Information Retrieval Journal (2022) 25:123–148
https://doi.org/10.1007/s10791-022-09403-0

SPECIAL ISSUE ON ECIR 2021

Open-domain conversational search assistants: the Transformer is all you need

Rafael Ferreira[1] · **Mariana Leite**[1] · **David Semedo**[1] · **Joao Magalhaes**[1]

Received: 9 July 2021 / Accepted: 5 February 2022 / Published online: 14 March 2022

Abstract
On the quest of providing a more natural interaction between users and search systems, open-domain conversational search assistants have emerged, by assisting users in answering questions about open topics in a conversational manner. In this work, we show how the Transformer architecture achieves state-of-the-art results in key IR tasks, leveraging the creation of conversational assistants that engage in open-domain conversational search with single, yet informative, answers. In particular, we propose a complete open-domain abstractive conversational search agent pipeline to address two major challenges: first, conversation context-aware search and second, abstractive search-answers generation. To address the first challenge, the conversation context is modeled using a query rewriting method that unfolds the context of the conversation up to a specific moment to search for the correct answers. These answers are then passed to a Transformer-based re-ranker to further improve retrieval performance. The second challenge, is tackled with recent Abstractive Transformer architectures to generate a digest of the top most relevant passages. Experiments show that Transformers deliver a solid performance across all tasks in conversational search, outperforming several baselines. This work is an expanded version of Ferreira et al. (Open-domain conversational search assistant with transformers. In: Advances in information retrieval—43rd European conference on IR research, ECIR 2021, virtual event, 28 March–1 April 2021, proceedings, Part I. Springer) which provides more details about the various components of the of the system, and extends the automatic evaluation with a novel user-study, which confirmed the need for the conversational search paradigm, and assessed the performance of our answer generation approach.

Keywords Conversational search · Transformers · Query rewriting · Re-ranking · Answer generation

This submission is an expanded and enhanced version of the paper Open-Domain Conversational Search Assistant with Transformers published in the 43rd European Conference on IR Research, 2021 (Ferreira et al. 2021).

✉ Rafael Ferreira
rah.ferreira@campus.fct.unl.pt

Extended author information available on the last page of the article

1 Introduction

The research area of Conversational Information Seeking (CIS) is emerging as a future trend in the field of Information Retrieval (Culpepper et al. 2018), as the natural evolution of the traditional search paradigm, aiming for a more natural interaction between users and search systems such as in Fig. 1. Building intelligent systems able to establish and develop this level of meaningful conversations is one of the key goals of Information Retrieval, AI, and the ultimate goal of Natural Language research (Dinan et al. 2019). Vtyurina et al. (2017) studied the interactions between a user and conversational systems and showed that users are willing to utilise conversational assistants as long as their needs are met with success. However, conversational search assistants still put a considerable burden on users that for each question, have to go through a list of documents, or passages, to find the information they need.

The goal of satisfying information needs through a conversational interface triggers the usage of a different type of dialogue, which is set apart from chit-chat and task-oriented ones, for its intrinsic and unique format type, where co-references, ambiguity and subtopic shifts can take place. This form of dialogue, which is comprised of several turns, is addressed by Conversational Search systems that are responsible for collecting relevant information in order to answer each query in the course of the dialogue correctly. These intelligent systems not only should be able to satisfy the information need unravelled in conversational searches but also provide straightforward answers as opposed to a ranked list of relevant documents (Gao et al. 2019). This jointly enables a more natural conversation with the intelligent agent and reduces the information seeking burden on the user.

