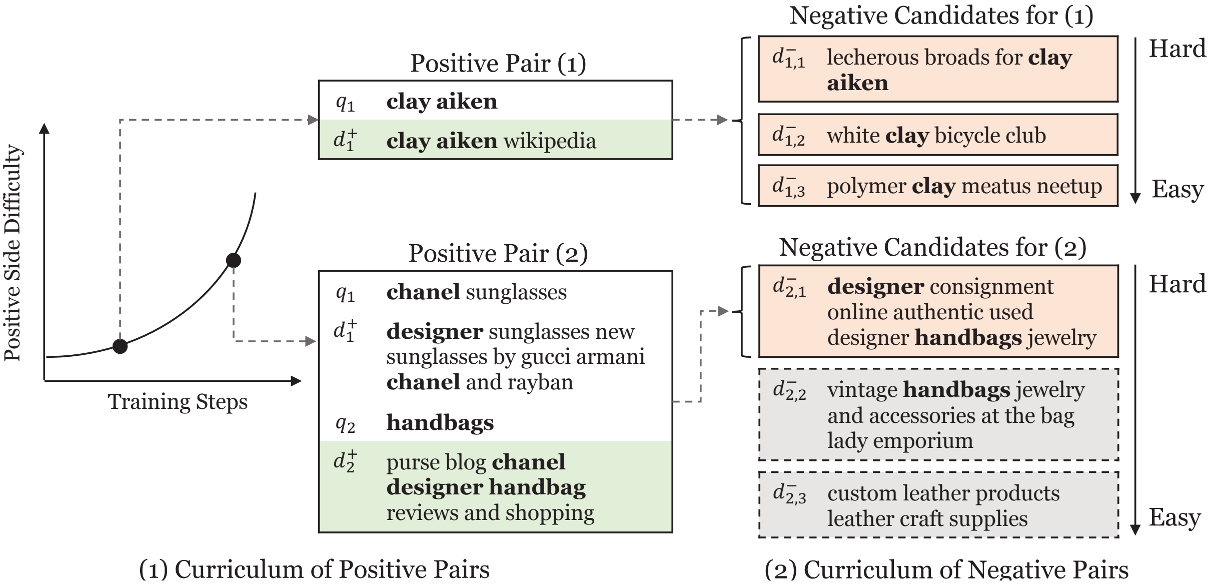
Users' search behaviors have evolved from one-shot queries to multiple interactions with search engines [**2**]. To fulfill a complex information retrieval task, users may issue a series of queries, examine and interact with some results. Many studies have shown that historical user behavior or search activities can be leveraged to improve document ranking, especially when users' search intent is ambiguous [**4, 9, 50, 83, 241, 253**].

Previous studies have exploited user search behaviors for understanding user intent and improving document ranking within a session [**9, 17, 169**]. Earlier research explored query expansion and learning-to-rank techniques [**9, 17, 64, 165**]. More recently, many neural architectures have been developed to model user behavior sequences and capture user search intent [**3, 4, 149, 169, 245**]. For example, a hierarchical RNN with attention mechanism was used to model the historical queries and the corresponding clicked documents, leading to better document ranking [**3, 169**]. Researchers also discovered that learning query suggestion as a supplementary task is also beneficial for document ranking [**4**]. Recently, pre-trained language models have also been used to capture search intent from user behavior sequences [**149, 245**].

Although the approaches proposed are different, they all rely on the information extracted from previous search logs, for example, (search context, document) pairs that are considered as positive or negative samples. The search context usually aggregates the current query and the previous queries (and interacted documents in some cases). All positive pairs are assumed to be samples of equal importance that reflect relevance, and are put together in the same pool for sampling a training batch. The same for negative pairs that reflect irrelevance. While it is true that positive and negative examples are generally useful for training a good ranking model, it is not true that they are equally useful at different training stages. In some cases, the relevance (or irrelevance) relation in a pair of (search context, document) is obvious, while in other cases, it is more subtle. Let us illustrate this by some examples. In the left part of Figure 8, we show two search contexts (sessions) containing queries and the documents clicked by the user. Let us try to understand the search intent in each of the search context. In the first case, the relevance relation between the search context formed by q_1 and the clicked document d_1^+ is clear because both are about the singer "Clay Aiken". In comparison, in the second case, the relation between the search context formed by $[q_1, d_1^+, q_2]$ and the clicked document d_2^+ is more difficult to capture. The underlying intent is "Chanel's designer handbags", which can only be understood with the help of the historical query q_1 and the clicked document d_1^+. If a human is asked to learn from these two samples, the relevance signal in the first one is much easier to capture than in the second. For a training process, the situation is similar: These pairs represent different levels of *difficulty* for a training process to digest. When a human learner is presented with samples of mixed difficulties, he/she can be confused because signals from different samples may appear inconsistent, or they do not have sufficient knowledge to understand difficult samples. Recent studies in machine learning also showed that learning with a batch of samples of mixed difficulties may disturb the optimization, especially when the network is deep [**7**]. In this case, curriculum learning, *i.e.*, learning from easy samples before hard samples, becomes particularly useful.

A similar problem occurs when negative samples are considered. As can be seen in the right side of Figure 8, for the second search context, the negative candidate $d_{2,1}^-$ contains information about "designer handbags". Compared with $d_{2,3}^-$, we can see that $d_{2,1}^-$ is much harder than $d_{2,3}^-$ to be recognized as a negative sample. Therefore, it is also desirable to learn from easy negative samples before hard ones. If we compare $d_{2,1}^-$ with the positive document d_2^+, we can learn that "chanel" is an important term reflecting the user's real intent. Such comparison/contrast is also common and critical for human learning, especially for discriminating similar concepts [**25**].

Motivated by the observations above, we propose a novel training framework that takes into account the levels of difficulty in the training samples. Our framework is inspired by both curriculum learning [**65, 139**] and contrastive learning [**23, 49, 173, 174**]. Curriculum learning simulates the human recognition process, *i.e.*, learning with easier samples first and more difficult samples later. It has achieved great performance on various tasks, such as image classification [**53, 65**], natural language understanding [**213**], and ad-hoc retrieval [**139**].


Positive Side Difficulty
Training Steps
Positive Pair (1)
q_1 **clay aiken**
d_1^+ **clay aiken** wikipedia
Positive Pair (2)
q_1 **chanel** sunglasses
d_1^+ **designer** sunglasses new sunglasses by gucci armani **chanel** and rayban
q_2 **handbags**
d_2^+ purse blog **chanel designer handbag** reviews and shopping
Negative Candidates for (1)
$d_{1,1}^-$ lecherous broads for **clay aiken**
$d_{1,2}^-$ white **clay** bicycle club
$d_{1,3}^-$ polymer **clay** meatus neetup
Hard
Easy
Negative Candidates for (2)
$d_{2,1}^-$ **designer** consignment online authentic used designer **handbags** jewelry
$d_{2,2}^-$ vintage **handbags** jewelry and accessories at the bag lady emporium
$d_{2,3}^-$ custom leather products leather craft supplies
Hard
Easy

(1) Curriculum of Positive Pairs

(2) Curriculum of Negative Pairs

Fig. 8. Illustration of DCL's utilization of samples. The left side shows positive pairs of different difficulties. In the early training steps, the model can only learn from easy pairs (*e.g.*, the first pair). Then, as the training progresses, harder pairs (*e.g.*, the second pair) are added. The right side is the curriculum of negative pairs. All negative candidates ($d_{1,1}^-$, $d_{1,2}^-$, and $d_{1,3}^-$) of easy cases are used for training in the early steps, while only hard negatives ($d_{2,1}^-$) are used in later steps.

Contrastive learning aims at learning representations such that similar samples stay close to each other, while dissimilar ones are far apart. By comparing similar and dissimilar samples, the model can be better optimized to capture their differences.

Concretely, we treat the search context and the candidate document as a pair and optimize the model by contrasting positive pairs (with clicked documents) and negative pairs (with unclicked documents). We design a dual curriculum learning framework incorporating two complementary curricula for positive and negative pairs, respectively. In the curriculum for positive pairs, sampling is restricted to easy pairs in early steps, and then extended gradually to the whole set of samples, so that hard pairs can also be learned. In the curriculum for negative pairs, we do the opposite: sampling from all pairs in early steps, then restricting gradually to hard pairs in late steps. These strategies are inspired by similar human learning: we select learning (positive) examples from easy to hard, but want to contrast them with all the negative examples at the beginning to have a better idea of the general differences between the positives and negatives. Toward the end of the learning, as the obvious differences have been learned, the easy negatives cannot provide effective learning signals, so we focus on distinguishing hard negative examples (these hard ones are demonstrated to be beneficial for model optimization [**212, 225**]). Our two curricula intend to follow the same principle during the training process.

The curriculum strategy can be used in any existing approach. We integrate it into three state-of-the-art approaches for context-aware document ranking. We conduct experiments on two large-scale search log datasets (AOL [**135**] and Tiangong-ST [**22**]). Experimental results show that our curriculum learning method significantly improves three strong baselines. The consistent performance gains demonstrate the effectiveness and wide applicability of our approach. Our further experiments show that both positive and negative curricula are beneficial to the ranking effectiveness.

Our contributions are three-fold:

(1) We propose a novel curriculum learning framework for context-aware document ranking, in which the difficulty of training samples is taken into account.

(2) We devise two complementary curricula for learning user intent from positive and negative pairs of (search context, candidate document). By learning them in an easy-to-hard manner, the model's performance can be improved gradually.

(3) Experimental results on two large-scale benchmark datasets show significant improvements. The experiments also confirm the broad applicability, flexibility, and high robustness of our method.

2. Related Work

2.1. Context-Aware Document Ranking

Context information in sessions has shown to be beneficial in modeling user search intent [**9, 50, 83**]. Early research focused on characterizing users' search intent by extracting contextual features from their search behaviors [**165, 198, 208**]. However, because these methods are built on handcrafted rules or manually collected features, they can only be used for a limited number of retrieval tasks. Researchers also developed predictive models for users' search intent or future behavior [**14**], but the learning of complex user-system interactions is limited by the predefined features.

The recent development of deep neural networks has triggered new approaches to context-aware document ranking. For example, researchers exploited hierarchical RNN-based architectures to model the sequence of historical queries [**79, 169, 201**]. These architectures were further enhanced by attention mechanism to better capture search behaviors [**24**]. It is also found that learning query suggestion and document ranking jointly can boost the performance on both tasks [**3**]. Besides, historical clicked documents are also reported to be helpful in predicting user search behaviors [**4**].

Recently, pre-trained language models, such as BERT [**34**], have achieved promising results on several NLP and IR tasks [**86, 87, 106, 113**]. Some researchers proposed concatenating all historical queries and candidate documents into a long sequence to compute a sequence representation using BERT, based on which the ranking score is determined [**149**]. Furthermore, contrastive learning has shown to be beneficial for optimizing the BERT encoder in context-aware document ranking [**245**].

Different from the studies above, we focus on improving the model optimization process by curriculum learning rather than designing new architectures or supplementary tasks for context-aware document ranking. Our work is orthogonal to the above approaches and can be combined with them. In fact, in most existing approaches, sampling of training data is necessary in the optimization process. This is typically done by random sampling or by selecting hard samples. Our work will show that selecting samples by curriculum from easy to hard can better optimize the existing models.

2.2. Curriculum Learning for IR

In the context of human learning, it is common to follow a curriculum that regulates the ordering and content of the education materials [**92, 137, 167**]. With this strategy, students can leverage previously learned concepts to help them learn new and more difficult ones. Inspired by research in cognitive science [**159**], researchers proposed machine learning algorithms based on a curriculum [**7, 37**]. The core idea is to train the model using easy samples first and increase the difficulty along the training process. Such a curriculum learning (CL) strategy has achieved great performance on several tasks, such as image classification [**53, 65**], machine translation [**142, 228**], dialogue generation [**172**], and natural language understanding [**213**].

In the area of IR, CL research is still in its early stage. The first attempt applied CL to learning-to-rank (LTR) [**42**], however, without much success. Later, researchers found that manually collected features in LTR can be a source of noise, and the CL strategy is more suitable for neural ranking models [**139**]. More recently, several heuristics have been proposed to determine the difficulty of different answers, based on which a CL-based method is used. This led to improved performance in answer ranking [**116**].

All existing CL-based methods for IR tasks are designed for organizing positive samples, but the influence of negative samples is neglected. We notice that some studies have focused on selecting hard negative samples for IR tasks [**97, 150, 225**], but they did not use CL. In this paper, we propose two complementary and contrastive curricula to enhance the model's learning using both positive and negative context-document pairs. Our experiments show that regulating the learning pace of both positive and negative samples is very effective.

3. Methodology

The goal of context-aware document ranking is to rank a list of candidate documents using the search context. In this work, we propose a **D**ual **C**urriculum **L**earning (`DCL`) framework

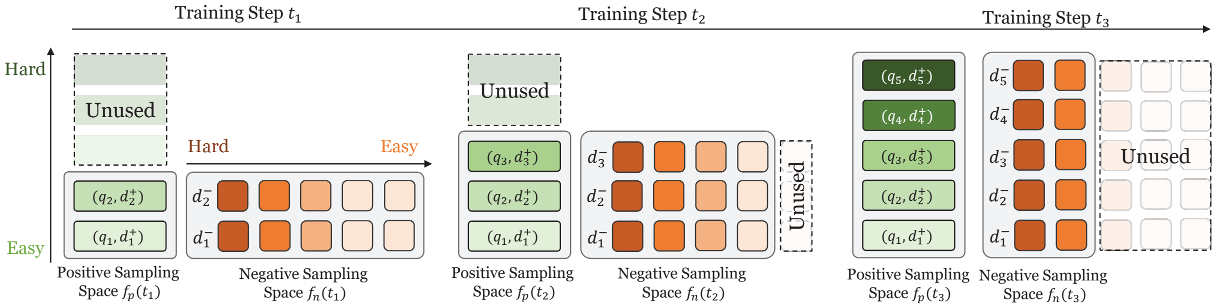


Fig. 9. The training process of our framework. For the curriculum of positive pairs, only easy samples are used at the beginning (t_1). Along the training process, the positive sampling space is gradually extended to the whole positive pairs (t_3). For the curriculum of negative pairs, the sampling space is shrinking from all samples (t_1) to only hard samples (t_3).

for this task. As shown in Figure 8, our framework consists of two complementary curricula for learning positive and negative context-document pairs. In each curriculum, the pairs are sorted according to their difficulty so that the model can learn them from easy to hard.

3.1. Notations and Task Definition

We first define some important concepts and notations before introducing our framework. A user's search behavior is represented as a sequence of n interactions $H_n = [q_1,d_1^+ \cdots ,q_n,d_n^+]$, where each query q_i is associated with a corresponding clicked document d_i^+. If there are several clicked documents, each of them is associated with the query to form a separate pair in the sequence. Each query q_i is represented by the original text string submitted to the search engine, while each clicked document d_i is represented by its text content. All queries are ordered according to the timestamps. For convenience, we further denote $C = [q_1,d_1^+, \cdots ,q_n]$ as a **search context** when q_n is submitted after a series of queries and interacted documents. When training a ranking model, for each positive clicked document, a set of m unclicked documents are selected as negative candidates.[9] As a result, the candidate document set for the search context C is represented as $D = \{d_n^+, d_1^-, \cdots d_m^-\}$ (the subscript n in d_n^+ will be omitted).

With the above notations, the task of context-aware document ranking can be defined as: ranking the candidate document set D based on the search context C so as to rank the clicked document d^+ as high as possible.

To make it clear, henceforth, we will call the pair (C, d^+) a **positive** pair, whereas the pairs $(C, d_i^-)_{i=1}^m$ are m **negative** pairs.

Notice that in this paper, we leverage user logs to learn a ranking model because user clicks can serve as a good proxy of relevance feedback [**4, 80, 81, 149**]. However, the same method can be used with human relevance judgments if available.

[9] Different selection strategies of negative candidates can be used in different datasets.

3.2. Overview

Our framework consists of two complementary curricula as follows:

(1) **Curriculum of Positive Pairs**. As shown in the left side of Figure 8, our first curriculum is designed for positive pairs. The target is to teach the model how to understand users' intent by capturing matching clues between search context and clicked documents. To achieve this goal, we first sort all positive pairs in the training set according to their difficulty and then let the model learn from easy ones to hard ones. By this means, the model can gradually increase its ability to capture matching signals.

(2) **Curriculum of Negative Pairs**. We also design a curriculum for negative pairs (right side of Figure 8) to enhance the model's ability to identify the mismatching information between search context and negative documents. Specifically, we progressively increase the difficulty of the negative candidate documents **associated with each selected positive pair** in the training set. In this way, the model is encouraged to gradually distinguish more subtle mismatching signals between search context and negative documents.

The principles used in the two curricula have some differences. The learning from positive pairs aims at identifying the relevance signals. In the curriculum, the model can gradually capture signals from shallow and easy matching (such as similar terms) to deep and hard matching (such as semantic information). On the contrary, a group of negative documents is associated with a specific positive pair, so they provide supplementary mismatching signals. As the curriculum progresses, the negative document becomes more similar to the positive document, so the model has to capture more fine-grained clues to distinguish them. Overall, we train the model with the two complementary curricula simultaneously so that its capability of modeling users' intent can be gradually enhanced. The whole process is shown in Figure 9.

It is worth noting that our `DCL` is a general training framework that works by organizing the learning order of the training samples. Therefore, it can be applied to various base models to improve their performance (this will be shown in our experiments).

3.3. Dual Curriculum Learning Framework

The implementation of our approach involves several key concepts, which we examine below.

How does curriculum learning work? When training neural networks, a mini-batch of training samples is usually randomly (*i.e.*, uniformly – every sample is selected with the same likelihood) sampled from the training set and used for optimizing the model at a step. Curriculum learning, on the other hand, aims at adjusting the order of the samples so that they are learned according to a predefined pace rather than at random. In our framework, we design two curricula for learning positive pairs and negative pairs from easy to hard, respectively. Following the paradigm of curriculum learning [**65, 139**], each curriculum is defined by two functions:

- A *difficulty* function determines the difficulty of samples so that the samples can be sorted according to their difficulty.
- A *pacing* function controls the learning pace. Essentially, it adjusts the sampling space to control the difficulty of the training samples at each step.

The curricula and their combination are described below.

3.3.1. Curriculum of Positive Pairs. The green part of Figure 9 illustrates the curriculum of positive pairs. Positive pairs are sorted from easy to hard according to a difficulty function. Thereafter, we gradually enlarge the sampling space so that more difficult positive samples will be included. The key is to define an appropriate difficulty function.

Difficulty Function. Different heuristics can be used to determine the difficulty of documents for a query [**16, 116**]. We consider two factors for measuring the difficulty of each positive training pair (C, d^+): (1) The first factor is the ranking score $M(\cdot,\cdot)$ between C and d^+ (see details in Section 3.3.4): A higher $M(C,d^+)$ indicates that it is easier to select d^+ based on C. (2) We also consider the position of the clicked document in the ranked list: A higher position indicates that it is easier to select it out of all documents. Formally, the difficulty for (C, d^+) is computed as follows:

$$d_p(C, d^+) = \underbrace{\text{rank}_C(d^+)}_{\in[1,|\mathcal{D}|]} + \underbrace{\left(1 - \frac{M(C,d^+)}{\max_{(C_i,d_i^+)\in\mathcal{D}} M(C_i,d_i^+)}\right)}_{\in(0,1]}, \tag{3.1}$$

where $\mathcal{D}$ is the training set. The first term is the ranking position of d^+ under the search context C; the second term is the normalized ranking score of (C,d^+). In this function, the difficulty is dominated by the position of the clicked document (the first term), while the normalized ranking score (the second term) makes an effect only when different (C,d^+) pairs are ranked at the same position. This particular definition of difficulty leads to good experimental performance among different alternatives we tested.

Pacing Function. The pacing function determines how the training process transitions from easy to hard pairs. Following previous studies on curriculum learning [**65, 139**], we define the pacing function $f_p(t)$ with respect to the training step t. The value of $f_p(t)$ is a proportion, and only the first $f_p(t) \times |\mathcal{D}|$ positive pairs can be used at the training step t. $f_p(t)$ is defined as follows:

$$f_p(t) = \min\left(1.0, \left(t \cdot \frac{1-\delta^k}{\alpha T} + \delta^k\right)^{\frac{1}{k}}\right), \tag{3.2}$$

where T is the total number of training steps, $\alpha,\delta \in (0,1)$ and $k \in [1, +\infty)$ are hyperparameters. As shown by the green line in Figure 10, this function has the following properties:

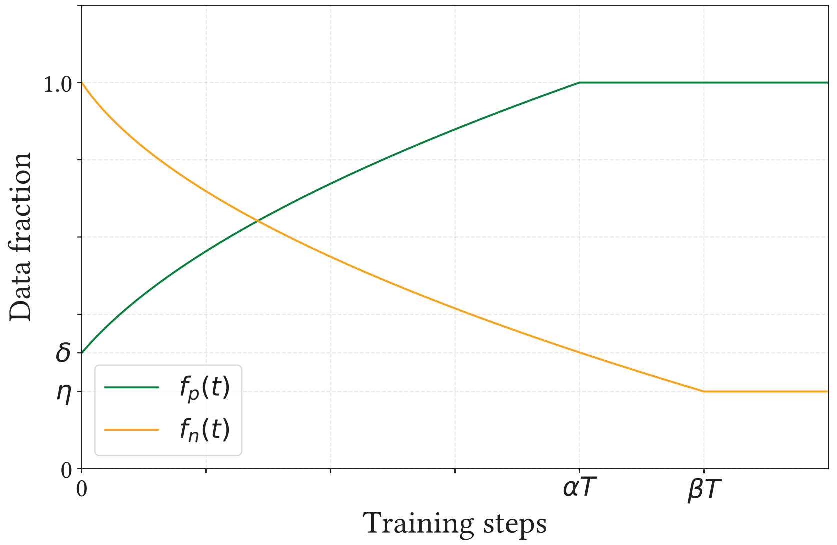


Fig. 10. Pacing functions used in both curricula.

(1) the initial value $f_p(0)$ is δ, so that the model can only use some easy pairs in the first training step; (2) it increases monotonically so that harder pairs are added to the training set gradually; (3) when it reaches αT steps ($f_p(\alpha T) = 1$), all pairs in the corpus can be used for training.

3.3.2. Curriculum of Negative Pairs. Negative samples are also very important for learning a ranking model [**225**]. By comparing positive and negative pairs, the model can learn what matching signals are vital in a contrastive manner.

The orange part of Figure 9 shows the curriculum of negative pairs. Similar to positive pairs, the negative pairs are also arranged according to their difficulties (more details later). By gradually constraining the sampling space, the model will focus on more difficult samples in later steps.

Difficulty Function. Similar to the positive curriculum, we use the relevance between the search context and the negative candidate as the difficulty of the negative pair. A negative candidate with a high ranking score to the search context is deemed to be hard to distinguish. For a negative pair (C,d_i^-), its difficulty is defined as:

$$d_n(C,d_i^-) = M(C,d_i^-), \tag{3.3}$$

where $M(\cdot,\cdot)$ is a scoring model similar to that used in the curriculum of positive pairs. Different from the difficulty function $f_p(t)$ in Equation 3.2, it is unnecessary to introduce the ranking position and normalization operation, because all negative candidates are associated with the same query; their ranking positions are determined by the ranking scores.

It is worth noting that we follow previous studies [**4**] to select negative candidates: the unclicked documents ranked around the clicked document (within a window) are considered as negative samples. By this means, we can avoid using too trivial or too hard negative documents. The obtained list of negative pairs is denoted as $\mathcal{L}$, in which all pairs are sorted according to their difficulties descendingly.

Pacing Function. The pacing function $f_n(t)$ for negative pairs is designed in a similar way to that for positive pairs. $f_n(t)$ adjusts the sampling space from which the negative sequences are sampled. It decreases with t. At time step t, we sample from the first $f_n(t) \times |\mathcal{L}|$ negative pairs for training. $f_n(t)$ is defined as:

$$f_n(t) = \max\left(\eta, 1 + \eta - \left(t \cdot \frac{1-\eta^k}{\beta T} + \eta^k\right)^{\frac{1}{k}}\right), \tag{3.4}$$

where β,$\eta \in$ (0,1) are hyperparameters. As shown by the organce line in Figure 10, this function is similar to $f_p(t)$, but makes opposite effect (*i.e.*, decreasing rather than increasing) along with t.

Remark. As shown in Figure 9 and Figure 10, we design the curriculum of negative pairs in a manner opposite to that of positive pairs, namely we gradually focus on only hard negative pairs. This is because: (1) All positive pairs are collected from human click data, which are very valuable for learning the real user intent. So, we choose to enlarge the positive sampling space and use all positive pairs in the end. (2) The negative pairs are sampled from the repository for facilitating the learning of the associating positive pairs. When the training progresses, too easy samples cannot provide enough "contrast effect", thus we discard them and only focus on hard samples. This strategy can lead to a more robust optimization in retrieval performance [**225**].

3.3.3. Combination of Two Curricula. `DCL` trains ranking models with the two curricula simultaneously. Specifically, for a training step t, we build a batch of training data as follows: First, we select a batch of positive pairs (C, d^+) according to the pacing function $f_p(t)$. Then, for each search context C in the positive pairs, we sample m negative candidate documents based on the pacing function $f_n(t)$. The process is summarized in Algorithm 1.

The whole training process of our framework naturally simulates two learning paradigms of human beings. On the one hand, the learning material is organized from easy to hard, which has been demonstrated to be effective for both animal training and human learning. In our case, such a training process is beneficial for model optimization. On the other hand, in cognitive science, learning through comparison is also an effective way to understand new concepts [**70**]. In our framework, the model can learn by contrasting positive and negative candidate documents.

Algorithm 1 Training in `DCL`

```
Input: the dataset $\mathcal{D}$; the ranking model; difficulty functions $d_p(\cdot,\cdot)$ and $d_n(\cdot,\cdot)$; pacing
  functions $f_p(\cdot)$ and $f_n(\cdot)$; the number of negative pairs $m$.
Score all positive pairs by $d_p(C,d^+)$ and sort them ascendingly;
for each training step $t$ do
    Collect the first $f_p(t) \cdot |\mathcal{D}|$ positive pairs as a subset $\mathcal{P}$;
    Uniformly sample a batch of positive pairs $\mathcal{B}_t$ from $\mathcal{P}$;
    for $[C_i,d_i^+]$ in $\mathcal{B}_t$ do
        Score all negative pairs by $d_n(C_i,d_j^-)$, get a list of negative pairs $\mathcal{L}$ by the heuristic
          rule, and sort them descendingly;
        Collect the first $f_n(t) \cdot |\mathcal{L}|$ negative pairs as a subset $\mathcal{N}$;
        Uniformly sample $m$ pairs $D_i^-$ from $\mathcal{N}$;
    end for
    Optimize the ranking model on the data $\{C_i,d_i^+,D_i^-\}_{i=1}^{|\mathcal{B}_t|}$;
end for
Output: Trained ranking model.
```

3.3.4. Selection of Scoring Model. In Equation (3.1) and (3.3), `DCL` applies a scoring model $M(\cdot,\cdot)$ to measure the ranking score between the search context and the candidate document. In our experiments, we use two different methods for $M(\cdot,\cdot)$: (1) BM25 [**157**] and (2) a BERT-based score commonly used in dense retrieval [**85, 87**]. Since we need to compute the ranking score between a search context and all documents in the training set, we apply dot-product based on BERT representations for fast computation:

$$M(C,d) = \text{BERT}(C)_{\text{[CLS]}} \cdot \text{BERT}(d)_{\text{[CLS]}}. \tag{3.5}$$

To achieve better performance, we fine-tune BERT encoders on positive sequences with in-batch negatives [**85**]. Then, we can obtain the representations of all the search contexts and documents. Afterwards, we use FAISS [**82**] to compute $M(C,d)$ efficiently.

4. Experiments

4.1. Datasets and Evaluation Metrics

Following previous work [**4, 20, 149, 245**], we conduct experiments on two public datasets: AOL search log data [**135**] and Tiangong-ST query log data [**22**].[10] Another possibility is the MS MARCO Conversational Search dataset [**128**], but the sessions in it are artificially constructed rather than derived from real search logs and the corresponding clicked documents are unavailable. Therefore, we do not use the MS MARCO dataset in experiments.

[10] We understand that the AOL dataset should normally not be used in experiments. We still use it because there are not many datasets available that fit our experiments well.

Table 5. The statistics of the datasets. The number in paretheses is the average number of relevant documents.

AOL	**Training**	**Validation**	**Test**
# Sessions	219,748	34,090	29,369
# Queries	566,967	88,021	76,159
Avg. # Query per Session	2.58	2.58	2.59
Avg. # Document per Query	5	5	50
Avg. Query Len	2.86	2.85	2.9
Avg. Document Len	7.27	7.29	7.08
Avg. # Clicks per Query	1.08	1.08	1.11
Tiangong-ST	**Training**	**Validation**	**Test**
# Sessions	143,155	2,000	2,000
# Queries	344,806	5,026	6,420
Avg. # Query per Session	2.41	2.51	3.21
Avg. # Document per Query	10	10	10
Avg. Query Len	2.89	1.83	3.46
Avg. Document Len	8.25	6.99	9.18
Avg. # Clicks per Query	0.94	0.53	(3.65)

We use the **AOL** dataset constructed by Ahmad et al. [**4**]. The dataset contains a large number of sessions, each of which consists of several queries. Each query in the training and validation sets has five candidate documents. The test set uses 50 documents retrieved by BM25 [**157**] as candidates for each query. All queries have at least one satisfied click in this dataset.

Tiangong-ST dataset is collected from a Chinese commercial search engine. It contains web search session data extracted from an 18-day search log. Each query in the dataset has ten candidate documents. In the training and validation sets, we use the clicked documents as the satisfied clicks. For queries with no satisfied clicks, we use a special token "[Empty]" for padding. In the test set, the candidate documents for the last query of each session have been manually annotated with relevance scores from from 0 to 4, which are used for evaluation. More details can be found in [**22**].

The statistics of both datasets are shown in Table 5. In Tiangong-ST test set, the relevance scores are provided instead of click labels, so we report the average number of documents with positive relevance scores (≥ 1). Following previous studies [**4, 74, 76, 245**], to reduce memory requirements and speed up training, we only use the document title as its content.

Evaluation Metrics. Similar to previous studies [**4, 149, 245**], we use Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at position k (NDCG@k, $k = \{1,3,5,10\}$) as evaluation metrics. In AOL, as clicked

Table 6. Experimental results on two datasets. All results using our framework (X+DCL) outperforms the original results (X) significantly at p-value < 0.01 with Bonferroni correction in paired t-test.