In this paper, we take a conversational approach to the classic information seeking problem, and investigate how this novel paradigm holds on an end-to-end open-domain abstractive conversational setting. In Ferreira et al. (2021), we explored the Transformer's (Devlin

#	Sender	Message
0	user	What is throat cancer?
1	assistant	Throat cancer refers to cancerous tumors that develop in your throat (pharynx), voice box (larynx) or tonsils. Your throat is a muscular tube that begins behind your nose and ends in your neck. Your voice box is made of cartilage and contains the vocal cords that vibrate to make sound when you talk.
2	user	Is it treatable?
3	assistant	Throat cancers may be cured when detected early. If the cancer has spread to surrounding tissues or lymph nodes in the neck, about half of patients can be cured. Treatment is aimed at prolonging and improving quality of life. After treatment, therapy is needed to help with speech and swallowing.
4	user	Tell me about lung cancer.
5	assistant	Lung cancer is a malignant lung tumor characterized by uncontrolled cell growth. If left untreated, this growth can spread beyond the lung by the process of metastasis. The two main types are small-cell lung carcinoma (SCLC) and non-small-cell Lung carcinoma. The most common symptoms are coughing, weight loss, shortness of breath, and chest pains.
6	user	What are its symptoms?
7	assistant	Throat cancer symptoms include hoarseness, a lump in the neck, sore throat, cough, problems breathing, bad breath, earache, and weight loss. Causes of throat cancer include smoking and excess alcohol consumption.

Fig. 1 Conversational search example of the proposed system for different turns in an open-domain conversation topic available at https://kwiz.ai/treccast2019

et al. 2019) advantages to: (a) capture complex relations between conversation turns to rewrite a query in the middle of a conversation; (b) look into the interactions between words in a conversation query and a candidate passage; and (c) compress multiple retrieved passages into one single, yet informative, search-answer. In this paper, we extend (Ferreira et al. 2021) and delve into the details of each stage of the proposed pipeline (Query Rewriting, First-stage retrieval, Passage Re-Ranking, and Abstractive Search-Answer Generation) and provide an extended and comprehensive analysis of the experiments. To further consolidate the outputs, we conducted new experiments with human assessors, revealing new insights. Hence, the core contributions of this paper are the following:

- **Transformers for IR.** In the proposed end-to-end conversational search pipeline the Transformer is a global solution for many traditional IR problems achieving state-of-the-art results.
- **Search with single-answers.** Instead of providing the users with a list of search results, abstractive answer generation can effectively compress the information of several retrieved passages into a short answer, as evidenced by automatic and human evaluation results. Ultimately, the user can explore the passages that support the generated answer.
- **User study.** The conversational search paradigm with single-answer was evaluated with real users in a user study who assessed the different answer generation algorithms. Results showed that in conversational search, the single generated answer provided a better user experience than the top retrieved passage.
- **Functioning prototype.** To demonstrate the feasibility of our work, we implemented a prototype of the entire pipeline described in this paper.[1] In addition, the results of our various answer generation models are available for inspection as in Fig. 1.

This extended work aims at providing a better comprehension of the challenges and characteristics of the TREC CAsT (Conversational Assistant Track) dataset (Dalton et al. 2020b), followed by a novel user study where the full conversational search assistant is evaluated. We performed a statistical analysis of the TREC CAsT dataset and asked human annotators to classify the type of each query and whether they are context-dependent. We expanded our results to include these query-type annotations, bringing new and valuable insights for this task. We also added the results of our various retrieval baselines to clearly show the impact of initial retrieval models in a conversational scenario. To complement our previous analysis using automatic metrics in the conversational answer summarisation/generation task, we conducted a user study using crowdsourcing. This user study allowed us to obtained novel insights regarding the effectiveness of the proposed methods, under two distinct aspects not fully covered by automatic metrics: Information quality, and Naturalness and Conciseness. Finally, we released an inspection tool to interact with the TREC CAST 2019 results of this paper.[2]

In the following section, we discuss the related work. In Sect. 3 we detail the Transformer-based conversational search pipeline: the conversational query rewriting, the first-stage retriever, the re-ranker, and abstractive answer generation. Extensive evaluation of the developed architecture including the performed user study is available in Sect. 4

[1] The prototype is available for testing at https://kwiz.ai/, and a video demonstrating the prototype is provided in https://youtu.be/VE_rSuNiiXg.

[2] https://kwiz.ai/treccast2019/.

and Sect. 5 presents the key takeaway messages. The code to reproduce our experimental results is publicly available.[3]

2 Related work

Open-domain conversational search systems must account for the dialog context to provide a relevant passage. While research on interactive search systems has started long ago (Belkin 1980; Croft and Thompson 1987; Oddy 1977), the recent interest in having intelligent conversation assistants (e.g. Alexa, SIRI), has re-ignited this research field. Recent models (Dinan et al. 2019; Lin et al. 2020; Qu et al. 2020; Voskarides et al. 2020) leverage large open-domain collections (e.g. Wikipedia) to learn rich language-models using self-supervised neural networks. The applicability of these models in conversational search is twofold: grasping the dialog context and passage re-ranking. Recently, the TREC CAsT (Dalton et al. 2020a) task introduced a multi-turn passage retrieval dataset, enabling the development and evaluation of such models.

Conversational context-aware search models in its simplest form need to (1) keep track of the dialog context, and (2) select the most relevant passage.

2.1 Tracking the context

To address (1), we highlight two general approaches to keep track of context: implicitly or explicitly. In the former, we highlight HAE (Qu et al. 2019a, b), where an extra layer is added to learn historical embeddings which are summed to BERT's original input embeddings (Devlin et al. 2019). Given that these methods do not change the content of the query, they are not suited for our scenario, where a retrieval step is required. On the contrary, in the latter, explicit context-independent queries can be obtained by performing coreference resolution or query rewriting.

In coreference resolution the aim is to detect mentions in text like pronouns and connect them to the appropriate subject, removing the ambiguity of the language. In Lee et al. (2017), a combination of CNNs and RNNs is used to detect and resolve coreferences. In Joshi et al. (2020), a BERT model (Devlin et al. 2019) is used for the same task, with an optimization objective based on corrupting spans of text, during training and fine-tuning, greatly increasing performance over (Lee et al. 2017).

Simple coreference resolution may be not enough to fully track the context. Thus, another line of work focuses on rewriting the entire query. Elgohary et al. (2019) observed that manually rewritten queries from QuAC (Choi et al. 2018) had enough context to be independently understandable. To automate the process, a sequence-to-sequence (seq2seq) model with attention and a copy mechanism was proposed. The model is given as input a sequence with the full conversation history and the query to be rewritten. In Voskarides et al. (2020), a BERT model (Devlin et al. 2019) is given as input a sequence of all terms of the current and previous queries, and is then fine-tuned on a binary term classification task. Also using both the query and conversation history, in Lin et al. (2020), a pre-trained Text-to-text Transfer Transformer (T5) model (Raffel et al. 2020) is fine-tuned on CANARD (Elgohary et al. 2019) to construct the context-independent query, and achieved

[3] https://github.com/novasearch/conversational-search-assistant-transformers.

state-of-the-art performance on the query-rewriting task. Keeping the tendency of using Transformer models, in Yu et al. (2020) a weakly supervised method based on the GPT-2 model (Radford et al. 2019) is trained on very limited manual query rewrites with results similar to the ones obtained in Lin et al. (2020).

2.2 Selecting the most relevant passages

Task (2) is commonly addressed through re-ranking. A number of works (Dai et al. 2018; Xiong et al. 2017; Yang et al. 2018) have explored the use of neural architectures to build re-ranking models by calculating a measure of similarity between the words in the query and each candidate passage.

More recently, large pre-trained Transformer models, such as BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), and XLNet (Yang et al. 2019b), have been widely adopted for re-ranking due to their generalisation capabilities. Examples of this are present in Han et al. (2020), Nogueira and Cho (2019), and Nogueira et al. (2019) where a Transformer-based model is fine-tuned on the question-answering relevance classification task by jointly encoding the text in the query and passage (cross-encoder). In the same line of work in Nogueira et al. (2020) the seq2seq Transformer T5 (Raffel et al. 2020) is used for ranking, outperforming previous encoder-only methods (Nogueira and Cho 2019).

Alternatively, there are approaches that separately encode the query and the passage, such as bi-encoders (Dinan et al. 2019; Mazaré et al. 2018; Khattab and Zaharia 2020) and poly-encoders (Humeau et al. 2020). These methods first extract embeddings for the full set of retrievable documents, from a pre-trained backbone model, that can be optionally fine-tuned (Lee et al. 2019; Guu et al. 2020). Then, at query time, they only encode the query and calculate a similarity measure between the query embedding and all of the documents embeddings using efficient methods. By not requiring to jointly encode each query–passage pair at query time, these approaches are more efficient and can potentially deal with larger pools of candidate passages. We also highlight Xiong et al. (2021), where a negative example selection method is proposed to more effectively train these types of models.