Model	**MAP**	**MRR**	**NDCG@1**	**NDCG@3**	**NDCG@5**	**NDCG@10**
			AOL			
ARC-I	0.3361	0.3475	0.1988	0.3108	0.3489	0.3953
ARC-II	0.3834	0.3951	0.2428	0.3564	0.4026	0.4486
KNRM	0.4038	0.4133	0.2397	0.3868	0.4322	0.4761
Duet	0.4008	0.4111	0.2492	0.3822	0.4246	0.4675
M-NSRF	0.4217	0.4326	0.2737	0.4025	0.4458	0.4886
M-Match	0.4459	0.4572	0.3020	0.4301	0.4697	0.5103
CARS	0.4297	0.4408	0.2816	0.4117	0.4542	0.4971
HBA	0.5281	0.5384	0.3773	0.5241	0.5624	0.5951
HBA+DCL	0.5599	0.5693	0.4074	0.5626	0.5961	0.6242
Improv.	+6.02%	+5.74%	+7.98%	+7.35%	+5.99%	+4.89%
RICR	0.5338	0.5450	0.3894	0.5267	0.5648	0.5971
RICR+DCL	0.5630	0.5742	0.4219	0.5589	0.5943	0.6257
Improv.	+5.47%	+5.36%	+8.35%	+6.11%	+5.22%	+4.79%
COCA	0.5500	0.5601	0.4024	0.5478	0.5849	0.6160
COCA+DCL	0.5794	0.5888	0.4281	0.5841	0.6167	0.6432
Improv.	+5.35%	+5.12%	+6.38%	+6.63%	+5.44%	+4.42%
			Tiangong-ST			
ARC-I	0.8580	0.9159	0.7088	0.7087	0.7317	0.8691
ARC-II	0.8611	0.9227	0.7131	0.7237	0.7379	0.8732
KNRM	0.8709	0.9261	0.7473	0.7505	0.7624	0.8891
Duet	0.8663	0.9273	0.7577	0.7354	0.7548	0.8829
M-NSRF	0.8517	0.9084	0.7124	0.7308	0.7489	0.8795
M-Match	0.8529	0.9211	0.7311	0.7233	0.7427	0.8801
CARS	0.8556	0.9268	0.7385	0.7386	0.7512	0.8837
HBA	0.8615	0.9316	0.7612	0.7518	0.7639	0.8896
HBA+DCL	0.8986	0.9538	0.8069	0.7985	0.8130	0.9122
Improv.	+4.31%	+2.38%	+6.00%	+6.21%	+6.43%	+2.54%
RICR	0.8147	0.8937	0.7670	0.7636	0.7740	0.8934
RICR+DCL	0.8963	0.9498	0.7995	0.7925	0.8078	0.9089
Improv.	+10.02%	+6.28%	+4.24%	+3.78%	+4.37%	+1.73%
COCA	0.8623	0.9382	0.7769	0.7576	0.7703	0.8932
COCA+DCL	0.8990	0.9501	0.7936	0.7922	0.8077	0.9088
Improv.	+3.21%	+1.27%	+2.15%	+4.57%	+4.86%	+1.75%

documents are used instead of manually judged documents, these measures reflect the capability of a model to rank clicked documents high. All evaluation results are obtained using the TREC's standard evaluation tool (trec_eval) [**63**].

4.2. Baseline

We compare our method with several baseline methods, including those for (1) ad-hoc ranking and (2) context-aware ranking.

(1) **Ad-hoc ranking methods** only use the current query without context information (historical queries and documents) for document ranking.

`ARC-I` [**72**] is a representation-based approach. Query and document representations are generated by CNNs. The ranking score is determined by a multi-layer perceptron (MLP).

`ARC-II` [**72**] is an interaction-based method. A matching map is constructed from the query and document, from which CNNs extract matching features. The score is also computed by an MLP.

`KNRM` [**211**] constructs a matching matrix by performing fine-grained interaction between the query and documents. The ranking features and scores are computed via kernel pooling.

`Duet` [**126**] uses both interaction- and representation-based features of the query and document extracted by CNNs and MLPs to compute ranking scores.

(2) **Context-aware ranking methods** utilize both context information and the current query to rank candidate documents.

`M-NSRF` [**3**] is a multi-task model that jointly predicts the next query and ranks corresponding documents. An RNN encodes a session's historical queries. The ranking score is calculated based on the representation of the query, the history, and the document.

`M-Match-Tensor` [**3**] (henceforth denoted as `M-Match`) is similar to `M-NSRF`, but learns a contextual representation for each word in the queries and documents. The ranking score is calculated by word-level representation.

`CARS` [**4**] also learns query suggestion and document ranking simultaneously. An attention mechanism is applied to compute representations for each query and document. The final ranking score is computed using the representation of historical queries, clicked documents, current query, and candidate documents.[11]

`HBA-Transformer` [**149**] (henceforth denoted as `HBA`) concatenates historical queries, clicked documents, and unclicked documents into a long sequence and applies BERT [**34**] to encode them into representations. A higher-level transformer structure with behavior embedding and relative position embedding enhances the representation. Finally, the ranking score is computed based on the representation of the "[CLS]" token.

`RICR` [**20**] is a unified context-aware document ranking model which takes full advantage of both representation and interaction. The session history is encoded into a latent representation and used to enhance the current query and the candidate document. Several

[11] We will notice some slight discrepancies between our results and those of the original paper of CARS. This is due to different tie-breaking strategies in evaluation. Following [**149, 245**], we use trec_eval while the authors of CARS use their own implementation.

matching components are applied to capture the interaction between the enhanced query and candidate documents. This model is based on RNNs and attention mechanism.

COCA [**245**] uses contrastive learning to improve a BERT's representation of user behavior sequences. By distinguishing similar user behavior sequences with dissimilar ones, the encoder can generate more robust representation. Then, the encoder is further used in context-aware document ranking. This is the current state-of-the-art document ranking method based on user behavior sequence.

Due to the limited space, the **implementation details** are provided in our code repository.[12]

4.3. Experimental Results

Experimental results are shown in Table 6. We choose three recently proposed methods (*i.e.*, HBA, RICR, and COCA) as the base model and train them with our proposed DCL framework. The corresponding results are reported as "X+DCL". To avoid the influence of randomness, we set three different random seeds and report the average performance. The standard deviation is less than 1e-3 for all results, which is omitted in the table. In general, our DCL significantly improves the performance of three base models in terms of all evaluation metrics on both datasets. This result clearly demonstrates DCL's superiority. We also have the following observations.

(1) The context-aware document ranking models generally perform better than ad-hoc ranking methods. For instance, on the AOL dataset, the weak contextualized model M-NSRF can still outperform the strong ad-hoc ranking model KNRM. This indicates that modeling user historical behavior is beneficial for understanding the user intent and determining the desired documents. For the three strong context-aware models, DCL can further improve their performance greatly, showing its effective utilization of training samples via curriculum learning.

(2) Compared with RNN-based methods (such as RICR and CARS), BERT-based methods (COCA and HBA) perform better. It is noticeable that DCL can bring improvements for both kinds of methods. Specifically, it improves the results by more than 4.7% and 1.2% in terms of all metrics on AOL and Tiangong-ST, respectively. This result demonstrates the wide applicability of our method to different base models and reflects the evident advantage of learning samples from easy to hard in context-aware document ranking.

(3) COCA is the state-of-the-art approach to context-aware document ranking. It involves a contrastive learning pre-training stage to help the BERT encoder learn more robust representations for user behavior sequences. In comparison, our DCL is a general training framework

[12] https://github.com/DaoD/DCL

Table 7. Ablation study of COCA+DCL. Mark "✓" and "×" indicate whether a curriculum is used or not. "Easy" or "hard" means only easy/hard samples are used for training.

Pos.	Neg.	MAP	MRR	NDCG@1	NDCG@3	NDCG@10
×	×	0.5500	0.5601	0.4024	0.5478	0.6160
✓	×	0.5750	0.5843	0.4222	0.5811	0.6391
×	✓	0.5740	0.5843	0.4231	0.5791	0.6381
✓	✓	**0.5794**	**0.5888**	**0.4281**	**0.5841**	**0.6432**
×	Easy	0.5240	0.5350	0.3677	0.5200	0.5939
×	Hard	0.5698	0.5792	0.4173	0.5741	0.6339

without a specific pre-training step, making it more efficient in practice.[13] In addition, COCA trained with DCL achieves a new state-of-the-art performance in context-aware document ranking task, showing the usefulness to combine both curriculum learning and contrastive learning.

4.4. Discussion

We further discuss several aspects of our proposed DCL. These analyses are based on the results on the AOL dataset, while we have similar findings on the Tiangong-ST dataset.

4.4.1. Impact of Both Curricula. As we propose two curricula for learning on positive and negative pairs, to validate their effectiveness, we conduct an ablation study by disabling each of them from COCA+DCL (*i.e.*, the sampling is done among all samples). The results are shown in Table 7. We can observe:

First, we can see that both curricula are useful. Applying any of them leads to performance improvement. When no curriculum learning is used, we observe large drops in performance. This directly validates our assumption in this paper that learning from easy to hard samples can guide the model in a good learning direction. Second, the curriculum of positive pairs brings slightly higher improvement than that of negative pairs. This suggests that the ability to capture positive matching signals is more critical than being able to discard negative signals. A possible explanation is that positive signals are more focused while the negative ones are diffuse.

Furthermore, to investigate the influence of samples' difficulty changes during the training, we replace the curriculum of negative pairs by only using easy or hard pairs, and the curriculum of positive pairs is disabled to avoid additional influence. The experimental results are shown in the bottom of Table 7.

[13] Compared to the pre-training step in COCA, DCL takes only around 1/3 time for training a BERT scorer, and this cost can be further reduced if BM25 scorer is applied.

Table 8. Performance with different difficulty scorers.

Pos.	Neg.	MAP	MRR	NDCG@1	NDCG@3	NDCG@10
None	None	0.5500	0.5601	0.4024	0.5478	0.6160
BM25	BM25	0.5661	0.5763	0.4159	0.5697	0.6293
BM25	BERT	0.5600	0.5701	0.4081	0.5632	0.6244
BERT	BM25	**0.5794**	**0.5888**	**0.4281**	**0.5841**	**0.6432**
BERT	BERT	0.5652	0.5755	0.4152	0.5684	0.6290

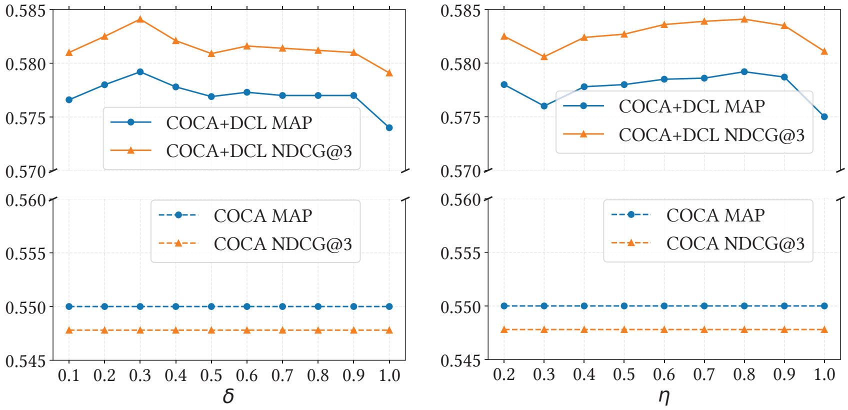


Fig. 11. Performance with different hyperparameters.

As can be seen, training with only hard negatives is even better than using all negatives (first row in the table). This finding is consistent with existing studies on using hard negatives to facilitate the optimization of dense retrievers [**212, 150, 225**]. However, only using easy samples for training makes the performance drop sharply. This is because the easy negatives cannot provide sufficient "contrastive signals" for learning the matching between search context and candidate documents. This is also why we design our curriculum of negative pairs as gradually shrinking to only hard negatives (details are presented in Section 3.3.2). Finally, dynamically adjusting the learning difficulty through curriculum is beneficial for model training (*e.g.*, MAP is improved from 0.5698 to 0.5740). This demonstrates again the effectiveness of applying curriculum learning.

4.4.2. Influence of Scoring Models. We proposed two scoring models for $M(\cdot,\cdot)$ – BM25 and BERT. We investigate their impact, and Table 8 shows the results of COCA+DCL.

We can observe the following: (1) Despite the differences in performance, DCL combined with each of the scoring methods can consistently bring improvements over training without curriculum. This shows that both scoring functions can help determine the difficulty of pairs. (2) BERT scores work better on the positive pair curriculum, while BM25 scores are better on the negative pair curriculum. The potential reasons could be: (a) The negative

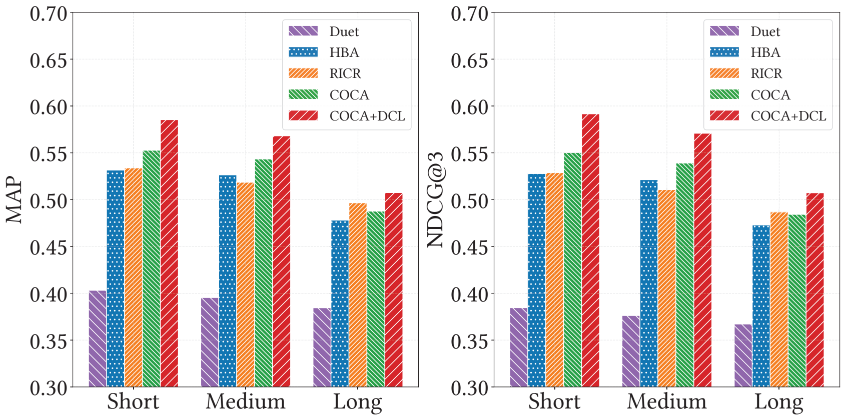


Fig. 12. Performance on different lengths of sessions on the AOL dataset.

candidates provided in the test set are also selected by BM25. As a result, the distribution of negative pairs selected by BM25 on the training set is closer to that on the test set, allowing the model to perform better. (b) Negative pairs are used for learning mismatching signals. Compared with BERT, BM25 can identify negative candidates containing similar terms. As term-level matching signals are critical in IR, such negative candidates can provide more useful information on term-level dissimilarity.

4.4.3. Influence of Hyperparameters. In DCL, δ (in Equation 3.2) and η (in Equation 3.4) are two hyperparameters that control the degree of difficulty in the initial set of positive pairs and in the set of final negative pairs. They are determined according to the validation set. We show their impact in Figure 11.

As we can see, when δ is small (< 0.4, *i.e.*, using very easy positive pairs at beginning), the performance is high. When δ becomes large (*i.e.*, including more difficult positive pairs at beginning), the performance drops. This confirms our hypothesis that the optimization process can be confused with more difficult pairs at beginning. When $\delta = 1.0$, the curriculum of positive pairs is disabled, so all positive pairs are learned in a random order. We can see that this common strategy used in the previous studies is suboptimal.

For negative pairs, when η is too small, we end the training process with a very small subset of the training data consisting of highly-ranked negative documents. In this case, the sampled documents may contain false negatives. The best performance is obtained with $\eta = 0.7$, *i.e.*, some mixture of easy and hard negative pairs is used at the end. However, when η is too large (*i.e.*, 1.0), all negative pairs are randomly sampled during the whole training, the model's performance also decreases because of the disabled curriculum effect. These observations confirm the impact of both curricula and suggest that the right degree of difficulty in the initial and final pools of samples may influence the effectiveness of DCL.

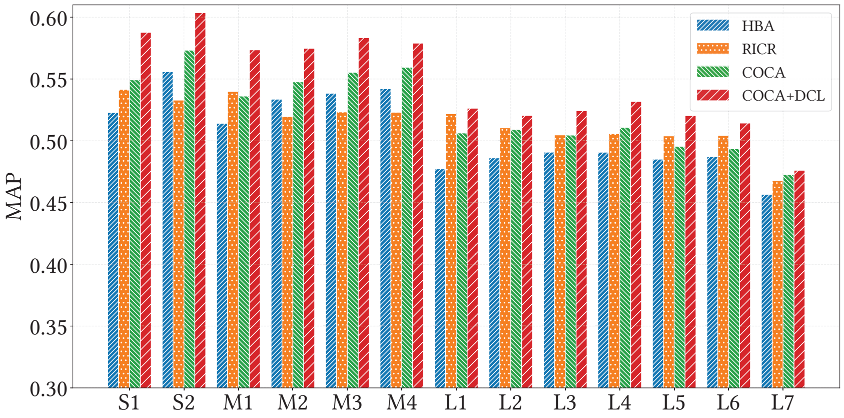


Fig. 13. Performance at different query positions in short (S1-S2), medium (M1-M4), and long sessions (L1-L7). The number after "S", "M", or "L" indicates the query index in the session.

4.4.4. Performance on Sessions with Different Lengths. To understand the impact of the session length on the final ranking performance, we categorize the sessions in the test set into three bins:

(1) Short sessions (with 1-2 queries) - 77.13% of the test set;

(2) Medium sessions (with 3-4 queries) - 18.19% of the test set;

(3) Long sessions (with 5+ queries) - 4.69% of the test set.

We compare `COCA+DCL` with `Duet`, `HBA`, `RICR`, and `COCA` and show the results regarding MAP and NDCG@3 in Figure 12. First, `COCA+DCL` improves the performance of `COCA` across all three session bins. This shows `DCL`'s high robustness for different kinds of search context. Second, we can see the ad-hoc ranking method `Duet` performs worse than other context-aware ranking methods. This highlights once again that modeling historical user behavior is essential for improving document ranking performance. Third, `COCA+DCL` performs better on short sessions than on long ones. We hypothesize that those longer sessions are inherently more difficult to understand, and a similar trend in baseline methods may corroborate this. This can be due to the fact that a long session may contain more diverse intents or exploratory search.

4.4.5. Effect of Modeling User Behavior Progression. It is important to study how the modeled search context contributes to document ranking as a search session progresses. We compare `COCA+DCL` with `HBA`, `RICR`, and `COCA` at individual query positions in short (S), medium (M), and long (L) sessions. The results are presented in Figure 13. Due to space limitations, long sessions with more than seven queries are omitted.

We can see that the ranking performance generally improves when the short and medium sessions progress (*e.g.*, S2 is higher than S1) because more search context information becomes available for predicting the next click. It benefits `COCA+DCL` and two baselines (`COCA` and HBA), while COCA+DCL improves more rapidly by better exploiting the context. One interesting observation is that, when the search sessions become longer (*e.g.*, from L4 to L7), the gain of DCL decreases. We attribute this to the noisier nature of long sessions.

5. Conclusion and Future Work

In this work, we proposed a novel curriculum learning framework for context-aware document ranking. Two complementary curricula were designed for learning positive and negative context-document pairs in an easy-to-hard manner. With these curricula, the model's capability of capturing matching signals and identifying mismatching signals is gradually enhanced. We conducted experiments with three recently proposed methods on two large-scale datasets. The results clearly demonstrate the effectiveness and wide applicability of our framework. Besides, we also investigated the influence of different settings on applying curriculum learning to context-aware document ranking. Our work is an early attempt to apply curriculum learning to IR, and there is still much space to be explored. For example, it may be useful to mine the most valuable negative candidate documents. Considering query and document weighting in computing the difficulty is also a future direction.

Third Article.

Content Selection Network for Document-grounded Retrieval-based Chatbots

by

Yutao Zhu[1], Jian-Yun Nie[1], Kun Zhou[2], Pan Du[1], and Zhicheng Dou[4]

([1]) University of Montreal, Quebec, Canada

([2]) School of Information, Renmin University of China, Beijing, China

([3]) Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

This article was published in Advances in Information Retrieval: 43rd European Conference on IR Research, Proceedings, Part I 43, pp. 755-769. 2021.

The main contributions of Yutao Zhu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Jian-Yun Nie and Zhicheng Dou contributed to the model design and paper writing; Kun Zhou helped conduct the experiments; Pan Du helped revise the paper.

RÉSUMÉ. Appuyer la conversation homme-machine sur un document est un moyen efficace d'améliorer les performances des chatbots basés sur la recherche. Cependant, seule une partie du contenu du document peut être pertinente pour aider à sélectionner la réponse appropriée à une conversation. Il est donc crucial de sélectionner la partie du contenu du document pertinente pour le contexte actuel de la conversation. Dans cet article, nous proposons un réseau de sélection du contenu des documents (`CSN`) pour effectuer une sélection explicite du contenu des documents pertinent, et filtrer les parties non pertinentes. Nous montrons dans des expériences sur deux ensembles de données de conversations publiques fondées sur des documents que `CSN` peut efficacement aider à sélectionner le contenu des documents pertinents pour le contexte de la conversation, et qu'il produit de meilleurs résultats que les approches de pointe. Notre code et nos jeux de données sont disponibles sur `https://github.com/DaoD/CSN`.

Mots clés : Sélection du contenu, dialogue basé sur des documents, chatbots basés sur la recherche

ABSTRACT. Grounding human-machine conversation in a document is an effective way to improve the performance of retrieval-based chatbots. However, only a part of the document content may be relevant to help select the appropriate response at a round. It is thus crucial to select the part of document content relevant to the current conversation context. In this paper, we propose a document content selection network (`CSN`) to perform explicit selection of relevant document contents, and filter out the irrelevant parts. We show in experiments on two public document-grounded conversation datasets that `CSN` can effectively help select the relevant document contents to the conversation context, and it produces better results than the state-of-the-art approaches. Our code and datasets are available at `https://github.com/DaoD/CSN`.

Keywords: Content Selection, Document-grounded Dialogue, Retrieval-based Chatbots

Prologue

At the time we wrote this article, existing studies on document-grounded retrieval-based dialogue systems concentrated on improving document-response interactions. The methods progressed from fusing the document and dialogue context as a single representation to respectively modeling their interactions with response candidates. Evidently, the latter methods can better capture the relationship between the document and response candidate, resulting in superior performance. Nonetheless, the document content is introduced to the dialogue process without careful selection. Only a soft attention mechanism has been used to assign weights. Since the document typically contains a lot of noise, even each piece of noise is assigned a small weight, the accumulated noise may strongly influence the dialogue. To address this issue, we proposed an approach to directly filter out the document content irrelevant to the current dialogue. A gate mechanism was designed to achieve this and brought significant improvement on this task.

After we published this paper, we noticed two concurrent studies that also performed document content filtering. All of these studies demonstrate the importance of reducing the noise of the document. As we wrote this thesis, pre-trained language models have become the dominant in various tasks, including document-grounded dialogues. The document is concatenated with the dialogue context as models' input. Content filtering is again a necessary component since the pre-trained language models have their own input length limit [**232**]. Therefore, our work is still relevant in this setting with pre-trained language models.

1. Introduction

Retrieval-based chatbots such as Microsoft XiaoIce [**166**] and Amazon Alexa [**153**] are widely used in real-world applications. Given a user input, an upstream retrieval system can provide a set of response candidates, and the retrieval-based chatbot should choose the appropriate one. This mainly relies on a matching score between the context and each candidate response. It has been found that the conversation context alone is insufficient in many cases for response selection [**180, 226**]. In fact, human conversations are usually also grounded in external knowledge or documents: our responses are strongly related to our knowledge or information contained in the documents at hand. On Reddit, for example, people usually discuss about a document posted at the beginning of a thread, which provides the background topics and basic facts for the following conversations. On Twitter, people may also exchange opinions related to a news article. In these cases, in addition to the conversation context, the document or news article also provides useful background information to guide response selection. A conversation that does not take into account the background information may lead to off-topic responses. This paper deals with the problem of document-grounded conversation - conversation based on a given document [**5, 58, 148, 226, 231**].

The task of document-grounded response selection is formulated as selecting a good response from a candidate pool that is consistent with the context and relevant to the document. Several existing studies have shown that leveraging the background document can significantly improve response selection [**58, 226, 231**]. Generally, the common strategy is selecting the response based on a combination of context-response matching and document-response matching. The latter can boost the responses that are related to the document content. However, a good response does not need to be related to the whole content of the document, but to a small part of it. The selection of the relevant part of the document is crucial.

The problem can be illustrated by an example from CMUDoG [**236**] in Figure 14. In this dataset, a movie-related wiki article is used as the grounding document. We can see that the conversation is highly related to the document. R1, R2, and R3 are three candidate responses for U6, and R3 is the desired response. The wrong response R1 could be highly scored because

Document			
Name	The inception	Year	2009
Director	Christopher Nolan	Genre	Scientific
Cast	Leonardo DiCaprio as Dom Cobb, a professional thief who specializes in conning secrets from his victims by infiltrating their dreams. Tom Hardy as Eames, a sharp-tongued associate of Cobb. ···		
Critical Resp.	Response DiCaprio, who has never been better as the tortured hero, draws you in with a love story that will appeal even to non-scifi fans. The movie is a metaphor for the power of delusional hype for itself.		
Intro.	··· Dominick Cobb and Arthur are extractors, who perform corporate espionage using an experimental military technology to infiltrate the subconscious of their targets and extract valuable information through a shared dream world. Their latest target, Japanese businessman Saito, reveals that he arranged the mission himself to test Cobb for a seemingly impossible job: planting an idea in a person's subconscious, or inception.		
Rating	Rotten Tomatoes: 86% and average: 8.1/10; IMDB: 8.8/10		

Conversation	
U1	Have you seen the **inception**?
U2	No, I have not but have heard of it. What is it about?
U3	It's about **extractors that perform experiments using military technology on people to retrieve info about their targets.**
U4	Sounds interesting. Do you know which actors are in it?
U5	I haven't watched it either or seen a preview. But it's scifi so it might be good. Ugh **Leonardo DiCaprio is the main character**. He plays as Don Cobb.
U6	I'm not a big scifi fan but there are a few movies I still enjoy in that genre. Is it a long movie?
R1	Many long shots are used to show the beautiful scene. Besides, it is really a good **story that will appeal even to non-scifi fans**!
R2	Well, not really. The **extractors** come out with the **military technology** and **infiltrate the subconscious**.
R3 ✓	Doesn't say how long it is. The **Rotten Tomatoes** score is **86%.**

Fig. 14. An example in CMUDoG dataset. The words in color correspond to those in the document. R3 is the ground-truth response.

it shares several key words with the document (*i.e.*, document-response matching score is high). However, R1 is not an appropriate response in the current context, which asks about the length of the movie. This example shows that a correct response is well grounded in the document not because it corresponds to the document content, but because it corresponds to the part relevant to the conversation context. Therefore, a first challenge is to select the part of the document content relevant to the current conversation context. R2 looks like a proper response to U6, yet it conveys similar information as U3, which makes the dialogue less informative. This response could be selected if we use the whole conversation history as conversation context - the response could have a high context-response matching score. In fact, the current context in this example is about the length of the movie. The previous utterances in the history are less relevant. This case illustrates the need to well calibrate and model the current conversation context.

The two key problems illustrated by the above example (R1 and R2) are not well addressed in previous studies:(1) They usually perform a soft selection of document content by assigning attention weights to them [**58, 231**]. Even though the less relevant parts could be assigned lower weights, the cumulative weight of many irrelevant parts could be large, so that they collectively influence the response selection in a wrong direction. We believe that a key missing element is a proper (hard) selection of the document content that fits the current conversation context, instead of a (soft) weighting. The hard selection of document content is motivated by the following observation: although the whole conversation can cover many aspects described in the grounding document, each of the step is related to only a small part of the document content. For example, in our conversation about a movie, we could discuss about an actor in one step. The selection of such a small part of the content is crucial. This observation advocates a hard selection rather than a soft weighting used in the previous studies. (2) The existing studies usually use the entire context to determine the weights of parts (sentences) of the document content. This strategy fails to distinguish the current conversation context from the ones in the history. As a result, a past round of conversation could be mistaken as the current one, leading to a redundant response as illustrated by the R2 example.

In this paper, we propose a **Content Selection Network** (`CSN`) to tackle these problems. **First**, we use a modified *gate mechanism* to implement the document content selection according to the conversation context, before using it to match with the response candidate. The content relevant to the current conversation step will be assigned a higher weight and pass the selection gate, while the irrelevant parts will be blocked. We use the gate mechanism to select sentences or words. **Second**, as the topic usually evolves during the conversation, we determine the current conversation context by focusing on the most recent utterances, rather than on the whole conversation history. To this end, we design a decay mechanism for the history to force the model focusing more on the current dialogue topic. The selected document contents and the conversation context are finally combined to select the candidate response.

The main contributions of this paper are: (1) We propose a content selection network to explicitly select the relevant sentences/words from the document to complement the conversation context. Our experiments show that this is a much more effective way to leverage the grounding document than a soft weighting. (2) We show that document-grounded conversation should focus on the topics in the recent state rather than using the whole conversation context. On two public datasets for document-grounded conversation, our method outperforms the existing state-of-the-art approaches significantly.

2. Related Work

2.1. Retrieval-based Chatbots

Existing methods for open-domain dialogue can be categorized into two groups: retrieval-based and generation-based. Generation-based methods are mainly based on the sequence-to-sequence (Seq2seq) architecture with attention mechanism and aim at generating a new response for conversation context [**13, 99, 162, 164, 209**]. On the other hand, retrieval-based methods try to find the most reasonable response from a large repository of conversational data [**111, 178, 205, 223**]. We focus on retrieval-based methods in this paper. Early studies use single-turn response selection where the context is a single message [**72, 78**], while recent work considers all previous utterances as context for multi-turn response selection [**178, 205, 223, 237**]. In our work, we also consider the whole conversation history (but with decaying weights).

2.2. Document-grounded Conversation

Multiple studies have shown that being grounded in knowledge or document can effectively enhance human-machine conversation [**52, 119, 226, 231**]. For example, a Seq2seq model is first applied to generate responses based on both conversation history and external knowledge [**52**]. An approach using a dually interactive matching network has been proposed, in which context-response matching and document-response matching are performed separately using a shared structure [**58**]. This model achieved state-of-the-art performance on persona-related conversation [**226**]. Recently, Zhao et al. [**231**] proposed a document-grounded matching network that lets the document and the context to attend to each other so as to generate better representations for response selection. Through the attention mechanism, different parts (sentences) of the document are assigned different weights and will participate in response selection to different extents. However, even though one may expect the noise contents (for the current step) be assigned with lower weights, they can still participate in response selection.

Our work differs from the existing studies in that we explicitly model the document content selection process and prevent the irrelevant contents from participating in response selection. In addition, we also define the current conversation context by focusing more on recent utterances in the history rather than taking the whole history indistinctly. These ideas will bring significant improvements compared to the existing methods.

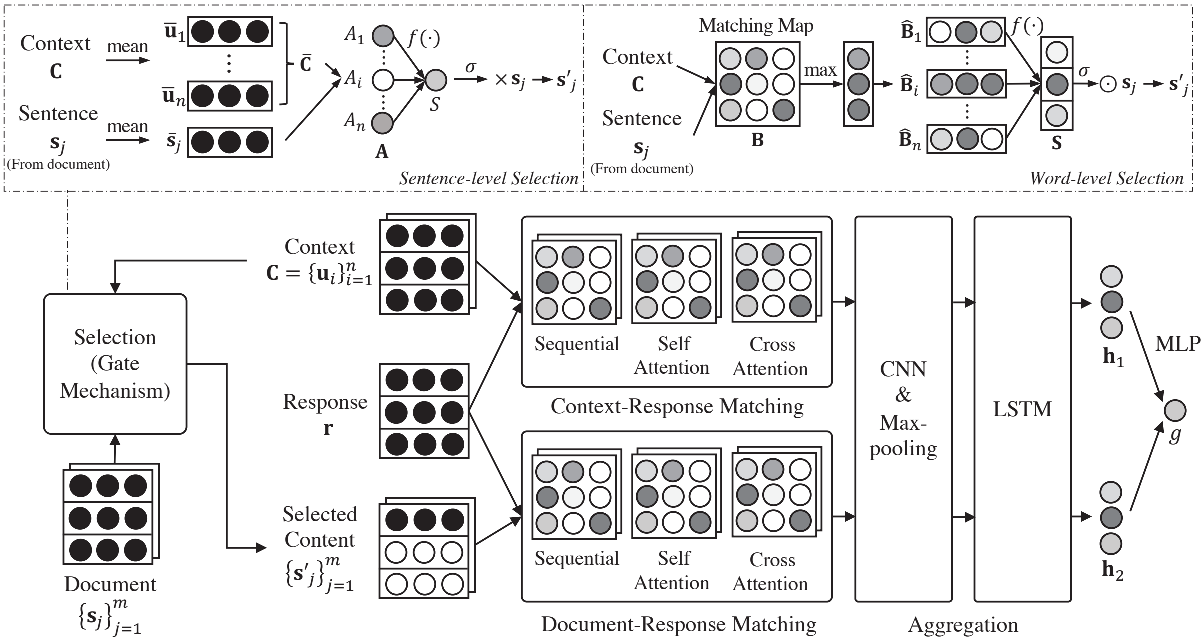


Fig. 15. The structure of CSN. From left to right, the model follows the "representation-matching-aggregation" paradigm. A gate mechanism is designed to select relevant content from the document.

3. Content Selection Network

3.1. Problem Formalization

Suppose that we have a dataset $\mathcal{D}$, in which each sample is represented as (c,d,r,y), where $c = \{u_1, \ldots, u_n\}$ represents a conversation context with $\{u_i\}_{i=1}^n$ as utterances; $d = \{s_1, \ldots, s_m\}$ represents a document with $\{s_i\}_{i=1}^m$ as sentences; r is a response candidate; $y \in \{0,1\}$ is a binary label, indicating whether r is a proper response. Our goal is to learn a matching model g from $\mathcal{D}$, such that for a new context-document-response triplet (c,d,r), $g(c,d,r)$ measures the degree of suitability of a response r to the given context c and the document d.