2.3 Generating answers

In this work, we go a step beyond conventional conversational search and introduce an answer generation/summarisation component, that given the dialogue context, requires the agent to generate a natural language response.

In chit-chat dialogue generation, most approaches use an encoder–decoder neural architecture that first encodes utterances and then the decoder generates a response (Li et al. 2016, 2017; Song et al. 2018; Tian et al. 2019; Zhuang et al. 2017). In Li et al. (2016, 2017), reinforcement learning is used to overcome uninformative and general responses of standard seq2seq models.

Another alternative is retrieval-based dialogue generation, in which the generator takes as input retrieved candidate documents to improve the comprehensiveness of the generated answer (Song et al. 2018; Zhuang et al. 2017). These approaches require a large dataset with annotated dialogues, which is not feasible in our scenario. Alternatively, Transformer models have shown to be highly effective generative language models (Lewis et al. 2020; Raffel et al. 2020; Zhang et al. 2020). While both T5 (Raffel et al. 2020) and BART (Lewis et al. 2020) are general language models, PEGAGUS (Zhang et al. 2020) focuses

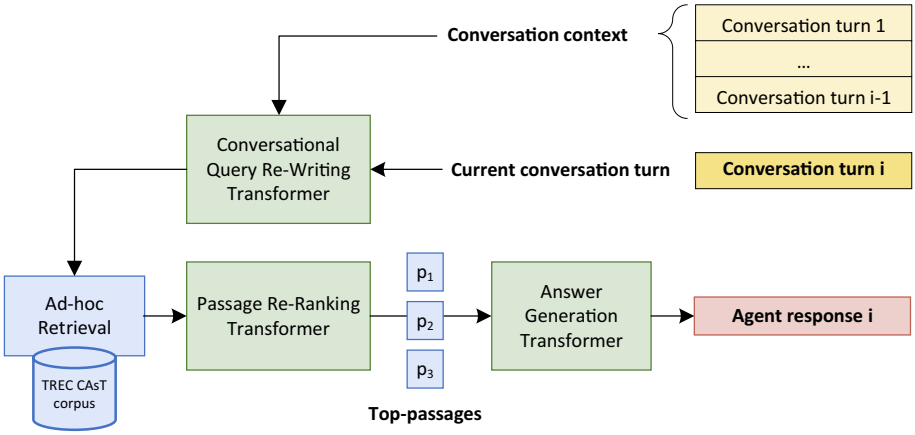


Fig. 2 The proposed Transformer-based conversational search assistant

on abstractive summarisation, and obtained state-of-the-art results on 12 summarisation tasks. Following such findings, recent works in abstractive summarisation such as ProphetNet (Qi et al. 2020) and GSum (Dou et al. 2021) adopted the Transformer architecture. ProphetNet (Qi et al. 2020) is a seq2seq model that to generate more coherent responses, uses a novel optimisation objective that predicts the following n tokens based on the context tokens in one single step. GSum uses a BART model (Lewis et al. 2020) with guidance signals obtained from BERT to bias generation on the most relevant sentences. While GSum is currently state-of-the-art on the task of abstractive summarisation in the CNN/Daily Mail dataset (Hermann et al. 2015), the use of two large Transformer models makes it a highly computationally expensive model.

3 Transformer-based conversational search assistant

In this section, we formulate the open-domain conversational search task and describe the conversational assistant retrieval and answer generation components, which constitute the pipeline of our system. The conversational search task is formally defined by a sequence of natural language conversational turns for a topic T, with queries q. Hence, for each conversation topic we have n conversation turns,

$$T = \{q_1, \ldots, q_i, \ldots, q_n\}, \tag{1}$$

and the conversational search task is to find relevant passages p_k for each query q_i, satisfying the user's information need for that turn according to the conversational context. The proposed approach uses a four-stage architecture: (a) context tracking, (b) retrieval, (c) re-ranking, and (d) answer generation. An overview of the system's architecture can be seen in Fig. 2 which we will detail in the following sections.