3.2. Model Overview

We propose a content selection network (CSN) to model $g(\cdot, \cdot, \cdot)$, which is shown in Figure 15. Different from the previous work [**58, 178, 205**] which uses the whole document contents, we propose a selection module with a gate mechanism to select the relevant parts of document content based on the context. Then, the context-response matching and the document-response matching are modeled based on the sequential, self-attention, and cross-attention representations. Finally, CNNs and RNNs are applied to extract, distill, and aggregate the matching features, based on which the response matching score is calculated.

3.3. Representation

Consider the i-th utterance $u_i = (w_1^u, \cdots, w_L^u)$ in the context, the j-th sentence $s_j = (w_1^s, \cdots, w_L^s)$ in the document, and the response $r = (w_1^r, \cdots, w_L^r)$, where L is the number of words.[14] `CSN` first uses a pre-trained embedding table to map each word w to a d_e-dimension embedding $\mathbf{e}$, *i.e.*, $w \Rightarrow \mathbf{e}$. Thus the utterance u_i, the sentence s_j, and the response r are represented by matrices $\mathbf{E}^{u_i} = (\mathbf{e}_1^{u_i}, \cdots, \mathbf{e}_L^{u_i})$, $\mathbf{E}^{s_j} = (\mathbf{e}_1^{s_j}, \cdots, \mathbf{e}_L^{s_j})$, and $\mathbf{E}^r = (\mathbf{e}_1^r, \cdots, \mathbf{e}_L^r)$, respectively. Then, `CSN` encodes the utterances, sentences and responses by bi-directional long short-term memories (BiLSTM) [**71**] to obtain their sequential representations: $\mathbf{u}_i = \text{BiLSTM}(\mathbf{E}^{u_i})$, $\mathbf{s}_j = \text{BiLSTM}(\mathbf{E}^{s_j})$, $\mathbf{r} = \text{BiLSTM}(\mathbf{E}^r)$. Note that the parameters of these BiLSTMs are shared in our implementation. The whole context is thus represented as $\mathbf{C} = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$. With the BiLSTM, the sequential relationship and dependency among words in both directions are expected to be encoded into hidden vectors.

3.4. Content Selection

In document-grounded conversation, the document usually contains a large amount of diverse information, but only a part of it is related to the current step of the conversation. To select the relevant part of document contents, we propose a content selection phase by a *gate mechanism*, which is based on the relevance between the document and the context. We design the gate mechanism at two different levels, *i.e.*, sentence-level and word-level, to capture relevant information at different granularities. If the sentences/words in the document are irrelevant to the current conversation, they will be filtered out. This is an important difference from the traditional gating mechanism, in which elements are assigned different attention weights, but no element is filtered out. We use the conversation context to control the gate, which contains several previous turns of conversation. Along the turns, the conversation topic gradually changes. The most important topic is that of the most recent turn, while more distant turns are less important. To reflect this fact, we design a decay mechanism on the history to assign a higher importance to the recent context than to the more distant ones. The selection process is automatically trained with the whole model in an end-to-end manner.

Sentence-level Selection. Let us first explain how document sentences are selected according to conversation context. Considering the context $c = (u_1, \cdots, u_n)$ and the j-th sentence s_j in the document, `CSN` computes a score for the sentence s_j by measuring its matching degree with the current dialogue context. In particular, `CSN` first obtains the sentence representations of the context c and the sentence s_j by mean-pooling over the word

[14] To simplify the notation, we assume their lengths are the same.

dimension of their sequential representations:

$$\bar{\mathbf{C}} = \underset{dim=2}{\text{mean}}(\mathbf{C}), \quad \bar{\mathbf{s}}_j = \underset{dim=1}{\text{mean}}(\mathbf{s}_j), \tag{3.1}$$

where $\bar{\mathbf{C}} \in \mathbb{R}^{n\times 2d}$ and $\bar{\mathbf{s}}_j \in \mathbb{R}^{2d}$. Then CSN computes a sentence-level matching vector $\mathbf{A}$ by cosine similarities:

$$\mathbf{A} = \cos(\bar{\mathbf{C}},\bar{\mathbf{s}}_j). \tag{3.2}$$

We can treat $\mathbf{A} \in \mathbb{R}^n$ as a similarity array $\mathbf{A} = [A_1, \cdots ,A_n]$ and compute a matching score S for the sentence s_j by fusing the similarity scores:

$$S = f(A_1,A_2, \cdots ,A_n). \tag{3.3}$$

The fusion function $f(\cdot)$ can be designed in different ways, which will be discussed later. After obtaining the matching scores for sentences, we select the relevant sentences and update their representations as follows:

$$S' = S \times (\sigma(S) \geq \gamma), \quad \mathbf{s}'_j = S' \times \mathbf{s}_j, \tag{3.4}$$

where $\sigma(\cdot)$ is the Sigmoid function and γ is a hyperparameter of the gate threshold. By this means, we will filter out a sentence s_j if its relevance score is below γ. The filtering is intended to remove the impact of clearly irrelevant parts of document content.

Word-level Selection. In the sentence-level selection, all words in a sentence are assigned the same weights. We can further perform a selection of words by computing a score for each word in the sentence. Specifically, CSN constructs a word-level matching map through the attention mechanism as follows:

$$\mathbf{B} = \mathbf{v}^\top \tanh(\mathbf{s}_j^\top \mathbf{W}_1 \mathbf{C} + \mathbf{b}_1), \tag{3.5}$$

where $\mathbf{W}_1 \in \mathbb{R}^{2d\times 2d\times h}$, $\mathbf{b}_1 \in \mathbb{R}^h$ and $\mathbf{v} \in \mathbb{R}^{h\times 1}$ are parameters. $\mathbf{B} \in \mathbb{R}^{n\times L\times L}$ is the word-alignment matrix between the context and the document sentence. Then, to obtain the most important matching features between s_j and each utterance in the context, CSN conducts a max-pooling operation as follows:

$$\hat{\mathbf{B}} = \max_{dim=3} \mathbf{B}, \tag{3.6}$$

where $\hat{\mathbf{B}} \in \mathbb{R}^{n\times L}$, and it can be represented in an array form as $\hat{\mathbf{B}} = [\hat{\mathbf{B}}_1, \cdots ,\hat{\mathbf{B}}_n]$. The element $\hat{\mathbf{B}}_i \in \mathbb{R}^L$ contains L local matching signals for all words in the document sentence s_j with respect to the utterance u_i. Thereafter, CSN applies a fusion function to combine these local matching signals and obtains a global matching vector:

$$\mathbf{S} = f(\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2, \cdots , \hat{\mathbf{B}}_n). \tag{3.7}$$

$\mathbf{S} \in \mathbb{R}^L$ thus contains L global matching scores for all words in $\mathbf{s}_j$ to the whole context. In the next step, `CSN` selects the relevant words in the document and updates the document representation as follows:

$$\mathbf{S}' = \mathbf{S} \odot (\sigma(\mathbf{S}) \geq \gamma), \quad \mathbf{s}'_j = \mathbf{S}' \odot \mathbf{s}_j, \tag{3.8}$$

where $\odot$ is the element-wise product. Different from the sentence-level matching score S' in Equation 3.4, the word-level matching score $\mathbf{S}'$ is a vector containing weights for different words.

Fusion Function. The fusion function $f(\cdot)$ in Equation (3.3) and (3.7) is used to aggregate the matching signals with each utterance in the context. Our fusion strategies attribute different weights to the utterances in the conversation history. Two different functions are considered: (1) Linear combination – the weight of each matching signal is learned during the model training. Ideally, an utterance containing more information about the conversation topic will contribute more to the selection of document content. (2) Linear combination with decay factors. This method assumes that the topic gradually changes along the conversation and the response is usually highly related to the most recent topic in the context. Therefore, we use a decay factor $\eta \in [0,1]$ on the utterances in the context to decrease their importance when they are far away. The matching scores are then computed as:

$$A_i = A_i * \eta^{n-i}, \quad \text{(sentence-level)} \qquad \hat{\mathbf{B}}_i = \hat{\mathbf{B}}_i * \eta^{n-i}. \quad \text{(word-level)} \tag{3.9}$$

The decay factor η is a hyperparameter. Note that when $\eta = 1$, it degenerates to the normal linear combination.

3.5. Matching and Aggregation

The next problem is to select the appropriate response by leveraging the selected document parts. Following a recent study [**58**], `CSN` uses a dually interactive matching structure (as shown in Figure 15) to determine context-response matching and document-response matching, where the two kinds of matching features are modeled by the same structure.

Based on the recent work [**205, 223, 237**] that constructs different matching feature maps, in addition to using the sequential representations of the sentences, `CSN` also uses matching on both self-attention and cross-attention representations. Given the sequential representations of the context $\mathbf{C} = [\mathbf{u}_1, \cdots, \mathbf{u}_n]$, the document $\mathbf{D} = [\mathbf{s}'_1, \cdots, \mathbf{s}'_m]$, and the response candidate $\mathbf{r}$, `CSN` first constructs a word-word similarity matrix $\mathbf{M}_1$ by dot product and cosine similarity:

$$\mathbf{M}_1^{cr} = \mathbf{C}\mathbf{H}_1\mathbf{r}^\top \oplus \cos(\mathbf{C},\mathbf{r}), \qquad \mathbf{M}_1^{dr} = \mathbf{D}\mathbf{H}_1\mathbf{r}^\top \oplus \cos(\mathbf{D},\mathbf{r}), \tag{3.10}$$

where $\mathbf{H}_1 \in \mathbb{R}^{2d\times 2d}$ is a parameter, and $\oplus$ is the concatenation operation.

To better handle the gap in words between two word sequences, `CSN` applies the attentive module, which is similar to that used in Transformer [**186**]. The input of an attentive module consists of three sequences, namely query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$). The output is a new representation of the query and is denoted as $f_{\text{ATT}}(\mathbf{Q},\mathbf{K},\mathbf{V})$ in the remaining description.

At first, `CSN` uses the attentive module over the word dimension to construct multi-grained representations, which is formulated as:

$$\hat{\mathbf{C}} = f_{\text{ATT}}(\mathbf{C},\mathbf{C},\mathbf{C}), \qquad \hat{\mathbf{D}} = f_{\text{ATT}}(\mathbf{D},\mathbf{D},\mathbf{D}), \qquad \hat{\mathbf{r}} = f_{\text{ATT}}(\mathbf{r},\mathbf{r},\mathbf{r}). \tag{3.11}$$

The second similarity matrix $\mathbf{M}_2$ is computed based on these self-attention representations:

$$\mathbf{M}_2^{cr} = \hat{\mathbf{C}}\mathbf{H}_2\hat{\mathbf{r}}^\top \oplus \cos(\hat{\mathbf{C}},\hat{\mathbf{r}}), \qquad \mathbf{M}_2^{dr} = \hat{\mathbf{D}}\mathbf{H}_2\hat{\mathbf{r}}^\top \oplus \cos(\hat{\mathbf{D}},\hat{\mathbf{r}}). \tag{3.12}$$

Then, another group of attentive modules (cross-attention) is also applied to represent semantic dependency between the context, the document, and the response candidate:

$$\tilde{\mathbf{C}} = f_{\text{ATT}}(\mathbf{C},\mathbf{r},\mathbf{r}), \qquad \tilde{\mathbf{r}}^c = f_{\text{ATT}}(\mathbf{r},\mathbf{C},\mathbf{C}), \tag{3.13}$$

$$\tilde{\mathbf{D}} = f_{\text{ATT}}(\mathbf{D},\mathbf{r},\mathbf{r}), \qquad \tilde{\mathbf{r}}^d = f_{\text{ATT}}(\mathbf{r},\mathbf{D},\mathbf{D}). \tag{3.14}$$

Next, `CSN` also constructs a similarity matrix $\mathbf{M}_3$ as:

$$\mathbf{M}_3^{cr} = \tilde{\mathbf{C}}\mathbf{H}_3\tilde{\mathbf{r}}^{c\top} \oplus \cos(\tilde{\mathbf{C}},\tilde{\mathbf{r}}^c), \qquad \mathbf{M}_3^{dr} = \tilde{\mathbf{D}}\mathbf{H}_3\tilde{\mathbf{r}}^{d\top} \oplus \cos(\tilde{\mathbf{D}},\tilde{\mathbf{r}}^d). \tag{3.15}$$

The above matching matrices are concatenated into two matching cubes:

$$\mathbf{M}^{cr} = \mathbf{M}_1^{cr} \oplus \mathbf{M}_2^{cr} \oplus \mathbf{M}_3^{cr}, \qquad \mathbf{M}^{dr} = \mathbf{M}_1^{dr} \oplus \mathbf{M}_2^{dr} \oplus \mathbf{M}_3^{dr}. \tag{3.16}$$

Then `CSN` applies a CNN with max-pooling operation to extract matching features from $\mathbf{M}^{cr}$ and $\mathbf{M}^{dr}$. The output feature maps are flattened as matching vectors. As a result, we obtain two series of matching vectors: (1) between the context and the response $\mathbf{v}^{cr} = [\mathbf{v}^{u_1}, \cdots, \mathbf{v}^{u_n}]$; and (2) between the selected document and the response $\mathbf{v}^{dr} = [\mathbf{v}^{s_1}, \cdots, \mathbf{v}^{s_m}]$.

Finally, `CSN` applies LSTMs to aggregate these two series of matching vectors into two hidden vectors (the last hidden states of the LSTMs):

$$\mathbf{h}_1 = \text{LSTM}(\mathbf{v}^{cr}), \quad \mathbf{h}_2 = \text{LSTM}(\mathbf{v}^{dr}). \tag{3.17}$$

These vectors are concatenated together and used to compute the final matching score by an MLP with a Sigmoid activation function:

$$g(c,d,r) = \sigma\Big(\text{MLP}(\mathbf{h}_1 \oplus \mathbf{h}_2)\Big). \tag{3.18}$$

`CSN` learns $g(c,d,r)$ by minimizing the following cross-entropy loss with $\mathcal{D}$:

$$\mathcal{L}(\theta) = -\sum_{(y,c,d,r)\in\mathcal{D}} [y\log(g(c,d,r)) + (1-y)\log(1-g(c,d,r))]. \tag{3.19}$$

4. Experiments

4.1. Dataset

We conduct experiments on two public datasets.

PersonaChat [**226**] contains multi-turn dialogues with user profiles. The goal is to generate/retrieve a response that corresponds to the user profile, which is used as a grounding document [**226**]. This dataset consists of 8,939 complete dialogues for training, 1,000 for validation, and 968 for testing. Response selection is conducted at every turn of a dialogue, and the ratio of the positive and the negative samples is 1:19 in training, validation, and testing sets, resulting in 1,314,380 samples for training, 156,020 for validation, and 150,240 for testing. Positive responses are real human responses while negative ones are randomly sampled from other dialogues. To prevent the model from taking advantage of trivial word overlap, the revised version of the dataset modified the persona profiles by rephrasing, generalizing, or specializing sentences, making the task much more challenging. We use "revised" and "original" to indicate the different versions of the dataset.

CMUDoG [**236**] is designed specifically for document-grounded conversation. During the conversation, the speakers are provided with a movie-related wiki article. Two scenarios are considered: (1) Only one speaker has access to the article thus she should introduce the movie to the other; (2) Both speakers have access to the article thus they have a discussion. We use the dataset provided by [**231**], where the data of both scenarios are merged because the size of each dataset is relatively small. Notice that the model is only asked to select a response for the user who has access to the document. The ratio of the positive and the negative is 1:19 in training, validation, and testing sets. This results in 723,180 samples for training, 48,500 for validation, and 132,740 for testing.

Following previous work [**231**], we employ recall at position k as evaluation metrics (R@k), where $k = \{1,2,5\}$. For a single sample, if the only positive candidate is ranked within top k positions, then R@$k = 1$, otherwise, R@$k = 0$. The final value is the average over all test samples. Note that R@1 is equivalent to hits@1 that is used in related work [**226, 58**].

4.2. Baseline Models

We compare `CSN` using sentence-level and word-level selection (denoted as CSN-sent and CSN-word respectively) with the following models:

(1) `Starspace` [**203**] concatenates the document with the context as a long sentence and learns its similarity with the response candidate by optimizing the embeddings using the margin ranking loss and k-negative sampling. Matching is done by cosine similarity of the sum of word embeddings.

Table 9. Experimental results on all datasets.

	PersonaChat-Original			PersonaChat-Revised			CMUDoG		
	R@1	**R@2**	**R@5**	**R@1**	**R@2**	**R@5**	**R@1**	**R@2**	**R@5**
`Starspace`	49.1	60.2	76.5	32.2	48.3	66.7	50.7	64.5	80.3
`Profile`	50.9	60.7	75.7	35.4	48.3	67.5	51.6	65.8	81.4
`KV Profile`	51.1	61.8	77.4	35.1	45.7	66.3	56.1	69.9	82.4
`Transformer`	54.2	68.3	83.8	42.1	56.5	75.0	60.3	74.4	87.4
`DGMN`	67.6	81.3	93.3	56.7	73.0	89.0	65.6	78.3	91.2
`DIM`	75.5	87.5	96.5	68.3	82.7	94.4	59.6	74.4	89.6
`CSN-sent`	77.5	88.8	96.8	70.1	83.4	95.1	**70.1**	82.5	**94.3**
`CSN-word`	**78.1**	**89.0**	**97.1**	**71.3**	**84.2**	**95.5**	69.8	**82.7**	94.0

(2) `Profile Memory Network` [**226**] uses a memory network with the context as input, then performs attention over the document to find relevant sentences. The combined representation is used to select the response. This model relies on the attention mechanism to weigh document contents.

(3) `Key-value (KV) Profile Memory Network` [**226**] uses dialogue histories as keys and the next dialogue utterances as values. In addition to the memory of the document, this model has a memory of past dialogues that can influence the response selection.

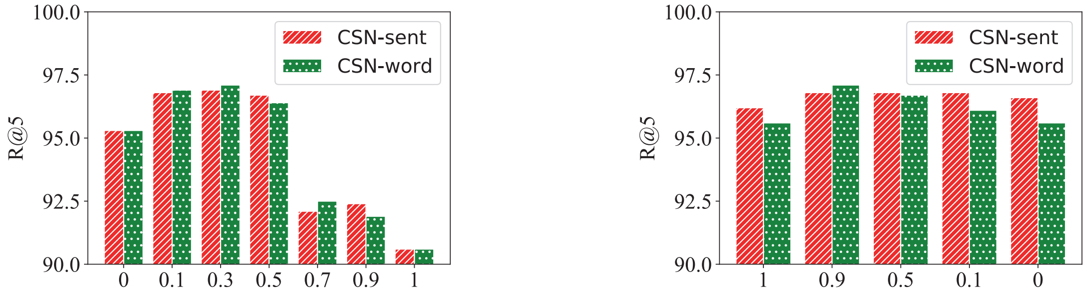
(4) `Transformer` [**186**] is used in [**119**] as an encoder for the context, document, and response. The obtained representations are input to a memory network to conduct matching in the same way as in Profile Memory Network.

(5) `DGMN` [**231**] is the state-of-the-art model on the CMUDoG dataset. It employs a cross attention mechanism between the context and document and obtains a context-aware document representation and a document-aware context representation. The two representations and the original context representation are all matched with the response representation. The three matching features are finally combined to output the matching score.

(6) `DIM` [**58**] is the state-of-the-art model on the PersonaChat dataset. It applies a dually interactive matching structure to model the context-response matching and document-response matching respectively. `DIM` conducts representation, matching, and aggregation by multiple BiLSTMs, and the final matching features are used to compute the matching score by an MLP.

4.3. Implementation Details

We use PyTorch [**136**] to implement the model. A 300-dimensional GloVe embedding [**141**] is used on all datasets. On PersonaChat, another 100-dimensional Word2Vec [**125**] embedding provided by [**58**] is used. Dropout [**171**] with a rate of 0.2 is applied to the word embeddings. All hidden sizes of the RNNs are set as 300. Two convolutional layers have



(a) Effect of γ. **(b)** Effect of η.

Fig. 16. Performance of different γ and η settings on original PersonaChat.

32 and 64 filters with the kernel sizes as [3, 3] and [2, 2]. AdamW [**110**] is employed for optimization with a batch size of 100. The initial learning rate is 0.001 and is decayed by 0.5 when the performance on the validation set is not increasing.

4.4. Experimental Results

The experimental results are shown in Table 9. The results on all three datasets indicate that our `CSN` outperforms all baselines, including `DGMN` and `DIM`, which are two state-of-the-art models. On the PersonaChat dataset, both `CSN-word` and `CSN-sent` achieve statistically significant improvements (p-value ≤ 0.05) compared with `DIM`, which is the best model on this dataset. In general, `CSN-word` performs better than `CSN-sent`, indicating the word-level selection is more able to select fine-grained document contents than the sentence-level selection. This comparison also confirms our intuition that it is advantageous for document-grounded conversation to rely on fine-grained information from the document. On CMUDoG, the two document content selection strategies work equally well. We explain this by the fact that the grounding document is longer in this dataset, and there is no obvious reason that one level of selection can determine more relevant parts than another. Nevertheless, both selection strategies show clear advantages over the baseline methods without selection.

Compared with other baselines that represent the whole document as a single vector, `DGMN`, `DIM`, and our `CSN` consider fine-grained matching between parts of the document and response. We can see that these models achieve clearly better performances, confirming the necessity to use parts of the document rather than the whole document. However, `DGMN` and `DIM` only assign attention weights to sentences according to the context, without eliminating low-weighted ones. In contrast, our `CSN` model filters out all the irrelevant parts. In so doing, we expect the model not to be influenced by clearly irrelevant parts. As we can see in the experimental results, `CSN` achieves significantly higher performance than `DGMN` and `DIM` on all the datasets, confirming the usefulness of explicit selection (and filtering) of document contents.

Effect of Content Selection. The hyperparameter γ in Equation (3.4) and (3.8) controls how much the document content is selected. We test the effect of this hyperparameter on the original PersonaChat dataset. Figure 16a shows that if γ is too small or too large, too much or too little information from the document may be selected. In particular, when $\gamma = 0$ – the whole document content is kept, the performance drops a lot. This strategy is comparable to that used in the existing models `DIM` and `DGMN` based on attention. We see again the usefulness of explicit document content filtering. On the other hand, when $\gamma = 1$, *i.e.*, no document content is selected, it degenerates to non document-grounded response selection and the performance also drops sharply. The best setting of γ is around 0.3 for both `CSN-sent` and `CSN-word`, which retains an appropriate amount of relevant document content for response matching.

Effect of Decaying Factor. The decay factor η works as prior knowledge to guide the model focusing more on the recent utterances. A lower η means the previous utterances have less contribution in the selection of the document. "$\eta = 1$" corresponds to the model with a normal linear combination (the first kind of fusion function). Based on the results, we can see that our decaying strategy ($\eta = 0.9$) performs the best. This confirms our assumption that focusing more on the recent topic of the conversation is helpful. However, when $\eta = 0$, only the last utterance in the history is used and the performance is lower. This illustrates the necessity of using a larger context.

5. Conclusion and Future Work

In this paper, we proposed a document content selection network to select the relevant content to ground the conversation. We designed a gate mechanism that uses conversation context to retain the relevant document contents while filtering out irrelevant parts. In addition, we also use a decay factor on the conversation history to focus on more recent utterances. Our experiments on two large-scale datasets for document-grounded response selection demonstrated the effectiveness of our model. We showed that both document content selection (and filtering) and the use of decay factor contributed in increasing the effectiveness of response selection. As a future work, it would be interesting to study if the selection can be done at topic level, in addition to sentence and word levels.

Fourth Article.

Leveraging Narrative to Generate Movie Script

by

Yutao Zhu[1], Ruihua Song[2], Jian-Yun Nie[1], Pan Du[3], Zhicheng Dou[2], and Jin Zhou[4]

([1]) University of Montreal, Quebec, Canada

([2]) Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

([3]) Thomson Reuters Labs, Canada

([4]) Beijing Film Academy, Beijing, China

This article was published in ACM Transactions on Information Systems (TOIS) 40, no. 4 (2022): 1-32.

The main contributions of Yutao Zhu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Ruihua Song and Jian-Yun Nie contributed to the model design and paper writing; Pan Du and Zhicheng Dou helped revise the paper; Jin Zhou helped annotate the experimental results.

RÉSUMÉ. La génération d'un texte basé sur une directive prédéfinie est un problème intéressant mais difficile. Une série d'études ont été menées ces dernières années sur ce sujet. Dans les systèmes de dialogue, les chercheurs ont exploré la conduite d'un dialogue basé sur un plan. Dans la génération d'histoires, l'utilité d'un scénario a également été prouvée. Dans cet article, nous abordons une nouvelle tâche : la génération de scénarios de films à partir d'une narration prédéfinie. Comme première exploration, nous étudions ce problème "basé sur la recherche", c'est-à-dire de sélectionner la meilleure réponse (*i.e.*, ligne de script suivante) parmi les candidats qui correspondent au contexte (*i.e.*, lignes de script précédentes) ainsi qu'à la narration donnée. Notre modèle, (`ScriptWriter-CPre`) peut garder la trace de ce qui a été dit dans la narration et de ce qui doit être dit. En outre, il peut également prédire quelle partie de la narration doit faire l'objet d'une plus grande attention lors de la sélection de la ligne de script suivante. Dans notre étude, nous constatons que la narration joue un rôle différent de celui du contexte. Par conséquent, des mécanismes différents sont conçus pour les traiter. En raison de l'indisponibilité de données pour cette nouvelle application, nous construisons une nouvelle collection de données à grande échelle *GraphMovie* à partir d'un site Web de cinéma où les utilisateurs peuvent télécharger librement leurs narrations lorsqu'ils regardent un film. Ce nouveau jeu de données est mis à la disposition du public afin de faciliter d'autres études sur la génération de textes selon des directives. Les résultats expérimentaux sur le jeu de données montrent que l'approche que nous proposons, basée sur la narration, est nettement plus performante que les méthodes de référence qui utilisent simplement la narration comme une sorte de contexte.
Mots clés : Génération de textes guidés par la narration, génération de script de films, méthode basée sur la recherche

ABSTRACT. Generating a text based on a predefined guideline is an interesting but challenging problem. A series of studies have been carried out in recent years. In dialogue systems, researchers have explored driving a dialogue based on a plan, while in story generation, a storyline has also been proved to be useful. In this paper, we address a new task–generating movie scripts based on a predefined narrative. As an early exploration, we study this problem in a "retrieval-based" setting. We propose a model (`ScriptWriter-CPre`) to select the best response (*i.e.*, next script line) among the candidates that fit the context (*i.e.*, previous script lines) as well as the given narrative. Our model can keep track of what in the narrative has been said and what is to be said. Besides, it can also predict which part of the narrative should be paid more attention to when selecting the next line of script. In our study, we find the narrative plays a different role than the context. Therefore, different mechanisms are designed for deal with them. Due to the unavailability of data for this new application, we construct a new large-scale data collection *GraphMovie* from a movie website where end-users can upload their narratives freely when watching a movie. This new dataset is made available publicly to facilitate other studies in text generation under the guideline. Experimental results on the dataset show that our proposed approach based on narratives significantly outperforms the baselines that simply use the narrative as a kind of context.
Keywords: Narrative-guided Text Generation, Movie Script Generation, Retrieval-based Method

Prologue

At the time we prepared this article, all existing dialogue systems work on providing a response in reaction to a user input. In this case, the dialogue would become diverse and scattered. So, we intended to explore whether the dialogue process can be more controllable. As an initial study, we set our goal as controlling a dialogue by a given text description (narrative). This problem can be treated as a kind of knowledge-grounded dialogue, where the given text serves as additional knowledge. However, the given text serves as a story, which should be completely covered by the dialogue, which is not required by the knowledge or dialogue context in knowledge-grounded dialogue. To support our study, we collected a new dataset from a website called GraphMovie. This website was an online platform where end users can use their own words to describe a movie with screenshots. We noticed that each description (called narrative) corresponds to a movie clip containing several lines of script. So, the narrative and the corresponding script lines naturally fit our study problem, *i.e.*, the narrative describes what happens in the script lines. Based on this dataset, we conducted many experiments and found that the narrative should be modeled in an different way than the dialogue (script) context. The model should be aware of which part of the narrative has been covered by the context and predict what should be mentioned in the next script line. After some exploration, we proposed a new mechanism to achieve our goal and successfully generated movie script based on given narratives. Our study demonstrated that it is feasible to control a dialogue process by a text description in a dialogue system.

In parallel to our study, some researchers explored using keywords to generate a paragraph [**41**] or a dialogue [**176, 147**]. Our work is different from theirs in that the narrative we used is a natural language text rather than some keywords. The model is required to understand the narrative content and model it in the script generation process. There was also an interesting study that investigates generating a recipe from given ingredients [**88**]. All ingredients should be covered by the recipe, which is similar to our narrative. Since the ingredients are usually selected from a fixed set, their vocabulary is smaller than our narrative. Overall, all these studies try to control text generation through another text. Such a task can find a wide applicability in the real life – to generate a story or a long text from a summary description. Notice that the recent ChatGPT has demonstrated such a capability. However, at the time we wrote the paper, no such system existed.

1. Introduction

A narrative is a series of related events or experiences, which can also be generally understood as a way of telling a story. The definition of narrative given by WordNet is "a message that tells the particulars of an act or occurrence or course of events; presented

Narrative	Jenny **doesn't like to go home**. To accompany Jenny, Gump decides to **go home later**. Gump is Jenny's **best friend**.	
Initial line	Mama's going to worry about me.	
1st line	Just **stay a little longer**.	✓
	Yeah, and I'll bet you $ 10,000 he laughs his ass off.	
2nd line	Ok, Jenny, **I'll stay**.	✓
	She lived in an old house.	
3rd line	He was a very loving man.	
	You are my most **special friend**.	✓

Fig. 17. An example of a part of a script with a narrative extracted from our *GraphMovie* dataset. The narrative generally describes the plot of a short session, which consists of several script lines. The checked lines are from a ground-truth session, while the unchecked ones are other candidates that are relevant but incoherent with the narrative. Our task is to select a proper line from a candidate list based on the given narrative and previous lines. By gradually selecting more lines, the whole session can be generated.

in writing or drama or cinema or as a radio or television program".[15] In natural language processing (NLP), narrative plays an important role in many tasks. For example, in story generation, the storyline can be treated as a type of narrative, which is helpful in generating coherent and consistent stories [**38, 39**]. In dialogue generation, the narrative can act as a global plan for the whole conversation session, so as to avoid generating inconsistent and scattered responses [**52, 180, 210**]. In movie script generation, the narrative can act as a guideline to organize the plot in order to faithfully present the ideas elaborated in the narrative.

In this work, we investigate the utilization of narratives in the special case of text generation–**movie script generation**. This special form of conversation generation is chosen due to the unavailability of the data for a more general form of application. Yet it does require the same care to leverage narratives as in general conversation, and hence can provide useful insight into a more general form of narrative-guided conversation or story generation. Our idea of using narrative as a guideline for script generation has two motivations: (1) **The use of narrative can make the generated script more consistent with previous script lines.** This has been demonstrated by existing studies that apply narratives or storylines as guidelines to conversation generation [**38, 39**]–Without the guidance of narratives, the utterances generated can easily deviate or become inconsistent, making the conversation useless and even unpleasant for the user. (2) **A narrative can provide a global view of what will happen in the plot so that the whole generated script lines can be**

[15] WordNet - narrative, `http://wordnetweb.princeton.edu/perl/webwn?s=narrative`

more coherent. The coherency of the generated texts is a problem not much addressed by the existing literature. Let us elaborate on the problem with a real example collected from the movie *Forrest Gump*. Given a narrative artificially written in several lines to retell a part of the plot in the movie, the corresponding conversation is shown in Figure 17. The first and third lines are spoken by Jenny, while the second line is given by Forrest. We can see that the narrative generally describes what happened in the several lines. If the narrative is not provided, the other choice of the second line ("She lived in an old house") can also be possible as it is consistent with the context (but it cannot lead to the following story). Therefore, it is important to leverage the narrative as a guideline for higher consistency and coherency of the generated movie script. This problem is very challenging, as has been recognized in some recent research [**38, 39**].

To alleviate the aforementioned problem, we study the problem of leveraging narrative to guide script generation, and we formulate our task as selecting the following lines by leveraging the narrative and previous lines. As an early exploration, we limit ourselves to the "retrieval-based" setting to simplify the task. To support our study on narrative-based generation, we collect a new dataset from GraphMovie, where end-users can retell the story of a movie by uploading descriptive paragraphs in their own words.[16] More details about the dataset will be presented in Section 3.2.

The problem we address is closely related to dialogue generation that takes into account the context [**205, 230, 237**]. However, a narrative plays a different role from a general context. Particularly, a narrative covers a part of a story, and a good conversation guided by a narrative should cover all aspects it delivers, which is not the case for generating dialogue with a general context. Intuitively, there are two challenges in leveraging narratives: (1) One should keep track of the coverage of the information of the narrative to be aware of what has been said before. This role of narrative is different from that of context. In dialogue generation, typical methods of context-based response selection focus on measuring the matching degree between the context and the response. If the same matching mechanism is used on a narrative, one will often see redundant utterances. Narrative tracking aims to cope with this problem. (2) We need to determine which remaining part of the narrative should be expressed when generating the next immediate line. The narrative is often organized in a specific chronological order. Thus, the model for narrative-guided text generation needs to be able to learn how to use the remaining information in the narrative. In summary, it is necessary to design a new mechanism to track and leverage the narrative in the script generation/selection process.

In this paper, we propose a new model called **`ScriptWriter-CPre`** to address the problem of script selection with the help of a narrative. This model is extended from our proposed

[16] Graph Movie, `http://www.graphmovies.com/home/2/index.php`. Unfortunately, we find this website closed recently.

ScriptWriter, which can keep track of what in the narrative has been said and what is still remaining to select the next line by an updating mechanism. Matchings between the updated narrative, context, and response are then computed respectively and finally aggregated as a matching score. Although the narrative status has already been tracked by comparing the current context (history) with the narrative, it is still hard for the model to determine which remaining part to cover in the next line. This problem is challenging largely because of the lack of direct supervision signals. With only the final loss, it is hard to tell whether the wrong line stems from the mismatching with the context or the wrong usage of the narrative. To tackle this problem, we extend ScriptWriter with a **C**ontent **Pre**diction (CPre) module, which is inspired by recent studies on knowledge-driven dialogue [**90, 104**]. Specifically, we use the last line in the context to predict a distribution over the narrative, which is denoted as the "prior distribution". Then, we also predict a distribution based on the ground-truth line, which is denoted as the "posterior distribution". By reducing the distance between the two distributions, the model can learn to predict (select) the content from the narrative that should be covered by the next line. This is achieved by adding a supplementary KL-divergence loss to the final loss. The clear feedback from this loss function can directly help the model learn to use the narrative.

Existing work on composing conversations includes generation-based methods and retrieval-based methods. Due to the advantages of informative and fluent responses, retrieval-based approaches for conversation generation have been widely explored in previous studies [**144, 205, 237, 247**]. Additionally, retrieval-based methods provide an easier way for us to evaluate the impact of the narratives in our work. Therefore, we frame our work as a retrieval-based conversation generation task in terms of "generating" responses by selecting proper responses from a set of candidates.

We conduct experiments on a dataset we collected and made publicly available (see Section 5). The experiments will show that using a narrative to guide the generation/selection of script is a much more appropriate approach than using it as part of the general context.

The problem we studied has several applications. Intuitively, our model can assist movie script authors to generate a new movie script. It can also be used for teaching, especially for beginners in movie creation. More generally, our method can be used in other text generation problems. For example, researchers have reported that generating a story from scratch is a very difficult problem. One of the solutions is providing the model with a predefined storyline. Our method can be directly applied for this problem. Besides, similar approaches could also be applied to dialogue generation with a narrative or any type of guidance.

Our work has three main contributions:

(1) Our work is an early exploration of movie script generation with a narrative. This task could be further extended to a more general text generation scenario when suitable data are available.

(2) We construct the first large-scale data collection *GraphMovie* to support research on narrative-guided movie script generation, which has been made publicly accessible.

(3) We propose a new model in which a narrative plays a specific role in guiding script generation. Our model can not only track what in the narrative has already been covered, but also predict the part that should be expressed in the next line. This will be shown to be more appropriate than a general context-based approach.

(4) Extensive comparisons with nine baseline methods on both automatic and human evaluation metrics demonstrate the superiority of our proposed method consistently. The ablation studies further validate the effects of different modules in our design. The influences of different hyper-parameters, narrative types, and context lengths are also revealed in the experiments. Moreover, a case study and error analysis are performed, which provides us some insightful findings and inspires some promising future work.

The rest of the paper is organized as follows. Related work is introduced in Section 2. The problem formulation and the collection of the dataset are presented in Section 3. Then we describe the details of our method in Section 4, followed by the description of the experiments in Section 5. More analysis is conducted in Section 6. Finally, we conclude our paper and discuss future work in Section 7.

2. Related Work

2.1. Narrative Understanding

It has been more than thirty years since researchers proposed "narrative comprehension" as an important ability of artificial intelligence [**154**]. The ultimate goal is the development of a computational theory to model how humans understand narrative texts. Early explorations used symbolic methods to represent the narrative [**11, 185**] or rule-based approaches to generate the narrative [**155**]. Recently, deep neural networks have been used to tackle the problem [**1**]. Related problems such as generating a coherent and cohesive narrative text [**26, 33, 77**], identifying relations in generated stories [**158**], and understanding relationship trajectories in narrative [**221**] have also been addressed. However, these studies only focused on how to understand a narrative itself (*e.g.*, how to extract information from a narrative or how to generate a narrative). They did not investigate how to utilize the narrative in an application task such as dialogue generation.

2.2. Dialogue Systems

Human-computer conversation is one of the most challenging tasks in NLP. The target of this task is to produce replies for human input messages. Conversation systems have been built for both open-domain and domain specific dialogues. The open-domain dialogue systems aim to produce natural and human-like conversation without restriction of domains. A typical application is the chatbot. On the contrary, domain specific dialogue systems are often task-oriented, such as ticket booking systems and automatic diagnosis systems.

2.2.1. Open-domain chatbot. Inspired by the Turing test proposed in 1950 [**183**], researchers and engineers have developed many conversational systems for chitchat [**29, 195**]. ELIZA, created in 1966, is perhaps the first chatbot known publicly [**195**]. It can only chat with people in a specific domain based on many hand-crafted scripts and limited domain knowledge. Recently, with large-scale human conversational data on the Internet, researchers have explored data-driven approaches to building a chatbot. Existing methods can be categorized into two groups. The first group of approaches learn response generation from the data. Most of the early works are inspired by statistical machine translation [**156**]. In recent years, neural network-based models have been widely used. Based on the sequence-to-sequence structure with attention mechanism [**164, 187**], multiple extensions have been made to tackle the "safe response" problem [**99, 244**]; to incorporate external knowledge [**209, 234**]; to generate responses with emotions or personas [**100, 115, 145**]; to model the hierarchical structure of the dialogue context [**162, 163, 210**] and to reduce the cost of response decoding [**206**]. Different from the generation-based methods, the second group of methods focus on searching for the most reasonable response from a large repository of conversational data. The key issue of these retrieval-based methods is how to measure the suitability of a response candidate for a user input. Early work studies single-turn response selection where the input is a single message [**72, 78**], while recent work pays more attention to context-response matching for multi-turn response selection. Representative methods include the deep learning to respond architecture (DL2R) [**217**], sequential matching network (SMN) [**205**], deep attention matching network (DAM) [**237**], deep utterance aggregation model (DUA) [**230**], interactive matching network (IMN) [**56**], and multi-hop selector network (MSN) [**223**]. Retrieval-based methods are widely used in real conversation products due to their more fluent and diverse responses and better efficiency. In this paper, we focus on extending retrieval-based methods by using a narrative as a plan for a session. This is a new problem that has not been studied before.

2.2.2. Task-oriented systems. Different from open-domain chatbots, task-oriented systems are designed to accomplish tasks in a specific domain [**94, 161, 182, 194**]. In these systems, a dialogue state tracking component is designed for tracking what has happened in a dialogue [**69, 199, 215**]. This inspires us to track the information in the narrative that has not been expressed by previous lines of conversation. However, existing methods for task-oriented dialogue cannot be applied to our task directly as they are usually predefined for specific tasks, and state tracking is often framed as a classification problem. For example, in a movie-booking dialogue system, "Where do you want to watch a movie?" would fit into a predefined dialogue act slot *request(city)*, while the reply “I want to watch it in Seattle.” is transformed into *inform(city=“Seattle”)* action [**46**]. Such state tracking and action design are impossible for our task because of the wide range of topics in movie scripts and corresponding narratives. To tackle this problem, we design a more general and flexible updating mechanism for the representation of the narrative, which keeps track of the information flow in movie dialogues.

2.3. Story Generation

Existing studies have also tried to generate a story. Early work relied on symbolic planning [**18, 120**] and case-based reasoning [**51, 216**], while more recent work uses deep learning methods. Some of them focused on story ending generation [**60, 138**], where the story context is given, and the model is asked to select a coherent and consistent story ending. This is similar to the dialogue generation problem mentioned above. Besides, attempts have been made to generate a whole story from scratch [**38, 39**]. Compared with the former task, the latter is more challenging since the story framework and storyline should all be controlled by the model.

2.4. Other Forms of Text Generation

Some other text generation work is related to ours. Feng et al. [**41**] proposed an LSTM-based model to generate a paragraph-level text with multiple topics. The topic is represented by five words which contain far less information compared to the narrative we use. Their goal of writing a long text, *e.g.*, an essay, is also different from movie script generation. Kiddon et al. [**88**] proposed an interesting task of generating a recipe from given ingredients. The ingredients are represented by a checklist of phrases, and the model needs to incorporate them into the recipe in a reasonable order. They propose a checklist mechanism to update an agenda to control which ingredients have not been used yet. This idea is similar to our idea of updating the narrative, but the narrative used in our task consists of open-domain natural language sentences, whose vocabulary size is much larger than the structured list of limited ingredients.

Some recent studies also tried to guide the generation of dialogues [**176, 204, 248**] or stories [**219**] with keywords - the next response is asked to include the keywords. This is a step towards guided response generation and bears some similarities with our study. However, a narrative is more general than keywords, and it provides a description of the dialogue session rather than imposing keywords to the next response.

Table 10. Statistics of *GraphMovie* corpus. A narrative is a description that summarizes a fragment of a movie. Each narrative corresponds to a session containing several lines of script. Micro-sessions are obtained by moving the prediction point through the session, each of which has a sequence of previous lines at that point of time, the same narrative as the session, and the next line to predict. The line candidates are used for prediction, which contain one golden line and several lines randomly sampled from the dataset.

	Training	**Validation**	**Test**
# Sessions	14,498	805	806
# Micro-sessions	136,524	37,480	38,320
# Candidates	2	10	10
Min. #lines in Session	2	2	2
Max. #lines in Session	34	27	17
Avg. #lines in Session	4.71	4.66	4.75
Avg. #words in Narrative	25.04	24.86	24.18

3. Problem Formulation and Dataset

3.1. Problem Formulation

Suppose that we have a dataset $\mathcal{D}$, in which a sample is represented as (y,c,p,r), where $c = \{s_1, \cdots, s_n\}$ represents a *context* formed by the preceding sentences/lines $\{s_i\}_{i=1}^{n}$; p is a predefined *narrative* that governs the whole script session, and r is a next line candidate (we refer to it as a *response*); $y \in \{0,1\}$ is a binary label, indicating whether r is a proper response for the given c and p. Intuitively, a proper response should be relevant to the context, and be coherent and aligned with the narrative. Our goal is to learn a model $g(c,p,r)$ with $\mathcal{D}$ to determine how suitable a response r is to the given context c and narrative p.

3.2. Data Collection and Construction

Data is a critical issue in research on story/dialogue generation. Unfortunately, no dataset has been created for narrative-guided story/dialogue generation. To fill the gap, we constructed a test collection from GraphMovie, where an editor or a user can retell the story of a movie by uploading descriptive paragraphs in his/her own words to describe the screenshots selected from the movie. Each movie on this website has, on average, 367 descriptions. A description often contains one to three sentences to summarize a fragment of a movie. It can be at different levels - from retelling the same conversations to a high-level description. We consider these descriptions as narratives for a sequence of dialogues, which we call a session in this paper. Each dialogue in a session is called a line of script (or simply a line).

To construct the dataset, we use the top 100 movies on IMDB as an initial list.[17] For each movie, we collect its descriptions from GraphMovie. Then we hire annotators to watch the movie and annotate the start time and end time of the dialogues corresponding to each description through an annotation tool specifically developed for this purpose. According to the start and end time, the sequence of lines is extracted from the subtitle file and aligned with the corresponding description.

As viewers of a movie can upload descriptions freely, not all descriptions correspond to a narrative and are suitable for our task. For example, some uploaded paragraphs express one's subjective opinions about the movie, the actors, or simply copy the script. Therefore, we manually review the data and remove such nonnarrative data. We also remove sessions that have less than two lines. Finally, we obtain 16,109 script sessions, each of which contains a description (narrative) and corresponding lines of the script. As shown in Table 10, on average, a narrative has about 25 words, and a session has 4.7 lines. The maximum number of lines in a session is 34.

Our task is to select one response from a set of candidates at any point during the session. By moving the prediction point through the session, we obtain a set of micro-sessions, each of which has a sequence of previous lines as the context at that point of time, the same narrative as the session, and the next line to predict. The candidates to be selected contain one ground-truth line–the one that is genuinely the next line, together with one (in the training set) or nine (in the validation/test set) other candidates retrieved with the previous lines by Solr.[18] The above preparation of the dataset follows the practice in the literature [**111, 205, 230**] for retrieval-based dialogue.

4. Proposed Method: ScriptWriter

4.1. Overview

A good response is required to be both coherent with the previous lines, *i.e.*, context, and consistent with the given narrative. For example, "Just stay a little longer" can respond "Mama's going to worry about me" and it has no conflict with the narrative in Figure 17. Furthermore, as our target is to generate all lines in the fragment successively, it is also required that the following lines should convey the information that the former lines have not conveyed. Otherwise, only a part of the narrative is covered, and we will miss some other aspects specified in the narrative.

We propose an attention-based model called `ScriptWriter-CPre` to solve the problem. The overview of our model is shown in Figure 18. `ScriptWriter-CPre` follows a representation-matching-aggregation framework. First, the narrative, the context, and the

[17] IMDB, `https://www.imdb.com/`

[18] Apache Solr, `https://lucene.apache.org/solr/`

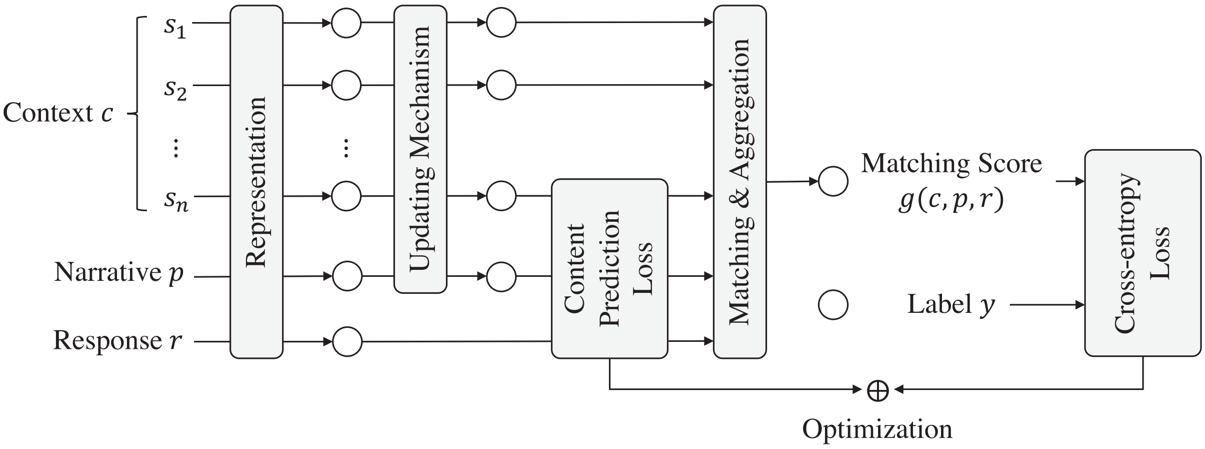


Fig. 18. The overview of our proposed `ScriptWriter-CPre`. `ScriptWriter-CPre` follows a representation-matching-aggregation framework. It is equipped with an updating mechanism to track what in a narrative has been expressed, based on which the representation of the narrative is updated. Besides, we also design a supplementary loss to facilitate the learning of content prediction.

response candidate are represented in multiple granularities by multi-level attentive blocks. Second, we propose an updating mechanism to keep track of what in a narrative has been expressed and explicitly lower their weights in the updated narrative so that more emphasis can be put on the remaining parts. Third, we propose using a supplementary loss to facilitate the learning of content prediction, which lets the model learn to predict which part of the narrative should be given more attention in the next line. Fourth, matching features are extracted between different elements: between context and response to capture whether it is a proper reply; between narrative and response to capture whether it is consistent with the narrative; and between context and narrative to implicitly track what in the narrative has been expressed in the previous lines. Finally, the above matching features are concatenated together and a final matching score is produced by convolutional neural networks (CNNs) and a multi-layer perceptron (MLP). The whole model is optimized by the combination of the supplementary content prediction loss and the final response selection loss.

4.2. Representation

To better handle the gap in words between two word sequences, we propose to use an attentive block, which is similar to that used in Transformer [**186**]. This structure is used to represent lines in the context, a narrative, and a response.

As illustrated in Figure 19, an attentive block contains two sub-layers: a sub-layer implementing an attention mechanism and a fully-connected feed-forward network (FFN). Each layer uses a residual connection to ease the training of networks [**68**] and layer normalization (LayerNorm) to avoid gradient vanishing and exploding [**6**]. The input of an attentive block

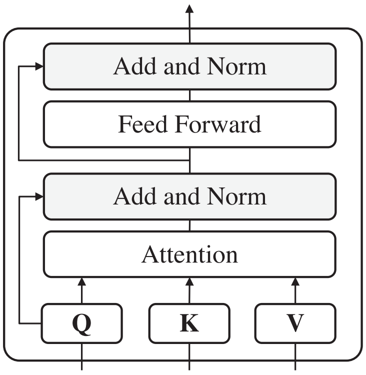


Fig. 19. The structure of the Attentive Block. The attention weights are computed by the query (**Q**) and key (**K**), and then applied to the value (**V**). **Q**, **K**, and **V** are three matrices. Residual connection, layer normalization and a feed-forward network are used to further fuse the information and output final results.

consists of three sequence representations, namely query (**Q**), key (**K**), and value (**V**).[19] The output is a new representation of query and is denoted as AttentiveBlock(**Q**,**K**,**V**) in the remaining parts. The detailed computation is as follows:

$$\mathrm{AttentiveBlock}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{LayerNorm}(\mathbf{X} + \mathrm{FFN}(\mathbf{X})), \tag{4.1}$$

$$\mathbf{X} = \mathrm{LayerNorm}(\mathbf{Q} + \mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V})), \tag{4.2}$$

$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) \cdot \mathbf{V}, \tag{4.3}$$

$$\mathrm{FFN}(\mathbf{X}) = \mathrm{FC}(\mathrm{ReLU}(\mathrm{FC}(\mathbf{X}))), \tag{4.4}$$

where FC($\cdot$) is the fully-connected layer, and ReLU($\cdot$) is the activation function. $\mathbf{Q} \in \mathbb{R}^{N_q \times d}$ denotes the attention query matrix, $\mathbf{K} \in \mathbb{R}^{N_k \times d}$ is the key matrix, and $\mathbf{V} \in \mathbb{R}^{N_k \times d}$ is the value matrix. N_q, N_k, and d are the number of attention query, key/value vectors, and the dimensions of the representation, respectively. The attentive block can be explained as follows: for each attention query vector in **Q**, it first computes the dot product of the attention query with all keys, aiming to evaluate the similarity between the attention query and each key. Then, it is divided each by $\sqrt{d}$, and applies a softmax function to obtain the weights of the values. Finally, the new representation of the attention query vector is calculated as a weighed sum of values. The original Transformer framework is used for machine translation, where a self-attention is proposed to better represent a sentence. In our case, self-attention is used to recognize the internal connections between words in an utterance. So, query, key, and value become the same utterance. Our purpose of doing intra-sentence self-attention is to connect a word to related words in the sentence so that its representation can be enhanced by the related words in the higher layer.

[19] Note that the "query" in the attention function and IR have different meanings.

More specifically, given a narrative $p = (w_1^p, \cdots, w_{n_p}^p)$, a line $s_i = (w_1^{s_i}, \cdots, w_{n_{s_i}}^{s_i})$, and a response candidate $r = (w_1^r, \cdots, w_{n_r}^r)$, `ScriptWriter` first looks up a pre-trained embedding table $\mathbf{E}$ and maps each word w into a d_e-dimension embedding $\mathbf{e}$:

$$\mathbf{e}_i^p = \text{LookUp}(w_i^p, \mathbf{E}), \quad \mathbf{e}_j^{s_i} = \text{LookUp}(w_j^{s_i}, \mathbf{E}), \quad \mathbf{e}_k^r = \text{LookUp}(w_k^r, \mathbf{E}). \tag{4.5}$$

Thus, the narrative p, the line s_i, and the response candidate r are represented by matrices $\mathbf{P}^0 = (\mathbf{e}_1^p, \cdots, \mathbf{e}_{n_p}^p)$, $\mathbf{S}_i^0 = (\mathbf{e}_1^{s_i}, \cdots, \mathbf{e}_{n_{s_i}}^{s_i})$ and $\mathbf{R}^0 = (\mathbf{e}_1^r, \cdots, \mathbf{e}_{n_r}^r)$.

Then `ScriptWriter` takes $\mathbf{P}^0$, $\{\mathbf{S}_i^0\}_{i=1}^n$ and $\mathbf{R}^0$ as inputs and uses stacked attentive blocks to construct multi-level self-attention representations. The output of the $(l-1)^{\text{th}}$ level of attentive block is input into the l^{th} level. The representations of p, s_i, and r at the l^{th} level are defined as follows:

$$\mathbf{P}^l = \text{AttentiveBlock}(\mathbf{P}^{l-1}, \mathbf{P}^{l-1}, \mathbf{P}^{l-1}), \tag{4.6}$$

$$\mathbf{S}_i^l = \text{AttentiveBlock}(\mathbf{S}_i^{l-1}, \mathbf{S}_i^{l-1}, \mathbf{S}_i^{l-1}), \tag{4.7}$$

$$\mathbf{R}^l = \text{AttentiveBlock}(\mathbf{R}^{l-1}, \mathbf{R}^{l-1}, \mathbf{R}^{l-1}), \tag{4.8}$$

where l ranges from 1 to L.

Inspired by previous studies [**223, 237**], we apply another group of attentive blocks, which is referred to as cross-attention, to capture the semantic dependency between p, s_i, and r. Considering p and s_i at first, their cross-attention representations are defined by:

$$\overline{\mathbf{P}}_{s_i}^l = \text{AttentiveBlock}(\mathbf{P}^{l-1}, \mathbf{S}_i^{l-1}, \mathbf{S}_i^{l-1}), \tag{4.9}$$

$$\overline{\mathbf{S}}_{i,p}^l = \text{AttentiveBlock}(\mathbf{S}_i^{l-1}, \mathbf{P}^{l-1}, \mathbf{P}^{l-1}). \tag{4.10}$$

Here, the words in the narrative can attend to all words in the line, and vice verse. In this way, some inter-dependent segment pairs, such as "stay" in the line and "go home later" in the narrative, become closer to each other in the representations. Similarly, we compute cross-attention representations between p and r and between r and s_i at different levels, which are denoted as $\overline{\mathbf{P}}_r^l$, $\overline{\mathbf{R}}_p^l$, $\overline{\mathbf{S}}_{i,r}^l$ and $\overline{\mathbf{R}}_{s_i}^l$:

$$\overline{\mathbf{P}}_r^l = \text{AttentiveBlock}(\mathbf{P}^{l-1}, \mathbf{R}^{l-1}, \mathbf{R}^{l-1}), \tag{4.11}$$

$$\overline{\mathbf{R}}_p^l = \text{AttentiveBlock}(\mathbf{R}^{l-1}, \mathbf{P}^{l-1}, \mathbf{P}^{l-1}), \tag{4.12}$$

$$\overline{\mathbf{S}}_{i,r}^l = \text{AttentiveBlock}(\mathbf{S}_i^{l-1}, \mathbf{R}^{l-1}, \mathbf{R}^{l-1}), \tag{4.13}$$

$$\overline{\mathbf{R}}_{s_i}^l = \text{AttentiveBlock}(\mathbf{R}^{l-1}, \mathbf{S}_i^{l-1}, \mathbf{S}_i^{l-1}). \tag{4.14}$$

These representations further provide matching information across different elements in the next step.

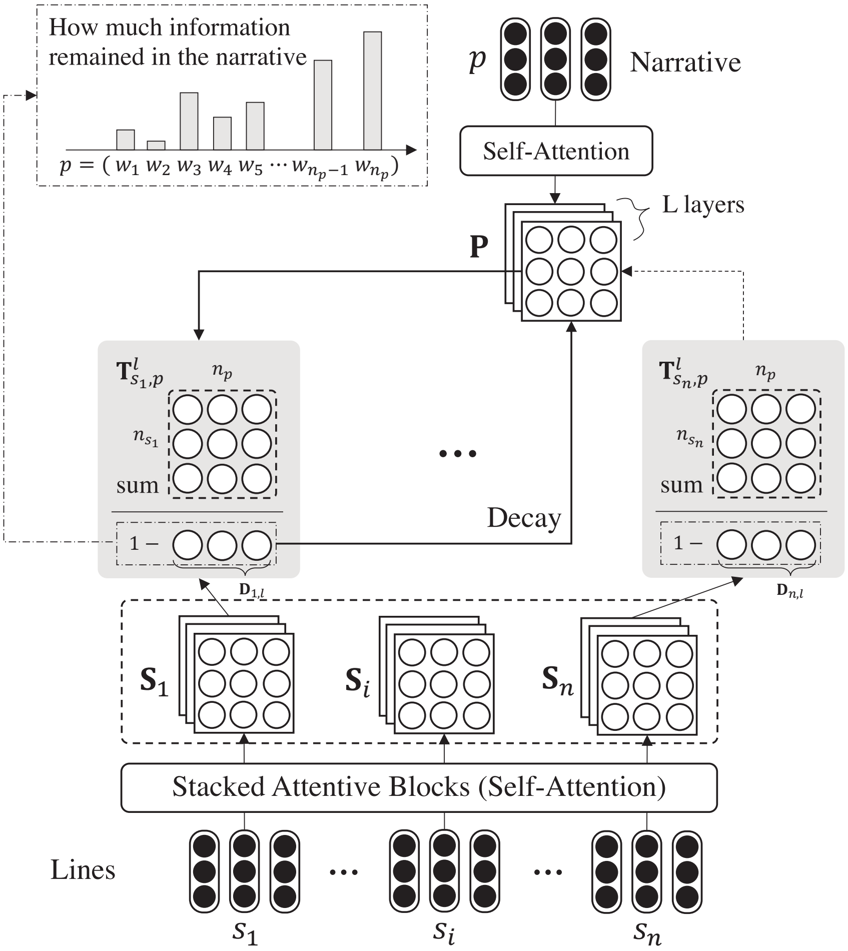


Fig. 20. Updating mechanism in `ScriptWriter`. The narrative and all lines in the context are first represented by the stacked attentive blocks. Then, the representation of the narrative is updated by lines in the context one by one. The information that has been expressed is decayed. As a result, the updated narrative focuses more on the remaining information.

4.3. Updating Mechanism

We design an updating mechanism to keep track of the coverage of the narrative by the lines so that the selection of the response will focus on the uncovered parts. The mechanism is illustrated in Figure 20. We update a narrative's representation gradually by all lines in the context one by one. For the i^{th} line s_i, we conduct a matching between $\mathbf{S}_i$ and $\mathbf{P}$ by their cosine similarity at all levels (l) of attentive blocks:

$$\mathbf{T}^l_{s_i,p}[j][k] = \cos(\mathbf{S}^l_i[j], \mathbf{P}^l[k]), \tag{4.15}$$

where j and k stand for the j^{th} word in s_i and k^{th} word in p respectively. To summarize how much information in p has been expressed by s_i, we compute a vector $\mathbf{D}_i$ by conducting summations along vertical axis on each level in the matching map $\mathbf{T}_{s_i,p}$. The summation at

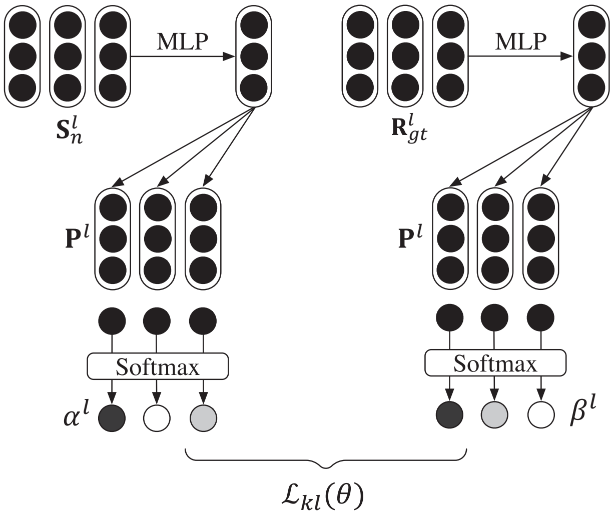


Fig. 21. A supplementary loss for the learning of content prediction. In the left side, for the the last line s_n, we compute its attention distribution over the narrative. This can be viewed as a "predicted" weight indicating which part of the content in the narrative should be focused on. To facilitate the prediction, in the right side, we use the ground-truth response r_{gt} to compute another attention distribution over the narrative. These attention weights reflect which part of the content in the narrative is indeed useful in selecting the ground-truth response. Therefore, we use the latter attention weights as the label and compute a KL-divergence loss to optimize the former predicted weights. By this means, the learning of content prediction in the narrative can be improved.

the l^{th} level is:

$$\mathbf{D}_i^l = [d_{i,1}^l, d_{i,2}^l, \cdots, d_{i,n_p}^l], \quad d_{i,k}^l = \gamma \sum_{j=1}^{n_{s_i}} \mathbf{T}_{s_i,p}^l[j][k], \tag{4.16}$$

where n_p, n_{s_i} denotes the number of words in p and s_i; $\gamma \in [0,1]$ is a parameter to learn and works as a gate to control the decaying degree of the mentioned information. Finally, we update the narrative's representation as follows for the i^{th} line s_i in the context:

$$\mathbf{P}_{i+1}^l = (1 - \mathbf{D}_i^l)\mathbf{P}_i^l. \tag{4.17}$$

The initial representation $\mathbf{P}_0^l$ is equal to $\mathbf{P}^l$ defined in Equation (4.6). As a result, if there are n lines in the context, this update is executed n times, and $(1 - \mathbf{D}^l)$ will produce a continuous decaying effect.

After the updating mechanism, the updated representation of the narrative will be used in the following operations.

4.4. Content Prediction

Inspired by recent studies on knowledge selection in knowledge-grounded dialogue [**90, 104**], we propose using a supplementary loss to facilitate the learning of content prediction (as shown in Figure 21). We define a task to predict the part of the narrative to which we should pay more attention in the next line. Given the last line s_n, we can predict a prior attention distribution over the narrative. However, this prior distribution cannot be effectively learned as no "correct distribution" is given as supervision. To tackle this problem, we propose using a posterior distribution over the narrative to supervise the process. This posterior distribution is computed by the ground-truth response and the narrative, thus can provide valuable information for response selection.

Specifically, `ScriptWriter-CPre` first computes the sentence representation of the last line s_n:

$$\hat{\mathbf{S}}^l_n = \mathrm{MLP}([\mathbf{S}^l_{n,1}, \cdots, \mathbf{S}^l_{n,n_{s_n}}]), \tag{4.18}$$

where $\mathbf{S}^l_{n,i}$ is the representation of the i^{th} word in s_n at the l^{th} layer of the attentive block. Then the prior attention distribution is computed as:

$$\alpha^l_i = \frac{\exp(\hat{\mathbf{S}}^l_n \cdot \mathbf{P}^l_i)}{\sum_{j=1}^{n_p} \exp(\hat{\mathbf{S}}^l_n \cdot \mathbf{P}^l_j)}, \tag{4.19}$$

where α^l_i is the attention weight of the i^{th} word in the narrative at the l^{th} layer. This distribution reflects how attention should be assigned to the narrative from the perspective of the last line.