Table 1 Conversation example about a specific topic, in this case the City of Lucca

Turn	Conversational Query	Context-independent Query
1	How is the climate in Lucca?	How is the climate in Lucca?
2	Tell me about its origins.	Tell me about Lucca's origins.
3	What monuments should I visit?	What monuments should I visit in Lucca?

Corefences are highlighted with underlines

3.1 Conversational query rewriting transformer

Due to the evolving nature of a conversational session, the current query is likely to not include all the information needed to retrieve the answer that the user is looking for. This challenge is illustrated in the conversation presented in Table 1, in turn 2, where the system needs to understand that "its" refers to "Lucca's" (explicit coreference), and in turn 3, where the important monuments should be focused in Lucca, although there is no direct evidence (implicit coreference), which makes the task even more challenging. We tackle this challenge by rewriting queries, using information from previous turns, making the current query context-independent.

To perform the query rewriting task, we need a model capable of performing coreference resolution and include context from previous turns. The T5 (Raffel et al. 2020) can be fine-tuned to reformulate conversational queries (Lin et al. 2020) by providing as input the sequence of conversational queries and passages, and as target, the rewritten query. In particular, we constructed the training input sequence as follows:

$$q_i\ [CTX]\ q_1\ p_1\ [TURN]\ q_2\ p_2\ [TURN]\ \ldots\ [TURN]\ q_{i-1}\ p_{i-1}, \tag{2}$$

where i is the current turn, q_i is a query, p_k is a passage retrieved from the index by the retrieval model, and $[CTX]$ and $[TURN]$ are special tokens. $[CTX]$ is used to separate the current query q_i from the context (previous queries and passages) and $[TURN]$ is used to separate each historical turn (query–passage pair). At training time the aim is to use the T5 model to rewrite the query q_i by using as target its non-conversational rewritten version. This approach is similar to the one in Lin et al. (2020), however, the creation of the input sequence differs, since we separate at every turn (query–passage pair) instead of at every utterance in order to clearly delimiter each turn.

3.2 First-stage retrieval

In the first-stage retrieval, since applying our re-ranking Transformer to the full set of passages is infeasible, similarly to Nogueira and Cho (2019), Nogueira et al. (2019) and Yang et al. (2019a) we considered term-matching retrieval models, BM25 (Robertson and Zaragoza 2009), and language models with Dirichlet (LMD) and Jelinek–Mercer (LMJM) smoothing (Zhai and Lafferty 2001), to recover a small set of passages from the millions of available passages, in a fast and effective fashion. From the passages retrieved by the retrieval model, we pass the top-n passages to our rich but more computationally demanding re-ranking model, that aims at improving the original rank.

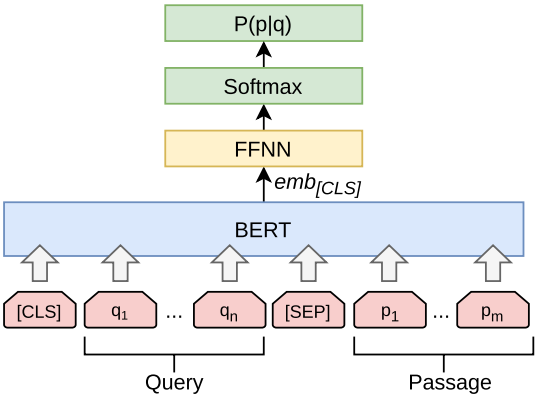


Fig. 3 BERT re-ranker architecture. The input is the query concatenated with each one of the passages at a time, using the structure [CLS] q [SEP] p

3.3 Passage re-ranking transformer

With transformer-based pre-trained neural language models, such as BERT (Devlin et al. 2019) and others (Liu et al. 2019; Yang et al. 2019b), it is possible to generate contextual embeddings for a sentence and each of its tokens. These embeddings can be used as input to a model to perform passage re-ranking (Nogueira and Cho 2019; Nogueira et al. 2019). Given the nature of contextual embeddings, which capture token's context within a sequence through a continuous representation (embedding), this re-ranking step allows going beyond term matching. Namely, after pre-training on large corpora, these language models acquire the capability of structuring individual terms as well as their interactions within sentences based on their semantics. This allows for a rich modeling of the interactions between queries and passages, thus yielding a more thorough judgement of relevance between a passage and a query.

Following this rationale, we tackle the passage re-ranking task with a BERT model (Devlin et al. 2019), fine-tuned on the passage ranking task (Nogueira and Cho 2019). To obtain the embedding of the query q, and passage p, a sequence with N tokens is given as input to BERT:

$$emb = BERT([CLS]\ q\ [SEP]\ p), \tag{3}$$

where $emb \in \mathbb{R}^{N \times H}$ is the embeddings matrix of all tokens (H is BERT embedding's size), and $[CLS]$ and $[SEP]$ are special tokens in BERT's vocabulary, representing the classification and separation tokens, respectively. From emb we extract the embedding of the first token, which corresponds to the embedding of the $[CLS]$ token, $emb_{[CLS]} \in \mathbb{R}^{H}$. This embedding is then used as input to a single layer feed-forward neural network (FFNN), followed by a *softmax*, to obtain the probability of the passage being relevant to the query:

$$P(p|q) = softmax(\text{FFNN}(emb_{[CLS]})). \tag{4}$$

With $P(p|q)$ calculated for each passage p given a query q, the final rank is obtained by re-ranking according to the probability of being relevant. The described re-ranking architecture can be seen in Fig. 3.

To fine-tune the model, we followed prior work (Nogueira and Cho 2019) and considered the task of binary relevance classification using the cross-entropy loss defined as:

$$L_q = -\sum_{j \in J_{pos}} \log(P(p_j|q)) - \sum_{j \in J_{neg}} \log(1 - P(p_j|q)), \tag{5}$$

where for a query q, J_{pos} are positive examples (relevant passages) and J_{neg} are negative examples (non-relevant passages) from the MS MARCO dataset (Nguyen et al. 2016) and $P(p_j|q)$ is the score given by BERT to that query–passage pair.

3.4 Abstractive search-answer generation transformer

Having identified a set of candidate passages according to the scores given by the re-ranker model (Eq. 4), the goal is to generate a natural language response that combines the information comprised in each of the passages. To address this, we follow an abstractive summarisation approach, which unlike extractive summarisation that just selects existing sentences, it can portray both reading comprehension and writing abilities, thus allowing the generation of a concise and comprehensive digest of multiple input passages.

The Transformer (Vaswani et al. 2017) architecture has proved to be highly effective at modelling large dependency windows of textual sequences. Text-to-text approaches (Lewis et al. 2020; Raffel et al. 2020; Zhang et al. 2020), trained over large and comprehensive collections, become effective at *understanding* different topics and retaining language regularities useful for several language tasks. Thus, to generate the agent's response using a transformer model, we give as input the following sequence:

$$p_1\, p_2\, \cdots\, p_N, \tag{6}$$

where each p_k, with $k \in [1, N]$, corresponds to each of top-N candidate passages. With this strategy, we implicitly bias the answer generation by asking the model to summarise the passages that are deemed as more relevant according to the retrieval component.

The implicit bias of the top passages is crucial to steer the Transformer response generation. The sequence of passages of Eq. 6 is given as input to the Transformer, which will then jointly attend to the different passages. As the multi-head attention layers look across the different passages, redundant parts will be merged, while the remaining information will be summarised, leading to a concise but comprehensive answer. The following Transformer models were considered for the task of abstractive summarisation.

3.4.1 Text-to-Text Transfer Transformer

The T5 (Raffel et al. 2020) is a text-to-text model based on the traditional encoder–decoder Transformer architecture, with a small change in its positional embeddings, which are learned at each layer. With this architecture, it is able to perform several NLP tasks that are transformed into text-to-text problems by using specific prefixes. For the summarisation task, we include "summarise:" at the beginning of the to-be-processed text.