Similarly, the posterior distribution can be computed with the ground-truth response r_{gt} as:

$$\beta^l_i = \frac{\exp(\hat{\mathbf{R}}^l_{\text{gt}} \cdot \mathbf{P}^l_i)}{\sum_{j=1}^{n_p} \exp(\hat{\mathbf{R}}^l_{\text{gt}} \cdot \mathbf{P}^l_j)}, \tag{4.20}$$

where $\hat{\mathbf{R}}^l_{\text{gt}}$ is the sentence representation of r_{gt} computed by the same process as Equation (4.18). This posterior attention distribution reflects which part of the narrative is useful for selecting the ground-truth response. So, it is desirable to distribute attention based on the posterior distribution. However, this latter is unknown during inference as no ground-truth response is given. Therefore, we propose to approximate the posterior distribution using the prior distribution so that `ScriptWriter-CPre` is capable of distributing appropriate attention even without posterior information. To this end, we introduce an auxiliary loss, namely the Kullback-Leibler divergence loss ($\mathcal{L}_{\text{kl}}$), to help optimize the model. The KL divergence loss is commonly used to measure the proximity between the prior distribution and the posterior distribution, which is computed as follows:

$$\mathcal{L}_{\text{kl}}(\theta) = -\sum_{(y,c,p,r)\in\mathcal{D}} \sum_{l=0}^{L} \sum_{i=1}^{n_p} \beta^l_i \log \frac{\beta^l_i}{\alpha^l_i}, \tag{4.21}$$

where θ denotes the model parameters.

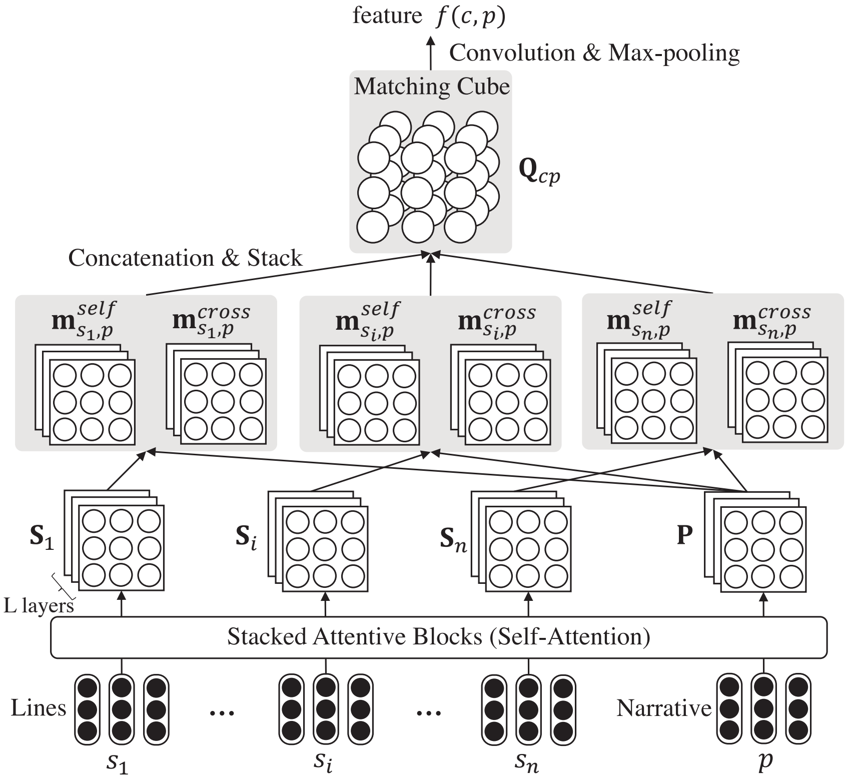

feature $f(c,p)$
Convolution & Max-pooling
Matching Cube
$\mathbf{Q}_{cp}$
Concatenation & Stack
$\mathbf{m}_{s_1,p}^{self}$
$\mathbf{m}_{s_1,p}^{cross}$
$\mathbf{m}_{s_i,p}^{self}$
$\mathbf{m}_{s_i,p}^{cross}$
$\mathbf{m}_{s_n,p}^{self}$
$\mathbf{m}_{s_n,p}^{cross}$
$\mathbf{S}_1$
$\mathbf{S}_i$
$\mathbf{S}_n$
$\mathbf{P}$
L layers
Stacked Attentive Blocks (Self-Attention)
Lines
Narrative
s_1
s_i
s_n
p

Fig. 22. The context-narrative matching. All lines and the narrative are represented by attentive blocks and the matching between them results in a matching cube $\mathbf{Q}_{cp}$. Matching features are extracted, aggregated, and distilled by a CNN with max-pooling operation.

When minimizing $\mathcal{L}_{\text{kl}}$, the posterior distribution β_i^l can be regarded as labels and `ScriptWriter-CPre` is trained to use the prior distribution α_i^l to approximate β_i^l. The representations of the last line and the narrative are directly used in computing the prior distribution. As a consequence, with $\mathcal{L}_{\text{kl}}$, the parameters of `ScriptWriter-CPre`, especially those for computing the last line's and narrative's representations, can be better tuned. It is worth noting that $\mathcal{L}_{\text{kl}}$ only works in training phase, thus the ground-truth response is not need in inference phase.

4.5. Matching

In the previous steps, we obtained the representations of all lines in the context, the representation of the response, and the updated representation of the narrative. In this step, we construct several matching maps and extract matching features based on these representations.

The matching between the narrative p and the line s_i is conducted based on both their self-attention and cross-attention representations, as shown in Figure 22.

First, `ScriptWriter` computes the dot product on these two representations separately as follows:

$$\mathbf{m}^{\text{self}}_{s_i,p,l}[j,k] = \mathbf{S}^l_i[j]^\top \cdot \mathbf{P}^l[k], \quad \mathbf{m}^{\text{cross}}_{s_i,p,l}[j,k] = \overline{\mathbf{S}}^l_{i,p}[j]^\top \cdot \overline{\mathbf{P}}^l_{s_i}[k], \tag{4.22}$$

where l ranges from 0 to L. Each element is the dot product of the j^{th} word representation in $\mathbf{S}^l_i$ or $\overline{\mathbf{S}}^l_{i,p}$ and the k^{th} word representation in $\mathbf{P}^l$ or $\overline{\mathbf{P}}^l_{s_i}$. Then the matching maps in different layers are concatenated together as follows:

$$\mathbf{m}^{\text{self}}_{s_i,p}[j,k] = \left[\mathbf{m}^{\text{self}}_{s_i,p,0}\,[j,k] \oplus \cdots \oplus \mathbf{m}^{\text{self}}_{s_i,p,L}\,[j,k]\right],$$
$$\mathbf{m}^{\text{cross}}_{s_i,p}[j,k] = \left[\mathbf{m}^{\text{cross}}_{s_i,p,0}\,[j,k] \oplus \cdots \oplus \mathbf{m}^{\text{cross}}_{s_i,p,L}\,[j,k]\right],$$

where $\oplus$ is concatenation operation. Finally, the matching features computed by the self-attention representation and the cross-attention representation are fused as follows:

$$\mathbf{M}_{s_i,p}\,[j,k] = \left[\mathbf{m}^{\text{self}}_{s_i,p}\,[j,k] \oplus \mathbf{m}^{\text{cross}}_{s_i,p}\,[j,k]\right].$$

The matching matrices $\mathbf{M}_{p,r}$ and $\mathbf{M}_{s_i,r}$ for narrative-response and context-response are constructed in a similar way. For the sake of brevity, we omit the formulas. After concatenation, each cell in $\mathbf{M}_{s_i,p}$, $\mathbf{M}_{p,r}$ or $\mathbf{M}_{s_i,r}$ has $2(L+1)$ channels and contains matching information at different levels.

The matching between narrative, context, and response serves for different purposes. Context-response matching ($\mathbf{M}_{s_i,r}$) serves to select a response suitable for the context. Context-narrative matching ($\mathbf{M}_{s_i,p}$) helps model "remember" how much information has been expressed and implicitly influences the selection of the next response. Narrative-response matching ($\mathbf{M}_{p,r}$) helps the model select a more consistent response with the narrative. As the narrative keeps being updated along with the lines of the context, `ScriptWriter` tends to dynamically choose the response that matches what remains unexpressed in the narrative.

4.6. Aggregation

To further use the information across two consecutive lines, `ScriptWriter` piles up all context-narrative matching matrices and all context-response matching matrices to construct two cubes $\mathbf{Q}_{cp} = \{\mathbf{M}_{s_i,p}[j,k]\}^n_{i=1}$ and $\mathbf{Q}_{cr} = \{\mathbf{M}_{s_i,r}[j,k]\}^n_{i=1}$, where n is the number of lines in the session. Then `ScriptWriter` employs 3D convolution to distill important matching features from the whole cube. We denote these two feature vectors as $f(c,p)$ and $f(c,r)$. For narrative-response matching, `ScriptWriter` conducts 2D convolution on $\mathbf{M}_{p,r}$ to distill matching features between the narrative and the response, denoted as $f(p,r)$.

The three types of matching features are concatenated together, and the matching score $g(c,p,r)$ for ranking response candidates is computed by an MLP with a sigmoid activation function, which is defined as:

$$f(c,p,r) = [f(c,p) \oplus f(c,r) \oplus f(p,r)], \tag{4.23}$$

$$g(c,p,r) = \text{sigmoid}\Big(\text{MLP}(f(c,p,r)\Big). \tag{4.24}$$

4.7. Model Training

`ScriptWriter-CPre` learns $g(c,p,r)$ by minimizing cross entropy with $\mathcal{D}$. The objective function is formulated as:

$$\mathcal{L}_{\text{ce}}(\theta) = -\sum_{(y,c,p,r)\in\mathcal{D}} [y \log(g(c,p,r)) + (1-y)\log(1-g(c,p,r))]. \tag{4.25}$$

The two losses are combined to tune the model with a hyperparameter λ to control their effect:

$$\mathcal{L}(\theta) = \lambda\mathcal{L}_{\text{ce}}(\theta) + (1-\lambda)\mathcal{L}_{\text{kl}}(\theta). \tag{4.26}$$

5. Experiments

5.1. Evaluation setup

As presented in Table 10, we randomly split the the *GraphMovie* collection into training, validation, and test sets. The split ratio is 18:1:1. We split the sessions into micro-sessions: given a session with n lines in the context, we will split it into n micro-sessions with length varying from 1 to n. These micro-sessions share the same narrative. By doing this, the model is asked to learn to select one line as the response from a set of candidates at any point during the session, and the dataset, in particular for training, can be significantly enlarged.

We conduct two kinds of evaluation as follows:

Turn-level task asks a model to rank a list of candidate responses based on its given context and narrative for a micro-session. The model then selects the best response for the current turn. This setting is similar to the widely studied response selection task [**205, 230, 237**]. We follow these previous studies and employ recall at position k in n candidates (R_n@k) and mean reciprocal rank (MRR) [**188**] as evaluation metrics. For example, R_{10}@1 means recall at one when we rank ten candidates (one positive sample and nine negative samples). The final results are the average numbers over all micro-sessions in the test set. Note that among different Recall metrics, R_2@1 and R_{10}@1 are two most relevant ones to our task, because there is only one positive line in our dataset. They correspond to the scenarios in the training and test sets, respectively.

Session-level task aims to predict all the lines in a session gradually. It starts with the first line of the session as the context and the given narrative and predicts the best next line. The predicted line is then incorporated into the context to predict the next line. This process continues until the last line of the session is selected. Finally, we calculate precision over the whole original session and report average numbers over all sessions in the test set. Precision is defined as the number of correct selections divided by the number of lines in a session. We consider two measures: 1) $\mathrm{P}_{\mathrm{strict}}$ which accepts a right response at the right position; 2) $\mathrm{P}_{\mathrm{weak}}$ which accepts a right response at any position.

5.2. Baselines

As no previous work has been done on narrative-based script generation, no proper baseline exists. Nevertheless, some existing multi-turn conversation models based on context can be adapted to work with a narrative: the context is simply extended with the narrative. Two different extension methods have been tested: the narrative is added into the context together with the previous lines; the narrative is used as a second context. In the latter case, two matching scores are obtained for context-narrative and narrative-response. They are aggregated through a fully-connected layer to produce a final score. This second approach turns out to perform better. Therefore, we only report the results with this latter method.[20]

(1) `MVLSTM` [**191**]: this model concatenates all previous lines as a context document and leverages an LSTM to obtain positional representations for all words in the document and the response candidate. Then the interactions between them at different positions are modeled by cosine similarity, resulting in a matching map. The matching features are extracted with a k-max pooling layer and aggregated as a matching score with an MLP. To incorporate a narrative into this model, we conduct the same operation to compute the matching score between the narrative and a response candidate. Finally, the two scores are combined as a final matching score with another MLP. This model considers all positional information in the sentence, but only the top k values in the matching map are used.

(2) `DL2R` [**217**]: the model reformulates the last line with other lines in a context with multiple approaches. The reformulated line and a response candidate are then represented by a composition of an RNN and a CNN. The matching score is computed in a similar way as `MVLSTM`. We use the same RNN and CNN to represent the narrative and compute a matching score between the context and the narrative, which is further combined with the context-response matching score to output a final score with an MLP. In this model, all previous lines in the context are used to reformulate the last line, thus the context-response matching is neglected.

(3) `SMN` [**205**]: it matches each line with the response sequentially, and then transforms each line-response pair into a matching vector with CNNs. The matching vectors are aggregated with an RNN as a matching score of the context and the response candidate. We apply

[20] We also tested some basic models such as RNN, LSTM, and BiLSTM [**111**] in our experiments. However, they cannot achieve comparable results to the selected baselines.

the same CNNs to obtain a matching vector for each line-narrative pair and use an RNN to compute a matching score. Two matching scores are finally combined by an MLP. This model is not equipped with an attention mechanism, which may provide better representations.

(4) DAM [**237**]: it represents a context and a response by conducting a self-attention and a cross-attention operation on them. It uses CNNs to extract features and uses an MLP to get a score. Similar to SMN, we perform the same operation to obtain a matching score of the context and the narrative and combine the score with context-response matching score by an MLP. Different from our model, this model only considers the context-response matching and does not track what in the narrative has already been expressed by the previous lines, *i.e.*, context.

(5) DUA [**230**]: the model concatenates the last line with each previous line in the context and response, respectively. Then it performs a self-attention operation to get the refined representations for both the context and the response, based on which matching features are extracted with CNNs and RNNs. Finally, it uses an MLP to get a matching score. We apply the same self-attention operation to get the refined representation of a narrative. Then we extract the matching features between the refined response and the narrative. Finally, both groups of matching features are aggregated by an RNN to get a final score. This model uses RNNs to represent sentences and aggregate the matching information, which is different from our model.

(6) IMN [**56**]: the model uses an attentive hierarchical recurrent encoder, which is capable of encoding sentences hierarchically and aggregating them with an attention mechanism to produce more descriptive representations. Then the bidirectional interactions between the whole multi-turn context and the response candidates are calculated to derive the matching information between them. We apply the same structure to derive the matching features between the narrative and the response. These features are fused with context-response matching features to output a final score with an MLP.

(7) IOI [**178**]: the model performs matching by stacking multiple interaction blocks in which the residual information from one step of interaction initiates the interaction process again. Therefore, the matching information within a line-response pair is extracted from the interaction of the pair iteratively, and the information flows along the chain of blocks via representation. To leverage the narrative, we use the same multiple interaction blocks to extract matching information within the narrative-response pair and compute a matching score. The two scores are added together to select the response.

(8) MSN [**223**]: this model first utilizes a multi-hop selector to select the relevant lines as context. Then, the model matches the filtered context with the candidate response based on their self-attention and cross-attention representations. Next, the matching features are extracted by a CNN and aggregated by an LSTM. The final matching score is computed by

an MLP. The matching process is similar to `SMN`, thus we use a similar way to incorporate the narrative.

(9) `ScriptWriter` [**249**]: This is the previous model we proposed. This model does not have the narrative representation optimization module, and the other structure is similar to `ScriptWriter-CPre` proposed in this paper.

5.3. Training Details

All models are implemented in Tensorflow.[21] Word embeddings are pre-trained by Word2vec [**122**] on the training set with 200 dimensions. We test the stack number in {1,2,3} and report our results with three stacks. Due to the limited resources, we cannot conduct experiments with a larger number of stacks, which could be tested in the future. Two 3D convolutional layers both have 32 filters, respectively. They both use [3,3,3] as the kernel size, and the max-pooling size is [3,3,3]. Two 2D convolutional layers of narrative-response matching both have 32 filters with [3,3] as the kernel size. The max-pooling size is also [3,3]. All parameters are optimized with Adam optimizer [**91**]. The learning rate is 0.001 and decreases during training. The initial value of γ is 0.5. The batch size is 64. We use the validation set to select the best models and report their performance on the test set. The maximum number of lines in the context is set as ten, and the maximum length of a line, response, and narrative sentence is all set as 50. All sentences are zero-padded to the maximum length. We also padded zeros if the number of lines in a context is less than 10. Otherwise, we kept the latest ten lines. The dataset and the source code of our model are available on GitHub.[22]

5.4. Evaluation Results

5.4.1. Automatic Metrics. The experimental results are reported in Table 11. The results on both turn-level and session-level evaluations indicate that `ScriptWriter` dramatically outperforms all baselines, including `MSN` and `IOI`, which are two state-of-the-art models for multi-turn response selection. Most improvements are statistically significant (p-value $\leq$ 0.01). `MSN` and `DAM` perform better than other baselines, indicating the effectiveness of the self- and cross-attention mechanism used in this model. `IOI`, `IMN`, and `DUA` also apply the attention mechanism. They outperform the other baselines that do not use attention. Both observations confirm the advantage of using attention mechanisms over pure RNN (such as `SMN`, `DL2R`, and `MVLSTM`).

In terms of session-level evaluations, `ScriptWriter-CPre` achieves 1.9% and 2.6% absolute improvements over the best results obtained by the baseline methods. This demonstrates

[21] Tensorflow, `https://www.tensorflow.org`

[22] Our project, `https://github.com/DaoD/ScriptWriter`

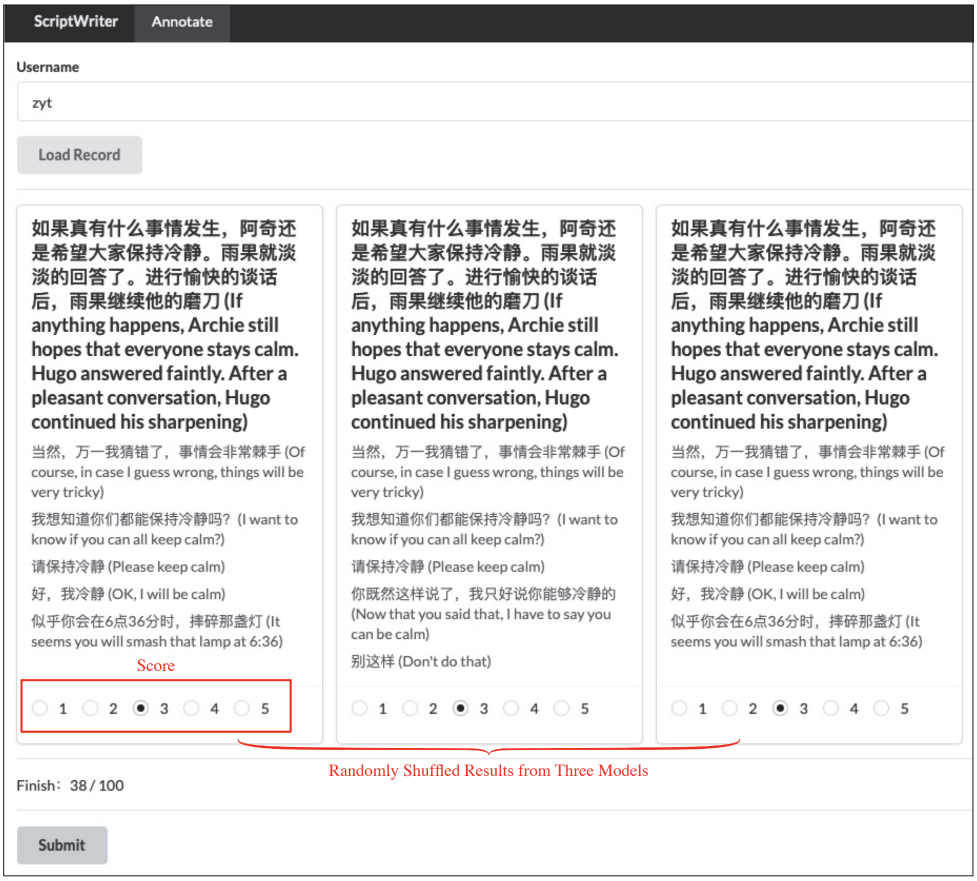
Table 11. Evaluation results on two response selection tasks: turn-level and session-level. The turn-level evaluation aims at measuring the performance of models on predicting a specific line in a session, while the session-level evaluation considers the quality of the whole session. † and ⋆ denote significant differences between each baseline and `ScriptWriter-CPre` measured in t-test with $p \leq 0.01$ and $p \leq 0.05$, respectively.

	Turn-level					Session-level	
	$\mathbf{R_2@1}$	$\mathbf{R_{10}@1}$	$\mathbf{R_{10}@2}$	$\mathbf{R_{10}@5}$	**MRR**	$\mathbf{P_{strict}}$	$\mathbf{P_{weak}}$
`MVLSTM`	$0.651^\dagger$	$0.217^\dagger$	$0.384^\dagger$	$0.732^\dagger$	$0.395^\dagger$	$0.198^\dagger$	$0.224^\dagger$
`DL2R`	$0.643^\dagger$	$0.210^\dagger$	$0.321^\dagger$	$0.638^\dagger$	$0.314^\dagger$	$0.230^\dagger$	$0.243^\dagger$
`SMN`	$0.641^\dagger$	$0.176^\dagger$	$0.333^\dagger$	$0.696^\dagger$	$0.392^\dagger$	$0.197^\dagger$	$0.236^\dagger$
`DAM`	$0.631^\dagger$	$0.240^\dagger$	$0.398^\dagger$	$0.733^\dagger$	$0.408^\dagger$	$0.226^\dagger$	$0.236^\dagger$
`DUA`	$0.654^\dagger$	$0.237^\dagger$	$0.403^\dagger$	$0.736^\dagger$	$0.396^\dagger$	$0.223^\dagger$	$0.251^\dagger$
`IMN`	$0.686^\dagger$	$0.301^\dagger$	$0.450^\dagger$	$0.759^\dagger$	$0.463^\dagger$	$0.304^\dagger$	$0.325^\dagger$
`IOI`	$0.710^\dagger$	$0.341^\dagger$	$0.491^\dagger$	$0.774^\dagger$	$0.464^\dagger$	$0.324^\dagger$	$0.337^\dagger$
`MSN`	$0.724^\dagger$	$0.329^\dagger$	$0.511^\dagger$	$0.794^\star$	$0.464^\dagger$	$0.314^\dagger$	$0.346^\dagger$
`ScriptWriter`	$0.730^\dagger$	$0.365^\dagger$	0.537	0.814	0.503	0.373	$0.383^\star$
`ScriptWriter-CPre`	**0.756**	**0.398**	**0.557**	**0.817**	**0.504**	**0.392**	**0.409**

that `ScriptWriter-CPre` can generate a more coherent and consistent script. Besides, between the two session-level measures, we observe that both our `ScriptWriter-CPre` and `ScriptWriter` are less affected when moving from $\mathrm{P_{weak}}$ to $\mathrm{P_{strict}}$. This shows that the two models can better select a response at the right position. We attribute this behavior to the utilization of narrative coverage.

As a retrieval-based method, the selected response (*i.e.*, the one with the highest score) is used for constructing the session, so R@1 is the most important metric. Comparing `ScriptWriter-CPre` to `ScriptWriter`, we can see the content prediction module can improve both $\mathrm{R_2}$@1 and $\mathrm{R_{10}}$@1 significantly. Correspondingly, we can see the performance on session-level evaluation is also improved. These results demonstrate the effectiveness of our proposed content prediction module. We also notice that MRR is only improved by 0.001. This means that, for those samples where the ground-truth response cannot be ranked at the top, the average position of the ground-truth response decreases. From this result, we can speculate that our proposed supplementary loss in content prediction module is helpful when the future lines can be inferred from the last line and the narrative. In the future, we plan to consider more script lines in the context for content prediction.

Besides, it is worth noting that the content prediction module only adds one MLP layer (with a few parameters) to the model, which has less influence on the training cost. However, this has improved the model's performance, especially on session-level evaluation. This suggests that this module is beneficial for our model.



Username

zyt

如果真有什么事情发生，阿奇还是希望大家保持冷静。雨果就淡淡的回答了。进行愉快的谈话后，雨果继续他的磨刀 (If anything happens, Archie still hopes that everyone stays calm. Hugo answered faintly. After a pleasant conversation, Hugo continued his sharpening)

当然，万一我猜错了，事情会非常棘手 (Of course, in case I guess wrong, things will be very tricky)

我想知道你们都能保持冷静吗？(I want to know if you can all keep calm?)

请保持冷静 (Please keep calm)

好，我冷静 (OK, I will be calm)

似乎你会在6点36分时，摔碎那盏灯 (It seems you will smash that lamp at 6:36)

○ 1 ○ 2 ● 3 ○ 4 ○ 5

如果真有什么事情发生，阿奇还是希望大家保持冷静。雨果就淡淡的回答了。进行愉快的谈话后，雨果继续他的磨刀 (If anything happens, Archie still hopes that everyone stays calm. Hugo answered faintly. After a pleasant conversation, Hugo continued his sharpening)

当然，万一我猜错了，事情会非常棘手 (Of course, in case I guess wrong, things will be very tricky)

我想知道你们都能保持冷静吗？(I want to know if you can all keep calm?)

请保持冷静 (Please keep calm)

你既然这样说了，我只好说你能够冷静的 (Now that you said that, I have to say you can be calm)

别这样 (Don't do that)

○ 1 ○ 2 ● 3 ○ 4 ○ 5

如果真有什么事情发生，阿奇还是希望大家保持冷静。雨果就淡淡的回答了。进行愉快的谈话后，雨果继续他的磨刀 (If anything happens, Archie still hopes that everyone stays calm. Hugo answered faintly. After a pleasant conversation, Hugo continued his sharpening)

当然，万一我猜错了，事情会非常棘手 (Of course, in case I guess wrong, things will be very tricky)

我想知道你们都能保持冷静吗？(I want to know if you can all keep calm?)

请保持冷静 (Please keep calm)

好，我冷静 (OK, I will be calm)

似乎你会在6点36分时，摔碎那盏灯 (It seems you will smash that lamp at 6:36)

○ 1 ○ 2 ● 3 ○ 4 ○ 5

Finish：38 / 100

Fig. 23. Illustration of the annotation system. Three sessions generated by different models (`ScriptWriter-CPre`, `ScriptWriter`, and `MSN`) are randomly shuffled and presented to the annotators. The annotators should label them with a score ranging from 5 (the best) to 1 (the worst). In total, 100 generated sessions with different narratives are annotated.

5.4.2. Human Evaluation. Similar to dialogue generation, the ground-truth script lines are not the only suitable ones for a given narrative due to the diversity of human language. Therefore, we also conduct a human evaluation to compare our method with previous baselines. We hire four experienced annotators to annotate 100 generated sessions of `MSN`, `ScirptWriter`, and `ScriptWriter-CPre`. The narrative is randomly selected from the test set, and the first line is also given. As shown in Figure 23, the four annotators are asked to label the randomly shuffled results with scores ranging in [1, 5]. The standard of scoring is based on the errors appearing in session coherence and narrative consistence, which is shown as follows:

- 5: The generated lines are coherent and consistent with the narrative perfectly (0 error in coherence, 0 error in consistence).
- 4: The generated lines are coherent but have one line that is not consistent with the narrative (0 error in coherence, 1 error in consistence).

Table 12. Examples of annotation results. For each narrative, the results from three different models (`ScriptWriter-CPre`, `ScriptWriter`, and `MSN`) are annotated. The labels given by a human annotator are shown in the last column of the table, which reflect the quality of the session.

Narrative: Maxie stopped the noodles at the entrance of the hospital and said that he would also go with him. Every time they have conflicts, they talk about swimming.		
`ScriptWriter-CPre`	(1) Hey! (2) Hey, let me go first. (3) I think… I should also go the beach. (4) Do you want to go swimming? (5) Yeah, let's go.	5
`ScriptWriter`	(1) Hey! (2) Hey, let me go first. (3) I think… I should also go the beach. (4) Do you want to go swimming? (5) It doesn't matter if you don't want to go. You can wait another five or six years.	4
`MSN`	(1) Hey! (2) Hey, let me go first. (3) Hey, I'm Lucas. (4) Do you want to go swimming? (5) Yeah, let's go. (6) I appreciate God every day.	2
Narrative: Bernstein asked the reporter to talk to Li Lan. Li Lan and Kane had different views on the Spanish-American War.		
`ScriptWriter-CPre`	(1) You should meet Jack Lilan. (2) Of course he and Mr. Kane. (3) Miss Tang Sheng, this is Mr. Kane. (4) Ask you to kill somebody.	3
`ScriptWriter`	(1) You should meet Jack Lilan. (2) Of course he and Mr. Kane. (3) Mr. Kane, I have something to look for you. (4) A little bit of disagreement.	3
`MSN`	(1) You should meet Jack Lilan. (2) Of course he and Mr. Kane. (3) Miss Tang Sheng, this is Mr. Kane. (4) Ask you to kill somebody. (5) I have one thing to say.	2
Narrative: Hickler invited her to dance together, and Napaloni's wife was very happy.		
`ScriptWriter-CPre`	(1) Nobody? I'm honored to invite you. (2) I'm honored too, please sit down and welcome. (3) Are you inviting me?	4
`ScriptWriter`	(1) Nobody? I'm honored to invite you. (2) I'm honored too, please sit down and welcome. (3) Are you inviting me? (4) I love music very much especially Beethoven. (5) Thank you very much. (5) I'm always happy to meet you.	2
`MSN`	(1) Nobody? I'm honored to invite you. (2) Fahan Laiju will come to my office tomorrow to see me. (3) Are you inviting me? (4) Ok, let me do it myself.	4

- 3: There is one line that is not coherent with the context and at most one line is not consistent with the narrative (1 error in coherence, ≤ 1 error in consistence).
- 2: There is at most one line that is incoherent with the context and more than one line are inconsistent with the narrative (≤ 1 error in coherence, ≥ 1 error in consistence).
- 1: There is more than one line that are incoherent or inconsistent with the narrative (≥ 1 error in coherence, ≥ 1 error in consistence).

Some annotation results are shown in Table 12.

Table 13. Human evaluation results. The quality of the whole generated session is evaluated.

(a) Absolute scoring. Annotators are asked to score the generated sessions from different models, respectively. The scores range from 5 (the best) to 1 (the worst). The average score reflects the overall performance of each model.

Model	**1**	**2**	**3**	**4**	**5**	**Average**
`MSN`	**13.50%**	**17.50%**	19.50%	23.00%	26.50%	3.3150
`ScriptWriter`	10.75%	15.75%	23.00%	**26.00%**	24.50%	3.3775
`ScriptWriter-CPre`	8.50%	13.50%	**25.50%**	24.25%	**28.25%**	**3.5025**

(b) Relative preference scoring. Annotators are asked to compare the results from two different models, and label which one is better. "Tie" indicates the quality of the generated sessions from two models are similar.

	Win	**Tie**	**Lose**
`ScriptWriter-CPre` vs. `ScriptWriter`	27.75	51.5	20.75
`ScriptWriter-CPre` vs. `MSN`	33.25	47.0	19.75
`ScriptWriter` vs. `MSN`	28.25	47.5	24.25

Following the recent work [**202**], we compute the Kendall tau-b correlation coefficient to evaluate the agreement between any two annotators, and then we average the Kendall tau scores over the 100 samples and six pairs of annotation results. The coefficient is 0.633, which indicates that annotators have moderate agreement on the scoring order of the generated script lines.

The evaluation results are shown in Table 13a, which are consistent with the automatic evaluation results in general. It is clear that `ScriptWriter-CPre` performs the best (3.5 score on average) among the three models. More specifically, 28.25% of the results generated by our `ScriptWriter-CPre` are perfect and 78% are scored higher than 2. These results demonstrate the effectiveness of our proposed method. `MSN` has 31% results with 1 or 2 score, which is the worst among the three models. Besides, we also run a pair-wise comparison between each pair of models, and this results in Table 13b. The relative preference scores also show that both `ScriptWriter-CPre` and `ScriptWriter` are preferred to `MSN`. This indicates that using a narrative to guide the generation of script is a much more appropriate approach than using it as a part of the general context. Between `ScriptWriter-CPre` and `ScriptWriter`, `ScriptWriter-CPre` wins, indicating that the extension we propose in this paper can further improve the approach.

6. Further Analysis

6.1. Model Ablation

We conduct an ablation study to investigate the impact of different modules in ScriptWriter-CPre. These studies are conducted from different perspectives:

- We investigate the influence of the number of the layers by setting $l = \{1,2,3\}$;
- We validate the effectiveness of our proposed updating mechanism. Specifically, we set $\gamma = 0$ and test the performance of our model. Under this circumstance, the representation of the narrative is not updated but static;
- We test the influence of the cross-attention representations used in our method;
- We explore the effectiveness of narrative-response, context-narrative, and context-response matching by removing them one by one from the entire model;
- Finally, we investigate the influence of our proposed supplementary loss $\mathcal{L}_{\text{kl}}$ for narrative representation optimization. This is achieved by setting $\lambda = 1$.

Model ablation results are shown in Table 14. We have the following findings:

(1) In general, the more layers used, the higher the performance of ScriptWriter-CPre. This result is consistent with existing work, which also shows that multi-layer structures are preferred [**178, 237**]. It is possible that more than three layers can yield even better results. However, due to the computational resources we have, we can only test at most three layers in the experiments. Consumed memory and time increase along with the number of layers. This suggests that adding more layers is a good strategy only if we have the necessary computation power.

(2) ScriptWriter-CPre performs better than it with static narrative representation, demonstrating the effectiveness of the updating mechanism for the narrative. The optimal value of γ is around 0.647 after training, which means that only about 35% of the information in the narrative is kept when a line conveys it.

(3) When cross-attention representations are removed from ScriptWriter-CPre, the performance greatly degrades. This indicates that the cross-attention mechanism can capture useful interaction features between different sources of information such as narrative and response.

(4) In both turn-level and session-level evaluations, the performance drops the most when we remove narrative-response matching. This indicates that the relevance of the response to the narrative is the most useful information in narrative-guided script generation.

(5) When we remove context-narrative matching, the performance drops too, indicating that context-narrative matching may provide implicit and complementary information for controlling the alignment of response and narrative.

Table 14. Ablation results in two response selection tasks: turn-level and session-level. We test our model with different layers of attention blocks. Besides, we also validate the effectiveness of different modules by removing them one by one from our model. These include the updating mechanism for the narrative representation, the cross-attention mechanism, the narrative-response (PR), context-narrative (CP), and the context-response (CR) matching, and the supplementary loss for learning content prediction.

	Turn-level					Session-level	
	R_2@1	**R_{10}@1**	**R_{10}@2**	**R_{10}@5**	**MRR**	**P_{strict}**	**P_{weak}**
`ScriptWriter-CPre`	**0.756**	**0.398**	**0.557**	**0.817**	**0.504**	**0.392**	0.409
0 layer	0.727	0.356	0.525	0.791	0.475	0.346	0.364
1 layer	0.729	0.359	0.533	0.806	0.490	0.358	0.368
2 layers	0.735	0.374	0.541	0.814	0.493	0.368	0.384
w static narrative	0.736	0.367	0.537	0.814	0.495	0.357	0.371
w/o Cross	0.720	0.365	0.532	0.809	0.500	0.357	0.361
w/o PR matching	0.650	0.242	0.401	0.720	0.387	0.210	0.230
w/o CP matching	0.734	0.381	0.537	0.809	0.495	0.375	0.391
w/o CR matching	0.727	0.340	0.493	0.768	0.469	0.361	**0.419**
w/o $\mathcal{L}_{kl}$	0.729	0.367	0.538	0.816	0.496	0.370	0.378

(6) In contrast, when we remove the context-response matching, the performance also drops, however, at a much smaller scale, especially on P_{weak}, than when narrative-response matching is removed. This contrast indicates that narrative is a more useful piece of information than context to determine what should be said next, thus it should be taken into account with an adequate mechanism.

(7) When removing the supplementary loss $\mathcal{L}_{kl}$, the model is similar to the original ScriptWriter proposed in [**249**].[23] We can see the performance drops significantly in both turn-level and session-level evaluations. This result demonstrates the effectiveness of $\mathcal{L}_{kl}$ in optimizing the representation of the narrative.

6.2. Performance with Different λs

To investigate the impact of supplementary loss $\mathcal{L}_{\mathrm{kl}}$ in our model, we vary λ from 0.1 to 1.0 and report R_2@1, R_{10}@1, and MRR results on the validation set in Figure 24. Generally, the performance increases along with the increase of λ and achieves the best at around $\lambda = 0.7$. Thereafter, the performance starts to decrease. When λ is very small, the model is optimized mainly according to $\mathcal{L}_{\mathrm{kl}}$. The performance is not good because the main task, generating script lines, cannot be well-learned. With increased λ, the main task plays a more important role in the optimization, so that the performance improves. However, when $\lambda > 0.7$, the

[23] There are some minor differences in hyperparameters, such as a different number of convolutional filters, leading to slightly different results than ScriptWriter reported in Table 11.

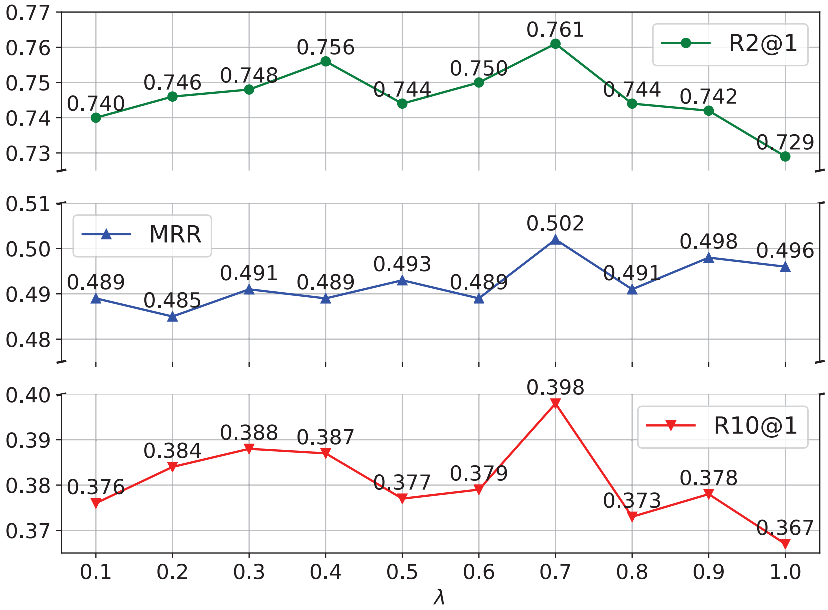


Fig. 24. The performance of `ScriptWriter-CPre` on the validation set with different λs.

performance start to decrease. This is because $\mathcal{L}_{\text{kl}}$ can no longer play a significant role with a very small weight. When $\lambda = 1$, no supplementary loss is used, and we have analyzed this result (at the last point in Section 5.4.1.

6.3. Performance across Narrative Types

As we explained, the narratives in our dataset are contributed by Internet users, and they vary in style. Some narratives are detailed, while others are general. The question we analyze is how general vs. detailed narratives affect the performance of response selection. We use a simple method to evaluate roughly the degree of detail of a narrative: a narrative that has a high lexical overlap with the lines in the session is considered to be detailed. Narratives are put into six buckets depending on their level of detail, as shown in Figure 25.

We plot the performance of `ScriptWriter-CPre`, `ScriptWriter`, and `MSN` in session-level evaluation over different types of narratives. We have the following observations:

(1) The first type "0" means no word overlap between narrative and dialogue sessions. This is the most challenging case, representing extremely general narratives. It is not surprising to see that all models perform poorly on this type compared to other types in terms of P_{strict}. Moreover, we find that `ScriptWriter` outperforms `ScriptWriter-CPre` on this type of data. The difference between these two models is that `ScriptWriter` does not apply the supplementary loss for narrative representation learning. The results reflect that our proposed supplementary loss cannot perform well when there are fewer overlapping words between the narrative and the lines. The reason is that, under this circumstance, it is hard

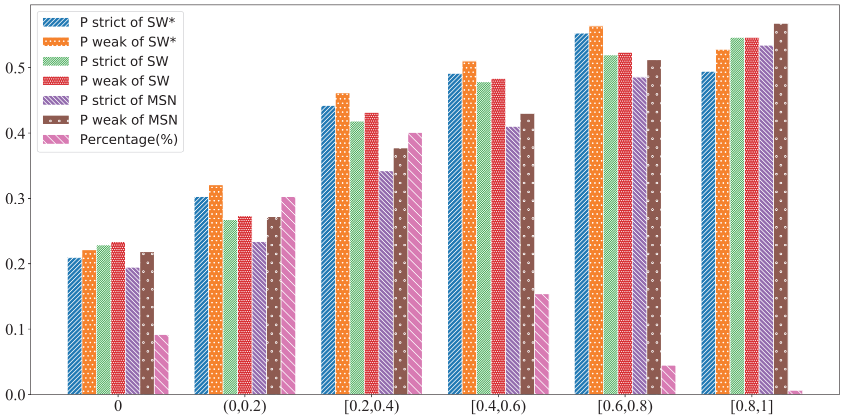


Fig. 25. The performance of ScriptWriter-CPre (SW*), ScriptWriter (SW), and MSN on the test set with different types of narrative in session-level evaluation.

to compute the similarity between the narrative and the line. Therefore, the loss $\mathcal{L}_{\text{kl}}$ is less accurate.

(2) The performance tends to become better when the overlap ratio is increased. This is consistent with our intuition: when a narrative is more detailed and better aligned with the session in wording, it is easier to choose the best responses. This plot also shows that both ScriptWriter-CPre and ScriptWriter can achieve better performance than MSN on all types of narratives, which further demonstrates the effectiveness of using narrative to guide the dialogue.

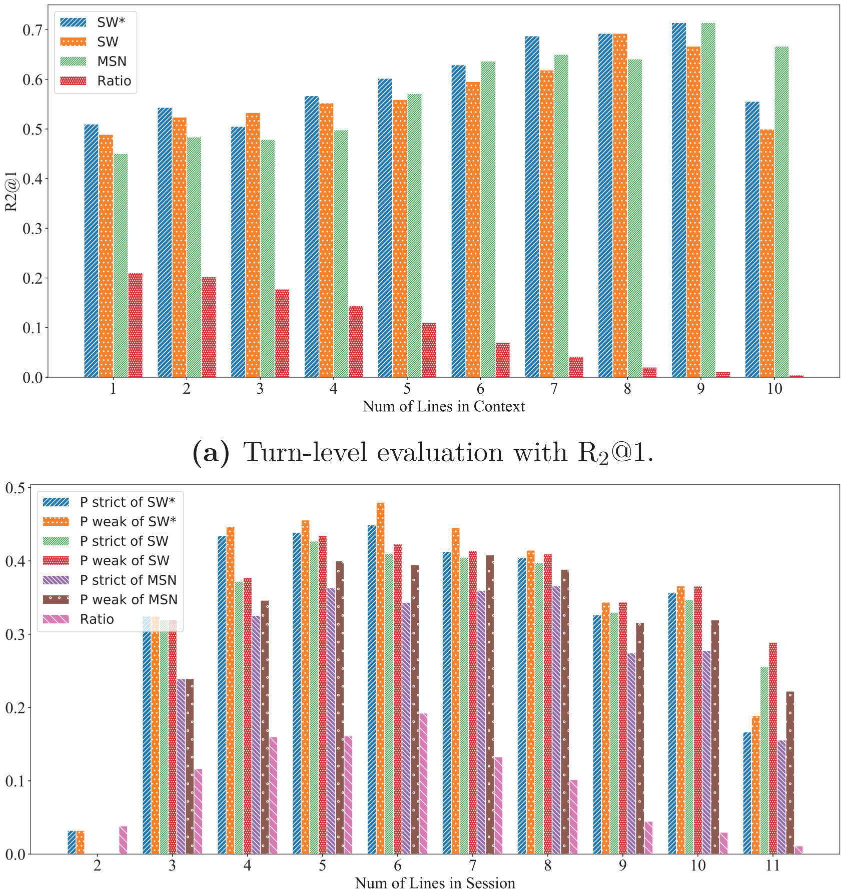
(3) Interestingly, we find that ScriptWriter-CPre performs worse than the other two models in buckets "[0.8, 1]". The potential reason is that there are a lot of words shared between the narrative and the lines, so the model can capture their relationships easily, and the supplementary loss has less effect. However, we cannot draw a solid conclusion on this as there are only 0.6% data lying in this bucket.

(4) We also observe that the buckets "[0, 0.2)" and "[0.2, 0.4)" contain the largest proportions of narratives. This indicates that most Internet users do not use the original lines to retell a story. The problem we address in this paper is thus non-trivial.

6.4. Performance with Various Context Lengths

We study how ScriptWriter-CPre performs in contexts of different lengths and compare it with ScriptWriter and MSN at both turn level and session level.

The results of the turn-level evaluation are shown in Figure 26a. At first, it is clear that the performance of all models increases along with the increase of the number of lines of context from one to eight. This is because more lines in the context can provide more information, which help the model to select better responses. Also, ScriptWriter-CPre performs much better than MSN on very short context. This demonstrates that our method



(a) Turn-level evaluation with $R_2@1$.

(b) Session-level evaluation.

Fig. 26. The performance of `ScriptWriter-CPre` (SW*), `ScriptWriter` (SW), and `MSN` on the test set with different number of lines. We show the performance in both turn-level and session-level evaluation.

can handle the case even when the matching information is insufficient. Finally, when there are more than nine lines in the context, the performance of all models decreases. This could be due to the fact that too many lines of context may more likely contain irrelevant information for the current turn. However, this part of the data only contain less than 2% samples. We will study the problem in the future when more data are available.

The evaluation results at the session level are illustrated in Figure 26b. The x-axis is the total number of lines in the session. Different from the results at turn level, all models perform best with around six to eight lines. The reasons include: (1) When there are only a few lines in the context, the context-response matching, which is essential in our model as shown in the ablation study, cannot perform well, thus limiting the model to selecting a proper response. (2) When a session contains many lines of script, intuitively, the narrative is also long. A long narrative usually contains a lot of points, which should be covered accordingly by the lines. Therefore, these cases are much harder for the model to capture.

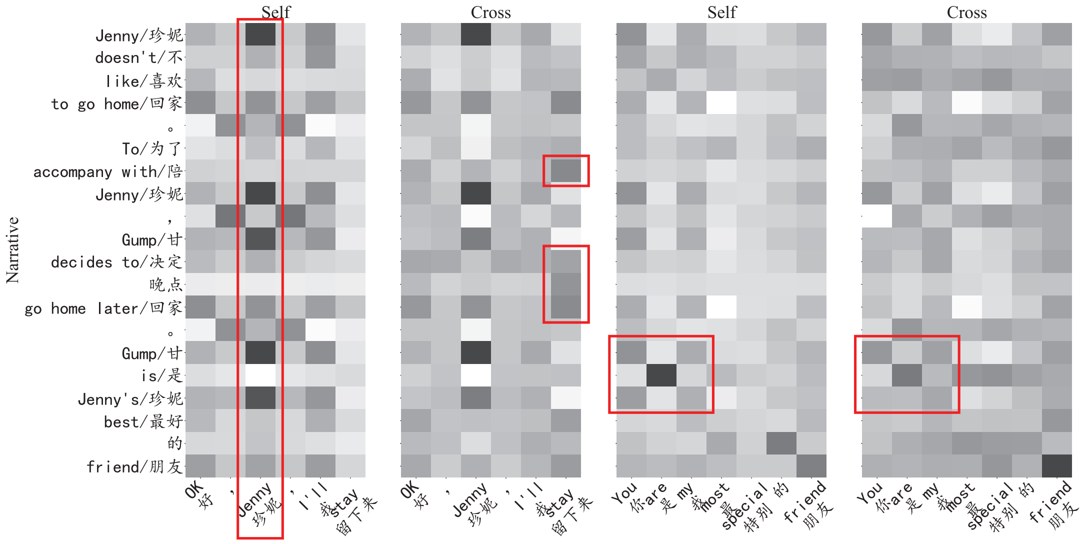


Fig. 27. Matching maps between the response (the second turn in the case study) and the given plan in the first and second level, namely $\mathbf{m}^{\mathrm{self}}_{p,r,0}$ and $\mathbf{m}^{self}_{p,r,1}$.

Nevertheless, our model outperforms MSN in almost all cases, which further demonstrates its effectiveness.

6.5. Visualization

We visualize four attention maps between a narrative and two consecutive responses in both self-attentive blocks and cross-attentive blocks ($\mathbf{m}^{self}_{p,r,0}$ and $\mathbf{m}^{cross}_{p,r,0}$) to see what information the attentive mechanism tries to capture.

From Figure 27, we show the heat map of attention for the example given in the introduction section. It is interesting to see that only some shared words, like "Jenny" and "Jenny", or highly co-occurred words, like "Jenny" and "Gump", can have high similarity in the self-attention matching, while the semantically related words, like "stay" and "go home later" or "accompany with", can have higher similarity in the cross-attention matching. This example provides additional support to the idea of leveraging both kinds of attentive blocks for semantic matching.

Furthermore, we can also see that some phrases such as "Gump is Jenny's" and "you are my" have higher similarity after training. This similarity cannot be captured by shallow semantics, but is based on a deeper understanding of the contextual information. This example shows the usefulness of the attention mechanism for information interactions and their aggregation.

Narrative	Jenny doesn’t like home①. To accompany with Jenny, Gump decides to go home later②. Gump is Jenny’s best friend.③			
Initial line	Mama's going to worry about me.			
Model	Ground-truth	ScriptWriter-CPre (Our model)	ScriptWriter	MSN
1st line	Just stay a little longer.	Just stay a little longer. ①	Just stay a little longer.①	Go file it.
2nd line	Ok, Jenny, I'll stay.	OK, Jenny, I'll stay.②	OK, Jenny, I'll stay.②	OK, Jenny, I'll stay.②
3rd line	You are my most special friend.	Forrest, I do love you.③	Forrest!	You are my most special friend.③
4th line	EOS	The best friend.③	The best friend.③	The best friend.③

Fig. 28. A case study. These are some results sampled from the session-level response selection task. The ground-truth session is extracted from the movie *Forrest Gump.* The highlighted text represents some key points of the narrative.

6.6. Case Study

We show some results generated by `ScriptWriter-CPre`, `ScriptWriter`, and `MSN` in Figure 28. As we highlighted, there are three key points in the narrative. The ground-truth session is perfectly consistent with the narrative. Both `ScriptWriter-CPre` and `ScriptWriter` successfully cover all three key points, whereas `MSN` misses one point.

Looking at the details in the first two lines, `ScriptWriter-CPre` chooses the best response probably because it captures the semantic matching between “go home later” in the narrative and “I’ll stay” in the dialogue, benefiting from multi-grained representations. In the third line, although `ScriptWriter-CPre` selects a nonoptimal response, this response does not conflict with either the context or the narrative. Compared with the 3rd line generated by `ScriptWriter`, the line generated by our method is more related to the given narrative. We attribute this to the design of the narrative representation optimization. Finally, `ScriptWriter-CPre` selects the response “The best friend” in the last line that matches the remaining key point of the narrative. On the contrary, `MSN` selects one (the fourth line) that is redundant with the previous utterance (the third line). This example shows that the method we propose in this paper can better serve our primary goal - covering the key aspects of a narrative.

6.7. Error Analysis

Finally, we conduct an error analysis to investigate the cases that our model cannot handle correctly and summarize some research questions for future work. We randomly sample 50 sessions that are different from the ground-truth and categorize their errors into four groups:

(1) The generated lines are inconsistent with the narrative or incoherent with the context (46%). As shown in the first case of Table 15, the second and last lines are irrelevant to the

Table 15. Examples of typical errors. Four kinds of errors are observed, including (1) being inconsistent with the narrative or incoherent with the context; (2) generating redundant lines; (3) generating information not included in the narrative; and (4) selecting a wrong line for the session ending.

Narrative: Sophie doesn't want to help the Witch, and she ask Howl to go.
Ground-truth Script: (1) So, try it yourself! (2) I'm not so kind to give you a hand (3) Come on Howl! (4) EOS
Generated Script: (1) So, try it yourself! (2) Sophie, let's get closer. (3) Come on Howl! (4) I want to eat Howl's heart.
Error: Inconsistent; No ending

Narrative: Maureen swam back to the bottom of the sea, asking other fish if they saw a boat. But they didn't reply Maureen.
Ground-truth Script: (1) Do you see a boat? (2) A white boat! (3) They took my son! (4) My son! Help me! Please! (5) EOS
Generated Script: (1) Do you see a boat? (2) Do you see it? (3) Do you see it? (4) That has nothing to do with you! (5) Talk to me at least!
Error: Redundant; No ending

Narrative: Father is confessing.
Ground-truth Script: (1) You are reported in the newspaper, son. (2) They say you do a lot of bad things. (3) They describe how the government force you to commit suicide. (4) When we think about this, maybe we are wrong, but it is always your home. (5) EOS
Generated Script: (1) You are reported in the newspaper, son. (2) Have you heard it? (3) EOS
Error: Not covered

narrative. Even though our method considers both narrative-response and context-response matching, the matching features are aggregated together for response selection. A response candidate can be selected because it highly matches either the narrative or the context, rather than both. The aggregation mechanism may not be sufficient to impose good matches for both. We believe that this problem could be solved by setting a gate for both kinds of matching to guarantee the selected response can match both narrative and context.

(2) The generated lines are redundant (26%). As shown in the second case of Table 15, the second and third lines are the same. This problem has also been addressed by other retrieval-based methods [**197**]. The reason is that existing matching-based methods pay much attention to semantic matching but neglect to model text coherence. Some researchers have tried to alleviate this problem by involving pre-trained language models [**55, 197**]. However, this is an unsolved problem and needs further exploration.

(3) The ground-truth lines contain information which is not covered by the narrative (22%), as shown in the third case of Table 15. These cases are very difficult for models to handle since only the context information can be used to select the proper response. Similar to the previous problem, we think modeling text coherence can be a possible way to alleviate this problem.

(4) The model cannot select an ending for a session of scripts (6%), as shown in the first two cases of Table 15. In our experiments, we add a special token "EOS" to mark the "end of session". This token is appended to each session of lines. We find that our model seldom selects this token at the end of a session but tends to select other lines related to the context or narrative. The potential reason is that the model is unable to recognize the ending situation. In other words, the matching model can hardly match a special token with either the context or the narrative. We plan to design an additional mechanism for ScriptWriter-CPre to decide if a session is finished.

7. Conclusion and Future Work

Although story generation has been extensively studied in the literature, no existing work addressed the problem of generating movie scripts following a given storyline or narrative. In this paper, we addressed this problem in the context of generating lines in a movie script. We proposed a model that uses the narrative to guide line generation/retrieval. We keep track of what in the narrative has already been expressed and what is remaining to select the next line through an updating mechanism. The final selection of the next response is based on multiple matching criteria between context, narrative, and response. We constructed a new large-scale data collection for narrative-guided script generation from movie scripts. This is the first public dataset available for testing narrative-guided dialogue generation/selection. Experimental results on the dataset showed that our proposed approach based on narrative significantly outperforms the baselines that use narrative as an additional context. They also showed the importance of using the narrative in the proper manner.

As a first investigation into the problem, our study has several limitations. For example, we have not considered the order in the narrative description, which could be helpful in generating dialogues in the correct order. Other methods to track the dialogue state and the coverage of the narrative can also be designed. We have limited ourselves to retrieval-based script generation. It would be interesting to extend the method to a generation-based approach. Further investigations are thus required to fully understand how narratives can be effectively used in dialogue generation.

Fifth Article.

Proactive Retrieval-based Chatbots based on Relevant Knowledge and Goals

by

Yutao Zhu[1], Jian-Yun Nie[1], Kun Zhou[2], Pan Du[1], Hao Jiang[3], and Zhicheng Dou[4]

([1]) University of Montreal, Quebec, Canada
([2]) School of Information, Renmin University of China, Beijing, China
([3]) Huawei Poisson Lab., Hangzhou, Zhejiang, China
([4]) Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

This article was published in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2000-2004. 2021.

The main contributions of Yutao Zhu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Jian-Yun Nie and Zhicheng Dou contributed to the model design and paper writing; Kun Zhou helped conduct the experiments; Pan Du helped revise the paper; Hao Jiang provided the computation resources for experiments.

Résumé. Un système de dialogue proactif a la capacité de diriger la conversation de manière proactive. Contrairement aux chatbots généraux qui ne font que réagir à l'utilisateur, les systèmes de dialogue proactifs peuvent être utilisés pour atteindre certains objectifs, par exemple recommander certains articles à l'utilisateur. La connaissance du contexte est essentielle pour permettre des transitions fluides et naturelles dans le dialogue. Dans cet article, nous proposons un nouveau cadre d'apprentissage multi-tâches pour le dialogue proactif basé sur la recherche, en s'appuyant sur les connaissances. Pour déterminer les connaissances pertinentes à utiliser, nous considérons la prédiction des connaissances comme une tâche complémentaire et nous utilisons des signaux explicites pour superviser son apprentissage. La réponse finale est sélectionnée en fonction des connaissances prédites, de l'objectif à atteindre et du contexte. Les résultats expérimentaux montrent que la modélisation explicite de la prédiction des connaissances et la sélection de l'objectif peuvent améliorer considérablement la sélection de la réponse finale. Notre code est disponible sur `https://github.com/DaoD/KPN/`.

Mots clés : Dialogue proactif, Chatbot basé sur la recherche, apprentissage multi-tâches

Abstract. A proactive dialogue system has the ability to proactively lead the conversation. Different from the general chatbots which only react to the user, proactive dialogue systems can be used to achieve some goals, *e.g.*, to recommend some items to the user. Background knowledge is essential to enable smooth and natural transitions in dialogue. In this paper, we propose a new multi-task learning framework for retrieval-based knowledge-grounded proactive dialogue. To determine the relevant knowledge to be used, we frame knowledge prediction as a complementary task and use explicit signals to supervise its learning. The final response is selected according to the predicted knowledge, the goal to achieve, and the context. Experimental results show that explicit modeling of knowledge prediction and goal selection can greatly improve the final response selection. Our code is available at `https://github.com/DaoD/KPN/`.

Keywords: Proactive Dialogue, Retrieval-based Chatbot, Multi-task Learning

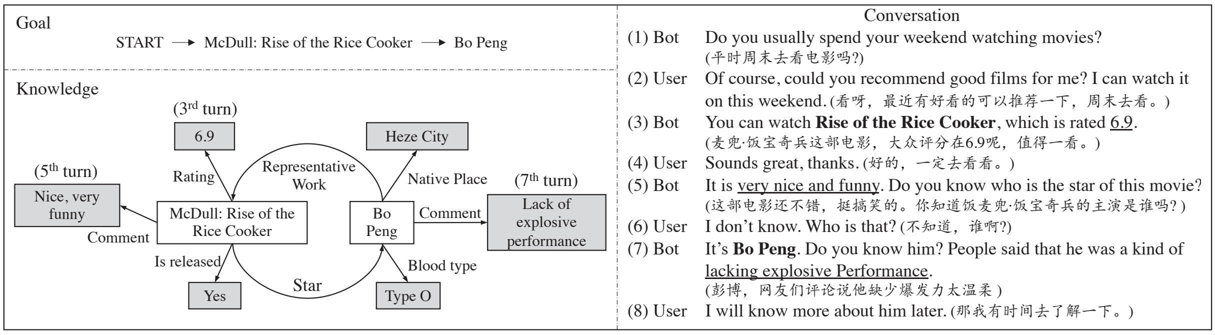
Prologue

In our previous work, we have used a description text to control the whole dialogue process. We now moved a step further: leveraging an additional text to make the dialogue system more proactive. At the time we explored this problem, the majority of dialogue systems worked passively, that is, they were designed to provide a response to user input. They cannot propose a new dialogue topic or lead the dialogue, making the dialogue boring quickly. We noticed some studies trying to design proactive dialogue systems, where they provide some entities with relevant knowledge and ask the dialogue system proactively mention these entities. This setting corresponds to a practical use – conversational recommendation systems, in which the systems are required to recommend the given entities during a dialogue. This problem is close to our previous study, with the difference that the narrative is replaced by an entity list and an additional knowledge graph is provided. After analyzing existing methods for this task, we discovered that they treated the entity list as a piece of knowledge and performed attention to make use of the knowledge. There was no explicit control on how to select relevant knowledge and no tracking of which entity has been covered. Therefore, we proposed a new method that explicitly tracks the entity coverage, and we designed a heuristic to obtain weak labels for knowledge selection. This labeling is based on the observation that a piece of knowledge is involved or not in the gold response. A knowledge selection model is built based on the labels. The experiments results showed that the knowledge selection accuracy can be greatly improved if such more clear feedback (labels) is given. Furthermore, with proper knowledge, the models are better at covering the given entities and completing the dialogue. To our knowledge, this work is one of the few that deal with proactive dialogue.

1. Introduction

From Microsoft Xiaoice, Apple Siri, to Google Assistant, dialogue systems have been widely applied in our daily life. In general, these systems are designed to make responses in reaction to the user's requirements, such as play music, set a clock, or show the weather forecast. These systems are perceived just as tools by users as they only react passively. Users may be bored quickly. The problem is even more severe in a chit-chat style dialogue system. **Proactive conversation** offers a solution to this problem [**176, 204, 224**]. A proactive dialogue system can lead the dialogue proactively to achieve some goals. For example, it can drive the conversation to educate the kids about some topics, to comfort a person, to place ads unobtrusively, or to recommend items to users. Various application scenarios are emerging, yet there has not been a standard definition of what proactive conversation should be. Variations have been observed in the way that a goal is defined: by a sentence [**249**] or by some entities that should be covered in the conversation [**204**], and whether the goal should be generated dynamically [**176**] or predefined [**224**]. We are still in the early stage of exploration in which people test different approaches in different settings. This study is a contribution to this exploration.

In this work, we follow the setting given by Wu et al. [**204**], where the goal is specified by a set of entities (topics) and the background knowledge about these entities is provided in a knowledge graph. The goal is to lead the conversation smoothly to mention the required entities. The knowledge graph helps to generate paths of conversation that appear natural. Despite its simplicity, this setting has many potential applications in practice, in particular in conversational recommendation systems [**84, 103, 147**], where some items can be set in advance for recommendation. An example of proactive conversation in the movie domain is shown in Figure 29. The goal is defined by two entities (topics): the movie *McDull: Rise of the Rice Cooker* and the star *Bo Peng*. The system is asked to cover both entities during



Conversation

(1) Bot Do you usually spend your weekend watching movies? (平时周末去看电影吗?)

(2) User Of course, could you recommend good films for me? I can watch it on this weekend. (看呀，最近有好看的可以推荐一下，周末去看。)

(3) Bot You can watch **Rise of the Rice Cooker**, which is rated 6.9. (麦兜·饭宝奇兵这部电影，大众评分在6.9呢，值得一看。)

(4) User Sounds great, thanks. (好的，一定去看看。)

(5) Bot It is very nice and funny. Do you know who is the star of this movie? (这部电影还不错，挺搞笑的。你知道饭麦兜·饭宝奇兵的主演是谁吗?)

(6) User I don't know. Who is that? (不知道，谁啊?)