As it is common with Transformer based models, this model went through a pre-training step. The T5 pre-training was performed on the large C4 corpus, which was derived from Common Crawl,[4] and involved both supervised and self-supervised training. The former was performed on the GLUE (Wang et al. 2018) and SuperGLUE (Wang et al. 2019)

[4] https://commoncrawl.org/.

benchmarks, each comprising a collection of text classification tasks meant to test general language understanding abilities.

The self-supervised training makes use of a masked language modelling objective, by removing and replacing 15% of sentence tokens of varying sizes with sentinel tokens. This corrupted sentence is then given as input to the encoder while the original is given to the decoder. The model is trained to predict the replaced tokens, which are delimited by their sentinel tokens.

3.4.2 BART

BART (Lewis et al. 2020) is a denoising autoencoder, that combines a Bidirectional Transformer encoder and an Auto-Regressive Transformer decoder. The pre-training tasks consists of corrupting text with an arbitrary noising function and learning an autoencoder to reconstruct the original text. The corrupted tokens are fed to the encoder while the decoder is fed the original tokens.

The best performing noise functions were text infilling (using single mask tokens to mask random sampled spans of text), and sentence shuffling (changing the order of sentences in passages). For the summarisation task, the strategy used to add noise to the text was using single mask tokens to mask random sampled spans of text.

3.4.3 PEGASUS

The Pre-training with Extracted Gap-sentences for Abstractive SUmmarization Sequence-to-sequence model (PEGASUS; Zhang et al. 2020) specialises on the abstractive summarisation task. It is a sequence-to-sequence model with the same encoder–decoder architecture of BART. PEGASUS was pre-trained jointly with two self-supervised objective tasks:

- Masked Language Modeling—in which the tokens given as input to the encoder are randomly replaced by mask tokens and the encoder, much like BERT, must predict the original tokens.
- Gap Sentence Generation—a novel summarisation specific pre-training objective in which whole sentences are replaced by a second mask token.

PEGASUS stands out from BART and T5 for not being a general language model and having a pre-training task which is intentionally similar to summarisation. The masked sentences are chosen for being important to the whole context and not by random chance. The output sentences are generated together as one output sequence from the remaining sentences.

Table 2 TREC CAsT dataset analysis

Parameter	Train set	Evaluation set	
		Original	Manual
# Conversations	30	50	50
# Judged conversations	13 (43.33%)	20 (40%)	20 (40%)
# Turns	269	479	479
# Judged turns	120 (44.6%)	194 (40.50%)	194 (40.50%)
# Turns where context is needed	79 (65.83%)	125 (64.43%)	0 (0%)
Avg. # turns	8.96 1.45	9.58 1.20	9.58 1.20
Avg. # terms per query	7.33 2.05	7.14 2.01	8.68 2.47
Avg. # judged docs per query*	19.99	151.28	151.28
Avg. # relevant docs per query*	5.33	41.85	41.85

*Considering only judged turns

4 Evaluation

4.1 Datasets and experimental protocol

4.1.1 CANARD dataset (Elgohary et al. 2019)

This dataset was used to train and evaluate the query rewriting method. It was created by manually rewriting the queries in QuAC (Choi et al. 2018) to form non-conversational queries. The training, development, and test sets have 31.538, 3.418, and 5.571, query-rewrites respectively.

4.1.2 TREC CAsT dataset (Dalton et al. 2020b)

This dataset was used to evaluate both the conversational search and answer generation components. The passage collection is composed by MS MARCO (Nguyen et al. 2016), TREC CAR (Dietz et al. 2018), and WaPo (NIST 2019) datasets, which creates a complete pool of close to 47 million passages.

TREC CAsT dataset analysis. Table 2 shows a summary of the information available in the dataset. The evaluation set complements its *Original* conversational queries with *Manual* queries, which are the same queries, but manually rewritten to a non-conversational format. In total there are 30 training topics and 50 evaluation topics with 43.33% and 40.50% topics available with relevance judgement labels respectively, amounting to a total of 269 and 479 turns for training and evaluation.