(7) Bot It's **Bo Peng**. Do you know him? People said that he was a kind of lacking explosive Performance. (彭博，网友们评论说他缺少爆发力太温柔)

(8) User I will know more about him later. (那我有时间去了解一下。)

Fig. 29. An example of proactive dialogue. The system is asked to exploit the background knowledge to lead the dialogue and accomplish the goal.

the conversation. By exploiting the knowledge graph, the system aims to naturally transit from one conversation topic to another and eventually fulfill the pre-defined goal.

In this work, we focus on the retrieval-based method due to its higher fluency. Although knowledge has been incorporated in some existing approaches to proactive conversation [**204**], it has been simply embedded in the response selection process [**52, 104, 105**], which is optimized globally according to the loss on the final response selection. Although the end-to-end training could be reasonable with a very large amount of training data, in practice, the limited training data may lead to sub-optimal solutions: when a wrong response is selected by the system, it is hard to tell if it is due to a poor knowledge prediction or a bad response selection, thus hard to optimize.

To tackle this problem, we design an explicit *Knowledge Prediction* (KP) module to select the relevant piece of knowledge to use. This module is combined with a *Response Selection* (RS) module, and both form a multi-task learning framework, called **Knowledge Prediction Network** (`KPN`). The two tasks are jointly learned. The KP module first tracks the state of goal achievement, *i.e.*, which part of the goal has been achieved, and then leverages the dialogue context to predict which knowledge should be used in the current turn. The RS module then relies on the selected knowledge to help select the final answer. Different from the existing methods, we explicitly optimize KP using automatically generated weak-supervision signals to help better learn to predict the relevant knowledge. Experimental results show that the explicitly trained KP process can indeed select the most relevant piece of knowledge to use, and this leads to superior performance over the state-of-the-art methods.

Our main contributions are two-fold: (1) We propose a multi-task learning framework for knowledge-grounded proactive dialogue, in which the knowledge prediction task is explicitly trained in a weakly supervised manner. (2) We show experimentally that our model significantly outperforms the existing methods, demonstrating the great importance of knowledge selection in proactive conversation.

2. Knowledge Prediction Network

Problem Formalization We follow the task definition formulated by Wu et al. [**204**]. For a dataset $\mathcal{D}$, each sample is represented as (c,g,k,r,y) (as shown in Figure 29), where $c = \{u_1, \cdots, u_L\}$ represents a conversation context with $\{u_i\}_{i=1}^{L}$ as utterances; g represents the goal containing some entities that the dialogue should talk about (*e.g.*, "Bo Peng"); $k = (k_1, \cdots, k_M)$ are knowledge triplets where k_i is in form of SPO (Subject, Predicate, Object); r is a response candidate; $y \in \{0,1\}$ is a binary label. The task is to learn a matching model $s(c,g,k,r)$ with $\mathcal{D}$ to measure the suitability of a response candidate r.

In this work, we propose a multi-task learning framework `KPN` that contains *response selection* (RS) and *knowledge prediction* (KP) as two distinct tasks, as illustrated in Figure 30. The predicted knowledge and updated goal from the KP task are used as input to the RS task. The loss functions in the two tasks are combined for training the model jointly. Different from the existing work that fuses the two tasks together and trains the whole model by only the final RS loss ($\mathcal{L}_{\text{rs}}$), we propose using a KP loss ($\mathcal{L}_{\text{kp}}$) to supervise the knowledge prediction process directly. The overall loss is as follows:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{kp}} + \mathcal{L}_{\text{rs}}, \tag{2.1}$$

where λ is a hyperparameter (set as 0.3 in our experiment) to control the influence of the KP loss. The joint learning process allows us to better tell if a wrong response is obtained due to a wrong prediction of knowledge or a wrong selection of response. Details of the two tasks are presented in Sections 2.1 and 2.2.

The processes of KP and RS are based on the following basic representations: an utterance u_i in the context, a goal g, a knowledge triplet k_j (concatenated as a word sequence), and a response r are first represented as matrices $\mathbf{e}^{u_i}$, $\mathbf{e}^{g}$, $\mathbf{e}^{k_j}$, and $\mathbf{e}^{r}$ respectively through a pre-trained embedding table. They will be used in different ways in the KP and RS processes.

2.1. Knowledge Prediction (KP) Task

It is widely believed that knowledge can help select suitable responses. However, not all knowledge triplets are useful in selecting responses for a conversation turn. Therefore, predicting whether a knowledge triplet should be used is a critical step.

Goal Tracking To decide what to say in a response, one has to know what part of the goal is still uncovered. `KPN` achieves this by a goal tracking process (shown in the *Tracking* part of Figure 30). The basic idea is to match the goal and the context, then the mismatched entities are considered as uncovered. Concretely, we concatenate all utterances in the context as a long sequence $\{\mathbf{e}_i^u\}_{i=1}^{N}$, where N is the total number of words in all the utterances, and then match it with the goal ($\mathbf{e}^g$) by cosine similarity: $\mathbf{m}_{ij} = \cos(\mathbf{e}_i^g, \mathbf{e}_j^u)$. Then max-pooling is applied to extract the strongest matching signals: $v_i = \text{ReLU}(\text{Maxpooling}(\mathbf{m}_{i,:}))$.

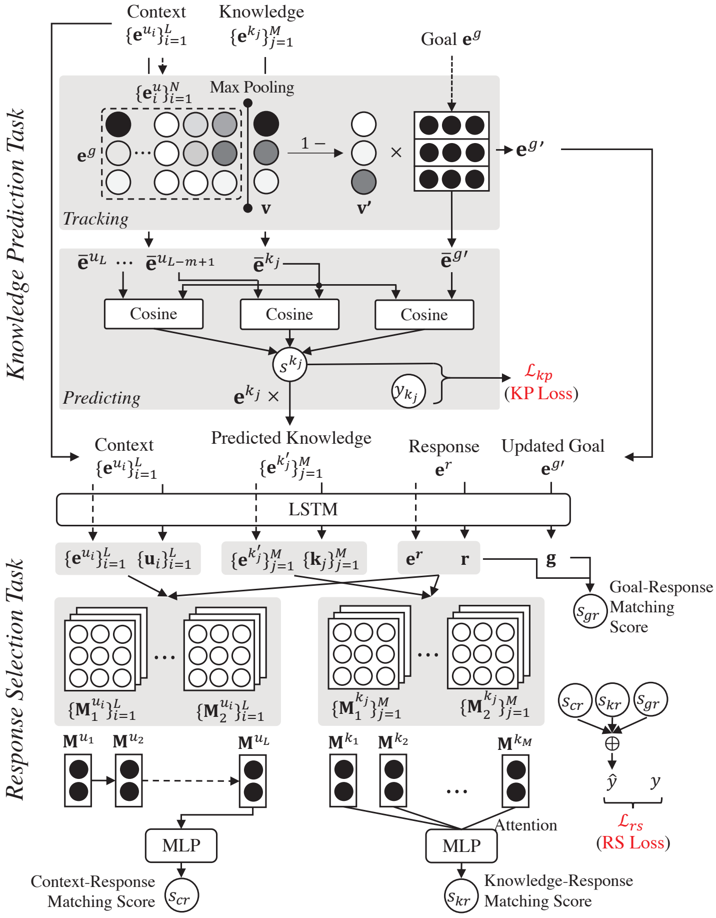

Knowledge Prediction Task
Context $\{\mathbf{e}^{u_i}\}_{i=1}^{L}$
Knowledge $\{\mathbf{e}^{k_j}\}_{j=1}^{M}$
Goal $\mathbf{e}^{g}$
$\{\mathbf{e}_i^{u}\}_{i=1}^{N}$
Max Pooling
$\mathbf{e}^{g}$
$1-$
×
$\mathbf{e}^{g\prime}$
$\mathbf{v}$
$\mathbf{v}'$
Tracking
$\bar{\mathbf{e}}^{u_L}$ … $\bar{\mathbf{e}}^{u_{L-m+1}}$
$\bar{\mathbf{e}}^{k_j}$
$\bar{\mathbf{e}}^{g\prime}$
Cosine
Cosine
Cosine
s^{k_j}
y_{k_j}
$\mathcal{L}_{kp}$
(KP Loss)
Predicting
$\mathbf{e}^{k_j}$ ×
Response Selection Task
Context $\{\mathbf{e}^{u_i}\}_{i=1}^{L}$
Predicted Knowledge $\{\mathbf{e}^{k_j'}\}_{j=1}^{M}$
Response $\mathbf{e}^{r}$
Updated Goal $\mathbf{e}^{g\prime}$
LSTM
$\{\mathbf{e}^{u_i}\}_{i=1}^{L}$ $\{\mathbf{u}_i\}_{i=1}^{L}$
$\{\mathbf{e}^{k_j'}\}_{j=1}^{M}$ $\{\mathbf{k}_j\}_{j=1}^{M}$
$\mathbf{e}^{r}$ $\mathbf{r}$
$\mathbf{g}$
s_{gr} Goal-Response Matching Score
$\{\mathbf{M}_1^{u_i}\}_{i=1}^{L}$
$\{\mathbf{M}_2^{u_i}\}_{i=1}^{L}$
$\{\mathbf{M}_1^{k_j}\}_{j=1}^{M}$
$\{\mathbf{M}_2^{k_j}\}_{j=1}^{M}$
s_{cr} s_{kr} s_{gr}
$\oplus$
$\hat{y}$ y
$\mathcal{L}_{rs}$
(RS Loss)
$\mathbf{M}^{u_1}$ $\mathbf{M}^{u_2}$ $\mathbf{M}^{u_L}$
$\mathbf{M}^{k_1}$ $\mathbf{M}^{k_2}$ $\mathbf{M}^{k_M}$
Attention
MLP
MLP
Context-Response Matching Score s_{cr}
s_{kr} Knowledge-Response Matching Score

Fig. 30. The structure of KPN. The predicted knowledge $\mathbf{e}^{k_i'}$ and updated goal $\mathbf{e}^{g'}$ in the knowledge prediction task will be used as input to the response selection task.

The obtained values ($\mathbf{v}$) represent the degree of coverage of the entities in the goal, while $\mathbf{v}' = \mathbf{1} - \mathbf{v}$ represents the remaining part that should be covered in the following dialogue. Finally, the vector $\mathbf{v}'$ is used to update the representation of the goal: $\mathbf{e}^{g'} = \mathbf{v}' \cdot \mathbf{e}^{g}$. This goal tracking method is simple but effective, and more sophisticated designs can be investigated as future work.

Knowledge Predicting The knowledge prediction process is shown in the *Predicting* part of Figure 30. The relevance of a piece of knowledge is determined by both the state of the goal and the current topic of the dialogue. The former determines the target, while the latter determines the starting point. Ideally, the relevant knowledge should pave a way leading from the current topic to the desired goal. Usually, the current topic is contained in the last several utterances, thus we leverage them to predict the relevant knowledge. Given the updated goal $\mathbf{e}^{g'}$, the last m utterances $\{\mathbf{e}^{u_i}\}_{i=L-m+1}^{L}$ (where L is the number of utterances in the context, and m is a hyperparameter set as 3 in our experiments), and the j-th piece of knowledge $\mathbf{e}^{k_j}$, we first compute their sentence-level representations by mean-pooling over word dimensions:

$\bar{\mathbf{e}}^{u_i} = \text{mean}(\mathbf{e}^{u_i})$, $\bar{\mathbf{e}}^{g'} = \text{mean}(\mathbf{e}^{g'})$, and $\bar{\mathbf{e}}^{k_j} = \text{mean}(\mathbf{e}^{k_j})$. Then we use cosine similarity $s_{g'}^{k_j} = \cos(\bar{\mathbf{e}}^{g'}, \bar{\mathbf{e}}^{k_j})$, $s_i^{k_j} = \cos(\bar{\mathbf{e}}^{u_i}, \bar{\mathbf{e}}^{k_j})$ to measure their relevance, where $i \in [L-m+1, L]$, and we obtain $(m+1)$ scores $[s_{g'}^{k_j}, s_{L-m+1}^{k_j}, \cdots, s_L^{k_j}]$. The relevance scores with respect to both the goal and the context topic are aggregated by a multi-layer perceptron (MLP) with a sigmoid activation function (σ), which is then used to update the representation of the j-th knowledge triplet:

$$s^{k_j} = \sigma\Big(\text{MLP}([s_0^{k_j}; s_{L-m+1}^{k_j}; \cdots; s_L^{k_j}])\Big), \quad \mathbf{e}^{k_j'} = s^{k_j}\mathbf{e}^{k_j}, \tag{2.2}$$

where s^{k_j} is the predicted probability of the j-th knowledge triplet to be used in the current turn.

Weakly Supervised Knowledge Prediction To make a correct prediction of knowledge, the common method is tuning the knowledge prediction process according to the final response selection error. The process is thus implicitly supervised [**104, 105, 204**]. To further improve the learning of the knowledge prediction, besides the response selection loss, we introduce a weakly supervised knowledge prediction loss to train it explicitly.

In practice, it is difficult to have manual labels for knowledge triplets in each dialogue turn. To address this problem, we propose a method to generate weak labels automatically. For each knowledge SPO triplet, we adopt an entity linking method to link it to the response: if the *object entity* appears in the ground-truth response, we label it as 1, otherwise as 0.[24] We assume this weak label can indicate whether such a piece of knowledge is used in the ground-truth response. With the weak labels y_{k_j}, we can compute a binary cross-entropy loss, which we call KP loss, as follows:

$$\mathcal{L}_{\text{kp}} = -\frac{1}{|\mathcal{D}|}\sum\Big(y_{k_j}\log s^{k_j} + (1-y_{k_j})\log(1-s^{k_j})\Big). \tag{2.3}$$

2.2. Response Selection (RS) Task

Response selection (RS) is the main task. As shown in Figure 30, `KPN` considers the interactions between response and three types of information, *i.e.*, the context, the knowledge, and the remaining goal. The former two can be modeled in the same way: similar to existing work [**73, 223, 247**], we compute matching matrices based on both the input representations ($\mathbf{e}^{u_i}$, $\mathbf{e}^{k_j'}$ and $\mathbf{e}^r$) and their sequential representations obtained by LSTM [**71**]. As a result, we denote the obtained matrices as $\mathbf{M}^u$ and $\mathbf{M}^k$ and apply a CNN with max-pooling to extract the matching features $\mathbf{v}^u$ and $\mathbf{v}^k$.

(1) **Context-Response Matching** The matching features between the context and response are aggregated by an LSTM and the corresponding final state is fed into an MLP

[24] For long descriptive entities (*i.e.*, non-factoid sentences such as the *Comment* entity about *Bo Peng* in the Knowledge Graph in Figure 29), if more than 70% part is covered by the ground-truth response, we label it as one. We do not use the subject entity (*e.g.*, "Bo Peng"), because it is shared by many triplets, thus is less accurate as the label.

to compute the matching score s_{cr}. We use LSTM because it can model the dependency and the temporal relationship of utterances in the context.

(2) **Knowledge-Response Matching** Different from the context, we assume knowledge triplets to be independent. Thus, we use an attention-based method to aggregate the matching features:

$$\alpha_i = \text{ReLU}\Big(\text{MLP}(\mathbf{v}_i^k)\Big), \;\; \mathbf{h}_2 = \sum_{i=1}^{k_M} \frac{e^{\alpha_i}}{\sum_j e^{\alpha_j}} \mathbf{v}_i^k, \;\; s_{kr} = \text{MLP}(\mathbf{h}_2). \tag{2.4}$$

This way, a knowledge triplet that is more related to the response will have a higher weight in the aggregated features and contributes more in computing the final matching score.

(3) **Goal-Response Matching** As the goal is a single sequence of tokens, which is much easier to model, we compute the goal-response matching score s_{gr} by an MLP based on their LSTM representations at the last time step.

The final matching score is then computed as: $\hat{y} = \Big(s_{cr} + s_{kr} + s_{gr}\Big)/3$. We use the binary cross-entropy loss to compute the errors:

$$\mathcal{L}_{\text{rs}} = -\frac{1}{|\mathcal{D}|} \sum \left(y \log \sigma(\hat{y}) - (1-y) \log(1-\sigma(\hat{y}))\right). \tag{2.5}$$

3. Experiments

3.1. Datasets and Baseline Models

We experiment on datasets DuConv and DuRecDial. **DuConv** [**204**] is built for knowledge-grounded proactive human-machine conversation. The dialogues are about movies and stars. The total number of training, validation, and test samples is 898,970, 90,540, and 50,000. **DuRecDial** [**109**] is created as a conversational recommendation dataset, which contains dialogues between a seeker and a recommender. The domain of dialogue includes movie, music, food, etc. The number of training, validation, and test samples is 342,340, 38,060, and 55,270. The negative responses are randomly sampled with a 1:9 positive/negative ratio in both datasets.

We compare our model against two groups of baseline methods:

`DuRetrieval` [**204**] is the only retrieval-based model specifically designed for proactive dialogue. It uses a Transformer-based encoder for context and response representation. The conversation goal is used as an additional piece of knowledge. All knowledge triplets are represented by a bi-GRU with attention mechanism.

The other group of methods are not proposed for proactive dialogue but for general knowledge-grounded dialogue. As they also incorporate knowledge into dialogue generation, we replace our knowledge selecting module in the KP task by theirs to make a comparison. `MemNet` [**52**] uses a memory network that performs "read" and "write" on the knowledge by matrix multiplication. PostKS [**104**] trains a knowledge prediction process to make the prior probability (using only the context) of the knowledge prediction close to the posterior probability (using both context and response). NKD [**105**] is similar to MemNet, but it first concatenates the context and knowledge representations and then uses an MLP to compute the weight for each piece of knowledge.

3.2. Evaluation

All models are evaluated in two scenarios.

On test set Similar to the existing work [**204, 226**], we evaluate the performance of each model by **Hits@1**, **Hits@3**, and Mean Reciprocal Rank (**MRR**) for selecting the correct response when it is mixed up with several other candidates. Hits@k measures the ratio of the ground-truth response among the top k results.

Practical application Following [**204**], we also evaluate the performance of the models in a more practical scenario, where each ground-truth utterance is mixed up with 49 utterances retrieved by Solr.[25]. The task is to rank the ground-truth response as high as possible. This test simulates a practical scenario where the model is acting as a reranker for the candidate list returned by an upstream retrieval system. We use several metrics to evaluate the model from different perspectives. **BLEUs** are used to measure the quality (similarity) of the response w.r.t. the ground-truth. To evaluate the model's ability to incorporate knowledge into dialogues, we compute the **knowledge precision/recall/F1** score used in previous studies [**104, 148, 204**], which measure how much knowledge (either correct or wrong) has been used in the responses. We also compute a more meaningful **knowledge accuracy** to measure if the selected response uses the same piece of knowledge as that involved in the ground-truth response. Similarly, **goal accuracy** measures if a goal in the ground-truth is correctly covered by the selected response.

3.3. Experimental Results

The evaluation results are shown in Table 16. Based on the results, we can observe: (1) KPN outperforms all baselines significantly by achieving the highest scores on all evaluation metrics. (2) Compared with DuRetrieval, KPN improves Hits@1, Hits@3, and MRR by a large margin. This strongly indicates that KPN has a better capability of selecting correct responses. (3) In the practical application scenario, according to the results on BLEU, we can conclude that KPN can select responses that are more similar to the golden responses. (4) On knowledge prediction, as a comparison, we also provide the evaluation result of the ground-truth. We find that our method outperforms other knowledge prediction models

[25] https://lucene.apache.org/solr/ If the number of retrieved results is less than 49, we use random samples to pad.

Table 16. Evaluation results. KLG. stands for knowledge and Acc. stands for accuracy. "`+X`" means that knowledge prediction is replaced by `X`. The improvement obtained by `KPN` over `DuRetrival` is statistically significant with p-value < 0.01 in t-test.

	`Ground-truth`	`DuRetrieval`	`KPN`	`+MemNet`	`+PostKS`	`+NKD`
DuConv						
Hits@1	-	50.12	**66.94**	52.54	39.98	56.42
Hits@3	-	75.68	**87.52**	78.70	65.70	57.09
MRR	-	63.13	**78.30**	67.90	81.54	70.77
BLEU1	1.00	0.47	**0.56**	0.50	0.48	0.50
BLEU2	1.00	0.32	**0.42**	0.34	0.33	0.35
KLG. P	38.24	30.11	**33.45**	29.24	28.55	29.40
KLG. R	9.20	7.24	**8.05**	7.03	6.87	7.07
KLG. F1	14.83	11.68	**12.97**	11.34	11.07	11.40
KLG. Acc.	100.00	53.64	**57.82**	50.90	50.42	52.94
Goal Acc.	100.00	58.90	**77.58**	72.36	69.44	74.62
DuRecDial						
Hits@1	-	77.38	**91.50**	75.34	82.45	82.74
Hits@3	-	89.02	**98.86**	93.92	96.60	97.03
MRR	-	84.47	**95.18**	85.00	89.58	89.96
BLEU1	1.00	0.46	**0.61**	0.51	0.53	0.53
BLEU2	1.00	0.39	**0.51**	0.39	0.41	0.41
KLG. P	52.64	43.42	**52.55**	41.04	43.70	42.87
KLG. R	3.76	5.79	**7.01**	5.48	5.83	5.72
KLG. F1	7.02	5.68	**12.97**	11.34	11.07	11.40
KLG. Acc.	100.00	94.90	**95.35**	94.32	94.90	94.81
Goal Acc.	100.00	78.34	**84.96**	82.58	83.12	83.93

(MemNet, PostKS, and NKD) on knowledge P/R/F1 and accuracy. This demonstrates that the explicit supervised knowledge prediction is more effective than the implicit ones used in the other methods. Nevertheless, there is still a big gap between our results and the ground-truth, showing that the process could be much improved.

Reliability of the Weak Labels As we use an entity linking method to automatically generate weak labels for knowledge prediction, to evaluate the reliability of these labels, we randomly select 100 samples comprising 1,437 knowledge triplets from the validation set of DuConv, and ask three human annotators to label which triplet is necessary to select the current response. The result indicates that 90.26% of the generated labels are consistent with human annotations.[26] This demonstrates the high reliability of the labels automatically generated by our entity linking method.

[26] The Fleiss Kappa is 0.698 that indicates the annotators achieve a substantial agreement.

We carried out detailed **Ablation Study** and **Influence of Hyperparameter**, showing that both the goal and knowledge strongly impact the final results. Due to space limit, these experiments are presented in our Github page.

4. Conclusion

In this paper, we proposed a new approach to retrieval-based proactive dialogue. In our model, we define two tasks for response selection and knowledge prediction. An interactive matching structure is applied to model the matching between the knowledge and the response. In order to make a good prediction of knowledge, explicit supervision signals are used, which are derived from the ground-truth responses. Experimental results demonstrated that our model can achieve better performance than the baseline models in which the two tasks are mixed up. In particular, it is shown that training the knowledge prediction explicitly is very effective. This work is a first demonstration of the importance of modeling knowledge and goals explicitly in proactive dialogue.

Chapter 3

Conclusion

In this thesis, we have presented several methods for improving context-aware ranking in two special scenarios: search and dialogue. Both of them have wide applications in practice. In the search scenario, we investigated the problem of context-aware document ranking. We proposed a contrastive learning and a curriculum learning framework, which respectively improved the robustness of the model and enhanced the model optimization process. Both of them achieved significant improvements over previous approaches. As for the dialogue scenario, we used different kinds of knowledge as additional context to facilitate the dialogue response ranking. We investigated grounding a dialogue on documents with hard content selection, controlling a movie dialogue by a pre-defined narrative, and proactively leading a dialogue according to a given goal and background knowledge. All of them boosted the dialogue capability in different perspectives.

In the first article, we discussed how to address the issue that current models cannot cope with the variations in user behavior sequences. We discovered that this is because all user behavior sequences are viewed as definite and exact sequences. We proposed three data augmentation strategies to generate possible variations for user behavior sequences. By masking some terms in a query or document, deleting some queries or documents from the sequence, or reordering the sequence, we can simulate some typical variations in user's behavior sequences. The generated behavior sequences can be considered similar to the observed ones. Then, we applied contrastive learning to identify what is similar and dissimilar. Compared to existing approaches, we showed that contrastive learning can better cope with the inherent variations of user behavior sequences and generate more robust models to deal with new sequences. As a first attempt to capture the variations in user behavior sequences, our proposed data augmentation strategies are simple and fully unsupervised. The augmented sequences may be unreasonable in practice. For example, if we randomly mask some terms of a query, its search intent may be totally changed. A potential way to tackle this problem is to leverage the search log. We could obtain a large number of query-clicked document pairs and use the documents under the same query to generate augmented sequences. These sequences have similar semantics and can better simulate the real variations in user behaviors.

In the second article, we discussed our findings that the relevance/irrelevance in different positive/negative search context-document pairs has different difficulties to identify. In the traditional learning paradigm, training samples are randomly (uniformly) selected to form a training batch and optimize the model. This may confuse the model with the difficulty of the task. We proposed a curriculum learning framework to improve model optimization. The training samples are organized according to their difficulty, then the model can learn them from easy to hard. Our proposed framework can be used to train any existing model. In our experiments, we showed that it greatly improved several strong baselines, demonstrating the effectiveness of curriculum learning for context-aware document ranking. Essentially, curriculum learning adjusts the order of the training samples so that the entire learning can be organized in an easy-to-hard manner. It does not modify the training samples. A further research question is how to obtain the most effective positive/negative samples for the training. In our experiments, we have found that adding more hard negative documents is beneficial for model optimization. These documents may provide more contrastive signals for learning matching signals. Therefore, in the future, we plan to investigate the mining of effective training samples for context-aware ranking.

In the third article, we focused on the problem of document-grounded dialogue response ranking. We reported that current methods often leveraged the whole document to select the response, which may lead to redundant or scattered results. To tackle this problem, we built a model with hard content selection. It can filter out irrelevant document content before performing matching with the response candidate. This significantly reduced the noise of the document. Besides, when selecting the document content, our model paid more attention to the recent dialogue topic rather than the whole dialogue. It further enhanced the accuracy of document content selection. Experimental results on two datasets validate the effectiveness of our proposed model. It demonstrates the necessity of content selection for document-grounded dialogue response ranking. A limitation of our method is that it cannot be combined with pre-trained models. In general, pre-trained language models adopt attention to selecting useful information from the document, so they also have the problem that irrelevant content may disturb the matching process. As a result, a possible future direction is how to equip pre-trained language models with our proposed hard selection mechanism.

In the fourth article, we paid attention to the dialogue response ranking task and discussed how to control a dialogue by following a pre-defined plan. Since there was no existing dataset, we constructed a new one in the movie domain, and defined a new task as narrative-guided movie script generation. The narrative plays the role of dialogue guidance, while the script session is the corresponding dialogue. To make the script session follow the given narrative, we designed a new model structure that can keep track of what has been covered in the narrative and predict what should be covered in the next script line. By iteratively selecting the next script line, our method can generate the whole script session. Both automatic and human evaluation validated that our proposed approach is capable of producing script sessions that are consistent with the given narratives. This study also showed that a narrative plays a different role from a general dialogue context. It is worth noting that, in this study, we assumed that the whole script is controlled by the narrative, which is different from human-machine dialogues, where only the chatbot side can be controlled. Therefore, a valuable research question is how to control the chatbot with a given narrative. Though we can apply our method to this question by only tracking the information contained on the chatbot side, we believe a more tailored tracking mechanism is necessary in this case.

In the fifth article, we explored a research question about how to make the dialogue model work more proactively. We followed a recent study where a dialogue model was provided with a knowledge graph and asked to mention several given entities during the dialogue. The model should work proactively to talk about some entities and make proper transitions between them. We found that tuning the whole model by the final response selection loss was insufficient to tell which part of the model is suboptimal. Therefore, we first devised an automatic method to annotate which piece of knowledge in the knowledge graph is used at each dialogue turn. Then, we computed a supplementary loss to supervise the knowledge selection process. It showed that with more explicit signals, the model can select more proper knowledge for the dialogue. Besides, in the knowledge selection process, we also considered which entities had already been mentioned in the dialogue in order to let the model concentrate on the knowledge about uncovered ones. The experiments indicated that our method is better at goal completion rate, knowledge selection accuracy, and dialogue quality. Nevertheless, we think the current setting is still too ideal, *i.e.*, the goal with entities is pre-defined. In most human dialogues, it is impossible to determine the dialogue topics (*e.g.*, the entities that should be mentioned) in advance. Therefore, a mechanism that can dynamically generate these entities along the dialogue process is urgently required.

Overall, this thesis contains several studies trying to make good use of search and dialogue contexts. While our approaches demonstrated improved effectiveness, they are still far from the natural behaviors of human beings. We observe two typical limitations in existing methods as follows: (1) The long-term dependency in the context cannot be well-captured. For humans, it is easy to recall communication that took place a long time ago. However, this is extremely challenging for neural models. Though some recent large language models can accept an input of thousands of tokens (*e.g.*, 2,048 tokens for GPT-3 and 8,192 tokens for ChatGPT), it is impractical to increase the input length indefinitely. (2) The context semantic is hard to model when there are a lot of omissions. It is natural for humans to understand a query or sentence with some omissions, to which the world's knowledge contributes significantly. A possible solution to this problem is increasing the model's capacity (*i.e.*, size), as recent large language models have done. Unfortunately, similar to the previous problem, it is also impractical to increase the model's size indefinitely. Recently, we have witnessed the great power of ChatGPT. It can accomplish a lot of human tasks with very high accuracy. Our study may be integrated with ChatGPT in several ways. For example, through the context-aware document ranking models, more relevant documents can be retrieved to enhance the factualness of ChatGPT's generated content. ChatGPT only uses implicit knowledge extracted from texts, but does not use explicit external knowledge. It is possible to enhance ChatGPT with the existing knowledge graphs, and for this, our proposed methods can be useful. Finally, making ChatGPT work more proactively is also a promising direction. In summary, despite the rapid evolution in this area, our research work is still relevant for future research in this field.

Publication

(1) Contrastive Learning for Legal Judgment Prediction, Han Zhang, Zhicheng Dou, **Yutao Zhu**, and Ji-Rong Wen. *ACM Transactions on Information Systems, 2023 (TOIS).*

(2) Learning from the Wisdom of Crowds: Exploiting Similar Sessions for Session Search. Yuhang Ye, Zhonghua Li, Zhicheng Dou, **Yutao Zhu**, Changwang Zhang, Shangquan Wu, and Zhao Cao. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI 2023).*

(3) Heterogeneous Graph-based Context-aware Document Ranking. Shuting Wang, Zhicheng Dou, and **Yutao Zhu**. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM 2023).*

(4) MCP: Self-supervised Pre-training for Personalized Chatbots with Multi-level Contrastive Sampling. Zhaoheng Huang, Zhicheng Dou, **Yutao Zhu**, and Zhengyi Ma. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022 Findings).*

(5) Coarse-to-Fine: Hierarchical Multi-task Learning for Natural Language Understanding. Zhaoye Fei, Yu Tian, Yongkang Wu, Xinyu Zhang, **Yutao Zhu**, Zheng Liu, Jiawen Wu, Dejiang Kong, Ruofei Lai, Zhao Cao, Zhicheng Dou, and Xipeng Qiu. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING 2022).*

(6) From Easy to Hard: A Dual Curriculum Learning Framework for Context-Aware Document Ranking. **Yutao Zhu**, Jian-Yun Nie, Yixuan Su, Haonan Chen, Xinyu Zhang, and Zhicheng Dou. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM 2022).*

(7) Enhancing User Behavior Sequence Modeling by Generative Tasks for Session Search. Haonan Chen, Zhicheng Dou, **Yutao Zhu**, Zhao Cao, Xiaohua Cheng, and Ji-Rong Wen. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM 2022).*

(8) GDESA: Greedy Diversity Encoder with Self-Attention for Search Results Diversification. Xubo Qin, Zhicheng Dou, **Yutao Zhu**, and Ji-Rong Wen. *ACM Transactions on Information Systems, 2022 (TOIS).*

(9) Knowledge Enhanced Search Result Diversification, Zhan Su, Zhicheng Dou. **Yutao Zhu**, and Ji-Rong Wen. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022).*

(10) Less is More: Learning to Refine Dialogue History for Personalized Dialogue Generation. Hanxun Zhong, Zhicheng Dou, **Yutao Zhu**, Hongjin Qian, and Ji-Rong Wen. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT 2022).*

(11) Leveraging Narrative to Generate Movie Script. **Yutao Zhu**, Ruihua Song, Jian-Yun Nie, Pan Du, Zhicheng Dou, and Jin Zhou. *ACM Transactions on Information Systems, 2022 (TOIS).*

(12) Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking. **Yutao Zhu**, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM 2021).*

(13) PSSL: Self-supervised Learning for Personalized Search with Contrastive Sampling. Yujia Zhou, Zhicheng Dou, **Yutao Zhu**, and Ji-Rong Wen. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM 2021).*

(14) Learning Implicit User Profile for Personalized Retrieval-Based Chatbot. Hongjin Qian, Zhicheng Dou, **Yutao Zhu**, Yueyuan Ma, and Ji-Rong Wen. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM 2021).*

(15) Interaction-Based Document Matching for Implicit Search Result Diversification. Xubo Qin, Zhicheng Dou, **Yutao Zhu**, Ji-Rong Wen. In *Proceedings of the 27th China National Conference on Information Retrieval (CCIR 2021).*

(16) Graph Neural Collaborative Topic Model for Citation Recommendation. Qianqian Xie, **Yutao Zhu**, Jimin Huang, Pan Du, and Jian-Yun Nie. *ACM Transactions on Information Systems, 2021 (TOIS).*

(17) Few-Shot Charge Prediction with Multi-grained Features and Mutual Information. Han Zhang, Zhicheng Dou, **Yutao Zhu**, and Ji-Rong Wen. In *Proceedings of the 20th China National Conference on Computational Linguistics (CCL 2021).*

(18) Proactive Retrieval-based Chatbots based on Relevant Knowledge and Goals. **Yutao Zhu**, Jian-Yun Nie, Kun Zhou, Pan Du, Hao Jiang, and Zhicheng Dou. In *Proceedings*

of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021).

(19) Modeling Intent Graph for Search Result Diversification. Zhan Su, Zhicheng Dou, **Yutao Zhu**, Xubo Qin, and Ji-Rong Wen. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021).*

(20) One Chatbot Per Person: Creating Personalized Chatbots based on Implicit User Profiles. Zhengyi Ma, Zhicheng Dou, **Yutao Zhu**, Hanxun Zhong, and Ji-Rong Wen. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021).*

(21) Pchatbot: A Large-Scale Dataset for Personalized Chatbot. Hongjin Qian, Xiaohe Li, Hanxun Zhong, Yu Guo, Yueyuan Ma, **Yutao Zhu**, Zhanliang Liu, Zhicheng Dou, and Ji-Rong Wen. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021).*

(22) Content Selection Network for Document-grounded Retrieval-based Chatbots. **Yutao Zhu**, Jian-Yun Nie, Kun Zhou, Pan Du, and Zhicheng Dou. In *Proceedings of the 43rd edition of the annual BCS-IRSG European Conference on Information Retrieval (ECIR 2021).*

(23) Neural Sentence Ordering Based on Constraint Graphs. **Yutao Zhu**, Kun Zhou, Jian-Yun Nie, Shengchao Liu, and Zhicheng Dou. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021).*

(24) S^3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. Kun Zhou, Hui Wang, Wayne Xin Zhao, **Yutao Zhu**, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM 2020).*

(25) ScriptWriter: Narrative-Guided Script Generation. **Yutao Zhu**, Ruihua Song, Zhicheng Dou, Jian-Yun Nie, and Jin Zhou. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics (ACL 2020).*

(26) Improving Multi-Turn Response Selection Models with Complementary Last-Utterance Selection by Instance Weighting. Kun Zhou, Wayne Xin Zhao, **Yutao Zhu**, Ji-Rong Wen, and Jingsong Yu. In *Proceedings of the 26th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2020).*

(27) ReBoost: A Retrieval-Boosted Sequence-to-SequenceModel for Neural Response Generation. **Yutao Zhu**, Zhicheng Dou, Jian-Yun Nie, and Ji-Rong Wen. *Information Retrieval Journal, 2019 (IRJ).*

(28) Deep Cross-platform Product Matching in E-commerce. Juan Li, Zhicheng Dou, **Yutao Zhu**, Xiaochen Zuo, and Ji-Rong Wen. *Information Retrieval Journal, 2019 (IRJ).*

(29) A Hybrid Framework of Emotion-Aware Seq2Seq Model for Emotional Conversation Generation. Xiaohe Li, Jiaqing Liu, Weihao Zheng, Xiangbo Wang, **Yutao Zhu**, and Zhicheng Dou. In *Proceedings of the 14th NTCIR Evaluation of Information Access Technologies (NTCIR 2019).*

(30) An Attribute-aware Neural Attentive Model for Next Basket Recommendation. Ting Bai, Jian-Yun Nie, Wayne Xin Zhao, **Yutao Zhu**, Pan Du, and Ji-Rong Wen. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018).*

References

[1] *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events, NUSE@ACL 2020, Online, July 9, 2020.* Association for Computational Linguistics, 2020.

[2] Eugene Agichtein, Ryen W. White, Susan T. Dumais, and Paul N. Bennett. Search, interrupted: understanding and predicting search task continuation. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 315–324. ACM, 2012.

[3] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Multi-task learning for document ranking and query suggestion. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[4] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 385–394. ACM, 2019.

[5] Siddhartha Arora, Mitesh M. Khapra, and Harish G. Ramaswamy. On knowledge distillation from complex networks for response prediction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3813–3822. Association for Computational Linguistics, 2019.

[6] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[7] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM, 2009.

[8] Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994.

[9] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the impact of short- and long-term behavior on search personalization. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 185–194. ACM, 2012.

[10] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on*

Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016, pages 10–21. ACL, 2016.

[11] Selmer Bringsjord and David Ferrucci. *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, A Storytelling Machine*. Psychology Press, 1999.

[12] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[13] Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. Retrieval-guided dialogue response generation via a matching-to-generation framework. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1866–1875. Association for Computational Linguistics, 2019.

[14] Huanhuan Cao, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 191–200. ACM, 2009.

[15] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 129–136. ACM, 2007.

[16] David Carmel and Elad Yom-Tov. Estimating the query difficulty for information retrieval. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, page 911. ACM, 2010.

[17] Ben Carterette, Paul D. Clough, Mark M. Hall, Evangelos Kanoulas, and Mark Sanderson. Evaluating retrieval over sessions: The TREC session track 2011-2014. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 685–688. ACM, 2016.

[18] Marc Cavazza, Fred Charles, and Steven J. Mead. Planning characters' behaviour in interactive storytelling. *Comput. Animat. Virtual Worlds*, 13(2):121–131, 2002.

[19] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pre-training tasks for embedding-based large-scale retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[20] Haonan Chen, Zhicheng Dou, Qiannan Zhu, Xiaochen Zuo, and Ji-Rong Wen. Integrating representation and interaction for context-aware document ranking. *ACM Trans. Inf. Syst.*, Mar 2022.

[21] Haonan Chen, Zhicheng Dou, Yutao Zhu, Zhao Cao, Xiaohua Cheng, and Ji-Rong Wen. Enhancing user behavior sequence modeling by generative tasks for session search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 180–190. ACM, 2022.

[22] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Tiangong-st: A new dataset with large-scale refined real-world web search sessions. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2485–2488. ACM, 2019.

[23] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[24] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. Attention-based hierarchical neural query suggestion. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1093–1096. ACM, 2018.

[25] Seth Chin-Parker and Julie Cantelon. Contrastive constraints guide explanation-based category learning. *Cogn. Sci.*, 41(6):1645–1655, 2017.

[26] Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xiujun Li, Michel Galley, Chris Brockett, Mengdi Wang, and Jianfeng Gao. Towards coherent and cohesive long-form text generation. In *Proceedings of the First Workshop on Narrative Understanding*, pages 1–11, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[27] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[28] Paul R Cohen. *Empirical methods for artificial intelligence*, volume 139. MIT press Cambridge, MA, 1995.

[29] Kenneth Mark Colby. *Artificial Paranoia: A Computer Simulation of Paranoid Process*. Pergamon Press, 1975.

[30] Andrew M. Dai and Quoc V. Le. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3079–3087, 2015.

[31] Zhuyun Dai and Jamie Callan. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 985–988. ACM, 2019.

[32] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. Cast 2020: The conversational assistance track overview. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020.

[33] Alexandra DeLucia, Aaron Mueller, Xiang Lisa Li, and João Sedoc. Decoding methods for neural narrative generation. *CoRR*, abs/2010.07375, 2020.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[35] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information*

Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 766–774, 2014.

[36] Jeffrey L. Elman. Finding structure in time. *Cogn. Sci.*, 14(2):179–211, 1990.

[37] Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.

[38] Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics, 2018.

[39] Angela Fan, Mike Lewis, and Yann N. Dauphin. Strategies for structuring story generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2650–2660. Association for Computational Linguistics, 2019.

[40] Hongchao Fang and Pengtao Xie. CERT: contrastive self-supervised learning for language understanding. *CoRR*, abs/2005.12766, 2020.

[41] Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. Topic-to-essay generation with neural networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4078–4084. ijcai.org, 2018.

[42] Nicola Ferro, Claudio Lucchese, Maria Maistro, and Raffaele Perego. Continuation methods and curriculum learning for learning to rank. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1523–1526. ACM, 2018.

[43] Ronald Aylmer Fisher. Design of experiments. *British Medical Journal*, 1(3923):554, 1936.

[44] Ken-ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801–806, 1993.

[45] Shubhashri G, Unnamalai N, and Kamalika G. LAWBO: a smart lawyer chatbot. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, COMAD/CODS 2018, Goa, India, January 11-13, 2018*, pages 348–351. ACM, 2018.

[46] Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational AI. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1371–1374. ACM, 2018.

[47] Jianfeng Gao, Xiaodong He, and Jian-Yun Nie. Clickthrough-based translation models for web search: from word models to phrase models. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1139–1148. ACM, 2010.

[48] Luyu Gao, Zhuyun Dai, and Jamie Callan. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *ECIR 2021*, volume 12657 of *Lecture Notes in Computer Science*, pages 280–286. Springer, 2021.

[49] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *CoRR*, abs/2104.08821, 2021.

[50] Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. Personalizing search results using hierarchical RNN with query-aware attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 347–356. ACM, 2018.

[51] Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás. Story plot generation based on CBR. *Knowl. Based Syst.*, 18(4-5):235–242, 2005.

[52] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5110–5117, 2018.

[53] Chen Gong, Dacheng Tao, Stephen J. Maybank, Wei Liu, Guoliang Kang, and Jie Yang. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Trans. Image Process.*, 25(7):3249–3260, 2016.

[54] Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2041–2044. ACM, 2020.

[55] Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. Speaker-aware BERT for multi-turn response selection in retrieval-based chatbots. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2041–2044. ACM, 2020.

[56] Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. Interactive matching network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 2321–2324. ACM, 2019.

[57] Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, Zhigang Chen, and Xiaodan Zhu. Filtering before iteratively referring for knowledge-grounded response selection in retrieval-based chatbots. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 1412–1422. Association for Computational Linguistics, 2020.

[58] Jia-Chen Gu, Zhen-Hua Ling, Xiaodan Zhu, and Quan Liu. Dually interactive matching network for personalized response selection in retrieval-based chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1845–1854. Association for Computational Linguistics, 2019.

[59] Dongyi Guan, Sicong Zhang, and Hui Yang. Utilizing query change for session search. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 453–462. ACM, 2013.

[60] Jian Guan, Yansen Wang, and Minlie Huang. Story ending generation with incremental encoding and commonsense knowledge. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6473–6480. AAAI Press, 2019.

[61] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. *CoRR*, abs/2011.01403, 2020.

[62] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. A deep look into neural ranking models for information retrieval. *Inf. Process. Manag.*, 57(6):102067, 2020.

[63] Christophe Van Gysel and Maarten de Rijke. Pytrec_eval: An extremely fast python interface to trec_eval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 873–876. ACM, 2018.

[64] Christophe Van Gysel, Evangelos Kanoulas, and Maarten de Rijke. Lexical query modeling in session search. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, pages 69–72. ACM, 2016.

[65] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR, 2019.

[66] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742. IEEE Computer Society, 2006.

[67] Janghoon Han, Taesuk Hong, Byoungjae Kim, Youngjoong Ko, and Jungyun Seo. Fine-grained post-training for improving retrieval-based dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1549–1558. Association for Computational Linguistics, 2021.

[68] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[69] Matthew Henderson, Blaise Thomson, and Steve J. Young. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*, pages 292–299. The Association for Computer Linguistics, 2014.

[70] Erin Higgins and Brian Ross. Comparisons in category learning: How best to compare for what. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society, CogSci 2011, Boston, Massachusetts, USA, July 20-23, 2011.* cognitivesciencesociety.org, 2011.

[71] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[72] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2042–2050, 2014.

[73] Kai Hua, Zhiyuan Feng, Chongyang Tao, Rui Yan, and Lu Zhang. Learning to detect relevant contexts and knowledge for response selection in retrieval-based dialogue systems. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 525–534. ACM, 2020.

[74] Jizhou Huang, Wei Zhang, Yaming Sun, Haifeng Wang, and Ting Liu. Improving entity recommendation with search log and multi-task learning. In *Proceedings of the Twenty-Seventh International*

Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, pages 4107–4114. ijcai.org, 2018.

[75] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2333–2338. ACM, 2013.

[76] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2333–2338. ACM, 2013.

[77] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Narrative text generation with a latent discrete plan. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3637–3650. Association for Computational Linguistics, 2020.

[78] Zongcheng Ji, Zhengdong Lu, and Hang Li. An information retrieval approach to short text conversation. *CoRR*, abs/1408.6988, 2014.

[79] Jyun-Yu Jiang and Wei Wang. RIN: reformulation inference network for context-aware query suggestion. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 197–206. ACM, 2018.

[80] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pages 154–161. ACM, 2005.

[81] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2):7, 2007.

[82] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547, 2021.

[83] Rosie Jones and Kristina Lisa Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 699–708. ACM, 2008.

[84] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul A. Crook, Y-Lan Boureau, and Jason Weston. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1951–1961. Association for Computational Linguistics, 2019.

[85] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics, 2020.

[86] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single QA system. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020,*

Online Event, 16-20 November 2020, pages 1896–1907. Association for Computational Linguistics, 2020.

[87] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM, 2020.

[88] Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 329–339. The Association for Computational Linguistics, 2016.

[89] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1328–1338. The Association for Computational Linguistics, 2016.

[90] Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. Sequential latent knowledge selection for knowledge-grounded dialogue. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[91] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[92] Kai A Krueger and Peter Dayan. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394, 2009.

[93] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[94] Esther Levin, Shrikanth S. Narayanan, Roberto Pieraccini, Konstantin Biatov, Enrico Bocchieri, Giuseppe Di Fabbrizio, Wieland Eckert, Sungbok Lee, A. Pokrovsky, Mazin G. Rahim, P. Ruscitti, and Marilyn A. Walker. The at&t-darpa communicator mixed-initiative spoken dialog system. In *Sixth International Conference on Spoken Language Processing, ICSLP 2000 / INTERSPEECH 2000, Beijing, China, October 16-20, 2000*, pages 122–125. ISCA, 2000.

[95] Esther Levin, Roberto Pieraccini, and Wieland Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Trans. Speech Audio Process.*, 8(1):11–23, 2000.

[96] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL 2020*, pages 7871–7880. Association for Computational Linguistics, 2020.

[97] Jia Li, Chongyang Tao, Wei Wu, Yansong Feng, Dongyan Zhao, and Rui Yan. Sampling matters! an empirical study of negative sampling strategies for learning of matching models in retrieval-based dialogue systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1291–1296. Association for Computational Linguistics, 2019.

[98] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*

San Diego California, USA, June 12-17, 2016, pages 110–119. The Association for Computational Linguistics, 2016.

[99] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics, 2016.

[100] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[101] Jiwei Li and Eduard H. Hovy. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 2039–2048. ACL, 2014.

[102] Jiwei Li and Dan Jurafsky. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 198–209. Association for Computational Linguistics, 2017.

[103] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9748–9758, 2018.

[104] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. Learning to select knowledge for response generation in dialog systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5081–5087. ijcai.org, 2019.

[105] Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1489–1498. Association for Computational Linguistics, 2018.

[106] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4487–4496. Association for Computational Linguistics, 2019.

[107] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[108] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[109] Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. Towards conversational recommendation over multi-type dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1036–1049. Association for Computational Linguistics, 2020.

[110] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[111] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 285–294. The Association for Computer Linguistics, 2015.

[112] Jiyun Luo, Sicong Zhang, and Hui Yang. Win-win search: dual-agent stochastic game in session search. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pages 587–596. ACM, 2014.

[113] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. PROP: pre-training with representative words prediction for ad-hoc retrieval. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pages 283–291. ACM, 2021.

[114] Zhengyi Ma, Zhicheng Dou, Guanyue Bian, and Ji-Rong Wen. PSTIE: time information enhanced personalized search. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 1075–1084. ACM, 2020.

[115] Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. One chatbot per person: Creating personalized chatbots based on implicit user profiles. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 555–564. ACM, 2021.

[116] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. Training curricula for open domain answer re-ranking. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 529–538. ACM, 2020.

[117] Kelong Mao, Zhicheng Dou, Hongjin Qian, Fengran Mo, Xiaohua Cheng, and Zhao Cao. Convtrans: Transforming web search sessions for conversational dense retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2935–2946. Association for Computational Linguistics, 2022.

[118] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2775–2779. Association for Computational Linguistics, 2018.

[119] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2775–2779. Association for Computational Linguistics, 2018.

[120] James R. Meehan. Tale-spin, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977*, pages 91–98. William Kaufmann, 1977.

[121] Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. COCO-LM: correcting and contrasting text sequences for language model pretraining. *CoRR*, abs/2102.08473, 2021.

[122] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[123] Tomás Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048. ISCA, 2010.

[124] Tomás Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pages 5528–5531. IEEE, 2011.

[125] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013.

[126] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1291–1299. ACM, 2017.

[127] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 845–854. Association for Computational Linguistics, 2019.

[128] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[129] Yifan Nie, Jiyang Zhang, and Jian-Yun Nie. Integrated learning of features and ranking function in information retrieval. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2019, Santa Clara, CA, USA, October 2-5, 2019*, pages 67–74. ACM, 2019.

[130] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *CoRR*, abs/1901.04085, 2019.

[131] Alice Oh and Alexander I. Rudnicky. Stochastic natural language generation for spoken dialog systems. *Comput. Speech Lang.*, 16(3-4):387–407, 2002.

[132] OpenAI. Introducing chatgpt.

[133] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155, 2022.

[134] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2793–2799. AAAI Press, 2016.

[135] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems, Infoscale 2006, Hong Kong, May 30-June 1, 2006*, volume 152 of *ACM International Conference Proceeding Series*, page 1. ACM, 2006.

[136] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.

[137] P Ivan Pavlov. Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex. *Annals of neurosciences*, 17(3):136, 2010.

[138] Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[139] Gustavo Penha and Claudia Hauff. Curriculum learning strategies for IR: an empirical study on conversation response ranking. *CoRR*, abs/1912.08555, 2019.

[140] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.

[141] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.

[142] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1162–1172. Association for Computational Linguistics, 2019.

[143] Hongjin Qian and Zhicheng Dou. Explicit query rewriting for conversational dense retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4725–4737. Association for Computational Linguistics, 2022.

[144] Hongjin Qian, Zhicheng Dou, Yutao Zhu, Yueyuan Ma, and Ji-Rong Wen. Learning implicit user profile for personalized retrieval-based chatbot. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, CIKM 2021, Virtual Event, Australia, November 1-5, 2021*, pages 1467–1477, New York, NY, USA, 2021. ACM.

[145] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Assigning personality/profile to a chatting machine for coherent conversation generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4279–4285. ijcai.org, 2018.

[146] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. Understanding the behaviors of BERT in ranking. *CoRR*, abs/1904.07531, 2019.

[147] Jinghui Qin, Zheng Ye, Jianheng Tang, and Xiaodan Liang. Dynamic knowledge routing network for target-guided open-domain conversation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8657–8664. AAAI Press, 2020.

[148] Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, Bill Dolan, Yejin Choi, and Jianfeng Gao. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5427–5436. Association for Computational Linguistics, 2019.

[149] Chen Qu, Chenyan Xiong, Yizhe Zhang, Corby Rosset, W. Bruce Croft, and Paul Bennett. Contextual re-ranking with behavior aware transformers. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1589–1592. ACM, 2020.

[150] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5835–5847. Association for Computational Linguistics, 2021.

[151] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[152] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[153] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. Conversational AI: the science behind the alexa prize. *CoRR*, abs/1801.03604, 2018.

[154] William J Rapaport, Erwin M Segal, Stuart C Shapiro, David A Zubin, Gail A Bruder, Judith Felson Duchan, and David M Mark. Cognitive and computer systems for understanding narrative text. 1989.

[155] Mark O. Riedl and Robert Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.

[156] Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 583–593. ACL, 2011.

[157] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[158] Melissa Roemmele. Identifying sensible lexical relations in generated stories. In *Proceedings of the First Workshop on Narrative Understanding*, pages 44–52, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[159] Douglas LT Rohde and David C Plaut. Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 72(1):67–109, 1999.

[160] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022.

[161] Stephanie Seneff, Edward Hurley, Raymond Lau, Christine Pao, Philipp Schmid, and Victor Zue. GALAXY-II: a reference architecture for conversational system development. In *The 5th International Conference on Spoken Language Processing, Incorporating The 7th Australian International Speech Science and Technology Conference, Sydney Convention Centre, Sydney, Australia, 30th November - 4th December 1998*. ISCA, 1998.

[162] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3776–3784. AAAI Press, 2016.

[163] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3295–3301. AAAI Press, 2017.

[164] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1577–1586. The Association for Computer Linguistics, 2015.

[165] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pages 43–50. ACM, 2005.

[166] Heung-Yeung Shum, Xiaodong He, and Di Li. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers Inf. Technol. Electron. Eng.*, 19(1):10–26, 2018.

[167] Burrhus F Skinner. Reinforcement today. *American Psychologist*, 13(3):94, 1958.

[168] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In Mário J. Silva, Alberto H. F. Laender, Ricardo A. Baeza-Yates, Deborah L. McGuinness, Bjørn Olstad, Øystein Haug Olsen, and André O. Falcão, editors, *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 623–632. ACM, 2007.

[169] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 553–562. ACM, 2015.

[170] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 196–205. The Association for Computational Linguistics, 2015.

[171] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[172] Yixuan Su, Deng Cai, Qingyu Zhou, Zibo Lin, Simon Baker, Yunbo Cao, Shuming Shi, Nigel Collier, and Yan Wang. Dialogue response selection with hierarchical curriculum learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1740–1751. Association for Computational Linguistics, 2021.

[173] Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. *CoRR*, abs/2202.06417, 2022.

[174] Yixuan Su, Fangyu Liu, Zaiqiao Meng, Tian Lan, Lei Shu, Ehsan Shareghi, and Nigel Collier. Tacl: Improving BERT pre-training with token-aware contrastive learning. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2497–2507. Association for Computational Linguistics, 2022.

[175] Zhan Su, Zhicheng Dou, Yutao Zhu, Xubo Qin, and Ji-Rong Wen. Modeling intent graph for search result diversification. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 736–746. ACM, 2021.

[176] Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric P. Xing, and Zhiting Hu. Target-guided open-domain conversation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5624–5634. Association for Computational Linguistics, 2019.

[177] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 267–275, 2019.

[178] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. One time of interaction may not be enough: Go deep with an interaction-over-interaction network for response selection in dialogues. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1–11. Association for Computational Linguistics, 2019.

[179] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, volume 12356 of *Lecture Notes in Computer Science*, pages 776–794. Springer, 2020.

[180] Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. How to make context more useful? an empirical study on context-aware neural conversational models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 231–236. Association for Computational Linguistics, 2017.

[181] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.

[182] Gokhan Tur and Renato De Mori. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley & Sons, 2011.

[183] Alan M. Turing. Computing machinery and intelligence. In *The Philosophy of Artificial Intelligence*, Oxford readings in philosophy, pages 40–66. Oxford University Press, 1990.

[184] Alan M Turing. Computing machinery and intelligence. In *Parsing the turing test*, pages 23–65. Springer, 2009.

[185] Scott R Turner. Minstrel: A computer model of creativity and storytelling. 1994.

[186] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[187] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.

[188] Ellen M. Voorhees. The TREC-8 question answering track report. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 1999.

[189] Dennis Wackerly, William Mendenhall, and Richard L Scheaffer. *Mathematical statistics with applications*. Cengage Learning, 2014.

[190] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2835–2841. AAAI Press, 2016.

[191] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2835–2841. AAAI Press, 2016.

[192] Shuting Wang, Zhicheng Dou, and Yutao Zhu. Heterogeneous graph-based context-aware document ranking. In *WSDM '23: The Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, Singapore, February 27-March 3, 2023*. ACM, 2023.

[193] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR, 2020.

[194] Yeyi Wang, Li Deng, and Alex Acero. Semantic frame-based spoken language understanding. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pages 41–91, 2011.

[195] Joseph Weizenbaum. ELIZA - a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, 1966.

[196] Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection. *CoRR*, abs/2009.04703, 2020.

[197] Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14041–14049. AAAI Press, 2021.

[198] Ryen W. White, Paul N. Bennett, and Susan T. Dumais. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1009–1018. ACM, 2010.

[199] Jason D. Williams and Steve J. Young. Partially observable markov decision processes for spoken dialog systems. *Comput. Speech Lang.*, 21(2):393–422, 2007.

[200] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[201] Bin Wu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Query suggestion with feedback memory network. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1563–1571. ACM, 2018.

[202] Chao-Chung Wu, Ruihua Song, Tetsuya Sakai, Wen-Feng Cheng, Xing Xie, and Shou-De Lin. Evaluating image-inspired poetry generation. In *Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9-14, 2019, Proceedings, Part I*, volume 11838 of *Lecture Notes in Computer Science*, pages 539–551. Springer, 2019.

[203] Ledell Yu Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. Starspace: Embed all the things! In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5569–5577, 2018.

[204] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 3794–3804. Association for Computational Linguistics, 2019.

[205] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 496–505. Association for Computational Linguistics, 2017.

[206] Yu Wu, Wei Wu, Dejian Yang, Can Xu, and Zhoujun Li. Neural response generation with dynamic vocabularies. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5594–5601. AAAI Press, 2018.

[207] Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. CLEAR: contrastive learning for sentence representation. *CoRR*, abs/2012.15466, 2020.

[208] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware ranking in web search. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 451–458. ACM, 2010.

[209] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. Topic aware neural response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3351–3357. AAAI Press, 2017.

[210] Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. Hierarchical recurrent attention network for response generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5610–5617. AAAI Press, 2018.

[211] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 55–64. ACM, 2017.

[212] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[213] Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6095–6104. Association for Computational Linguistics, 2020.

[214] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2324–2335. Association for Computational Linguistics, 2019.

[215] Wei Xu and Alexander I. Rudnicky. Task-based dialog management using an agenda. In *ANLP-NAACL 2000 Workshop: Conversational Systems*, 2000.

[216] Rafael Pérez y Pérez and Mike Sharples. MEXICA: A computer model of a cognitive account of creative writing. *J. Exp. Theor. Artif. Intell.*, 13(2):119–139, 2001.

[217] Rui Yan, Yiping Song, and Hua Wu. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 55–64. ACM, 2016.

[218] Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of BERT for ad hoc document retrieval. *CoRR*, abs/1903.10972, 2019.

[219] Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence,*

EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 7378–7385. AAAI Press, 2019.

[220] Yuhang Ye, Zhonghua Li, Zhicheng Dou, Yutao Zhu, Changwang Zhang, Shangquan Wu, and Zhao Cao. Learning from the wisdom of crowds: Exploiting similar sessions for session search. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, The Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*. AAAI Press, 2023.

[221] Keen You and Dan Goldwasser. "where is this relationship going?": Understanding relationship trajectories in narrative text. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics, *SEM@COLING 2020, Barcelona, Spain (Online), December 12-13, 2020*, pages 168–178. Association for Computational Linguistics, 2020.

[222] Steve J. Young, Milica Gasic, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Comput. Speech Lang.*, 24(2):150–174, 2010.

[223] Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 111–120. Association for Computational Linguistics, 2019.

[224] Hao Yuan and Jinqi An. Multi-hop memory network with graph neural networks encoding for proactive dialogue. In *ICCAI '20: 2020 6th International Conference on Computing and Artificial Intelligence, Tianjin, China, April 23-26, 2020*, pages 24–29. ACM, 2020.

[225] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Optimizing dense retrieval model training with hard negatives. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1503–1512. ACM, 2021.

[226] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics, 2018.

[227] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022.

[228] Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J. Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. An empirical exploration of curriculum learning for neural machine translation. *CoRR*, abs/1811.00739, 2018.

[229] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 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, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics, 2020.

[230] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3740–3752. Association for Computational Linguistics, 2018.

[231] Xueliang Zhao, Chongyang Tao, Wei Wu, Can Xu, Dongyan Zhao, and Rui Yan. A document-grounded matching network for response selection in retrieval-based chatbots. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5443–5449, 2019.

[232] Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3377–3390. Association for Computational Linguistics, 2020.

[233] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 730–739. AAAI Press, 2018.

[234] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4623–4629. ijcai.org, 2018.

[235] Kangyan Zhou, Shrimai Prabhumoye, and Alan W. Black. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 708–713. Association for Computational Linguistics, 2018.

[236] Kangyan Zhou, Shrimai Prabhumoye, and Alan W. Black. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 708–713. Association for Computational Linguistics, 2018.

[237] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1118–1127. Association for Computational Linguistics, 2018.

[238] Yujia Zhou, Zhicheng Dou, Bingzheng Wei, Ruobing Xie, and Ji-Rong Wen. Group based personalized search by integrating search behaviour and friend network. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 92–101. ACM, 2021.

[239] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. Encoding history with context-aware representation learning for personalized search. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1111–1120. ACM, 2020.

[240] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. Enhancing re-finding behavior with external memories for personalized search. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 789–797. ACM, 2020.

[241] Yujia Zhou, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. PSSL: self-supervised learning for personalized search with contrastive sampling. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 2749–2758. ACM, 2021.

[242] Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. Flexible end-to-end dialogue system for knowledge grounded conversation. *CoRR*, abs/1709.04264, 2017.

[243] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society, 2015.

[244] Yutao Zhu, Zhicheng Dou, Jian-Yun Nie, and Ji-Rong Wen. Reboost: a retrieval-boosted sequence-to-sequence model for neural response generation. *Inf. Retr. J.*, 23(1):27–48, 2020.

[245] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. Contrastive learning of user behavior sequence for context-aware document ranking. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 2780–2791. ACM, 2021.

[246] Yutao Zhu, Jian-Yun Nie, Yixuan Su, Haonan Chen, Xinyu Zhang, and Zhicheng Dou. From easy to hard: A dual curriculum learning framework for context-aware document ranking. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 2784–2794. ACM, 2022.

[247] Yutao Zhu, Jian-Yun Nie, Kun Zhou, Pan Du, and Zhicheng Dou. Content selection network for document-grounded retrieval-based chatbots. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I*, volume 12656 of *Lecture Notes in Computer Science*, pages 755–769. Springer, 2021.

[248] Yutao Zhu, Jian-Yun Nie, Kun Zhou, Pan Du, Hao Jiang, and Zhicheng Dou. Proactive retrieval-based chatbots based on relevant knowledge and goals. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2000–2004. ACM, 2021.

[249] Yutao Zhu, Ruihua Song, Zhicheng Dou, Jian-Yun Nie, and Jin Zhou. Scriptwriter: Narrative-guided script generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8647–8657. Association for Computational Linguistics, 2020.

[250] Yutao Zhu, Ruihua Song, Jian-Yun Nie, Pan Du, Zhicheng Dou, and Jin Zhou. Leveraging narrative to generate movie script. *ACM Trans. Inf. Syst.*, 40(4):86:1–86:32, 2022.

[251] Yutao Zhu, Kun Zhou, Jian-Yun Nie, Shengchao Liu, and Zhicheng Dou. Neural sentence ordering based on constraint graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14656–14664. AAAI Press, 2021.

[252] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6001–6011. IEEE, 2019.

[253] Xiaochen Zuo, Zhicheng Dou, and Ji-Rong Wen. Improving session search by modeling multi-granularity historical query change. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 1534–1542. ACM, 2022.