The average number of turns is similar in both datasets and were labelled on average until turn depth 8, using a graded relevance of 0–2 in training and 0–4 in evaluation, where 0 corresponds to irrelevant and a higher number corresponds to a more relevant query–passage pair. It is also important to note that the process used to collect annotations for the training and evaluation sets were different, which justifies the difference between number of judged documents in each split (Dalton et al. 2020a).

To better assess the conversational nature of the dataset, we performed a study where five annotators classified whether it was possible to answer each individual query of the dataset without the context, showing us that about 65% of queries are conversational, demonstrating the importance of context in this dataset.

Finally the difference in the number of terms between the *Original* and *Manual* queries in the evaluation set evidences one of the characteristics of conversational search, where conversational queries (*Original*) are smaller than their non-conversational counterparts (*Manual*).

4.1.3 Experimental protocols

To analyse query rewriting performance, we used the BLEU-4 score (Papineni et al. 2002) between the model's output and the queries rewritten by humans, on the CANARD dataset.

In the passage retrieval experiment, we used the TREC CAsT setup and the official metrics, nDCG@3 (normalised Discounted Cumulative Gain at 3), MAP (Mean Average Precision), and MRR (Mean Reciprocal Rank), along with Recall and P@3 (Precision at 3).

In the answer generation experiment, we used METEOR and the ROUGE variant ROUGE-L. For each query in TREC CAsT, we used as reference passages, all the passages with a relevance judgement of 3 and 4. Hence, the goal is to generate answers that cover, as much as possible, the information contained in all relevant passages, in one concise and summarised answer.

4.2 Implementation

4.2.1 Query rewriting

We fine-tuned the T5 (Raffel et al. 2020) model according to Lin et al. (2020) using standard maximum likelihood on the CANARD's training set (Elgohary et al. 2019), providing as input the concatenation of the conversational queries and passages, and as target the rewritten query. In particular, we used the T5-BASE model and trained for 4000 steps, using a maximum input sequence length of 512 tokens, a maximum output sequence length of 64 tokens, a learning rate of 0.0001, and batches of 256 sequences.

4.2.2 First-stage retrieval

To index and search, we used the well tuned Anserini framework (Yang et al. 2017), in particular, the Python implementation Pyserini.[5] We applied stop word removal, using Luceneâ€™s default list, and stemming using Kstem.[6] We experimented with: BM25 (Robertson and Zaragoza 2009) and language models with Dirichlet (LMD) and Jelinek–Mercer (LMJM) smoothing (Zhai and Lafferty 2001), tuning the parameters to maximise recall in the training set with the aim of having the greatest amount of relevant documents for the re-ranking phase.

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

[6] http://lexicalresearch.com/kstem-doc.txt.

Table 3 BLEU-4 scores for the CANARD test set and for TREC CAsT using the manually rewritten queries of the evaluation set

	CANARD	TREC CAsT
Human (Elgohary et al. 2019)	59.92	–
Raw (Elgohary et al. 2019)	47.44	–
T5-BASE (Lin et al. 2020)	**58.08**	75.07
Our T5-BASE	56.84	**79.67**

Bold indicates best performance

Table 4 Example of query rewriting inputs, targets and predictions

CANARD	
Original Query	What was his agreement with McMahon?
T5 Input Query	What was his agreement with McMahon? [CTX] Superstar Billy Graham. Return to WWWF (1977-1981) [TURN] Why did he return to the WWWF? An agreement with promoter Vincent J. McMahon Senior.
T5 Predicted Query	What was Superstar Billy Graham's agreement with McMahon?
Target Query	What was Billy Graham's agreement with McMahon?
TREC CAsT 2019	
Original Query	What are its symptoms?
T5 Input Query	What are its symptoms? [CTX] What is throat cancer? [TURN] Is throat cancer treatable? [TURN] Tell me about lung cancer.
T5 Predicted Query	What are throat cancer's symptoms?
Target Query	What are lung cancer's symptoms?
Original Query	What are some of the possible causes?
T5 Input Query	What are some of the possible causes? [CTX] Tell me about the Bronze Age collapse? [TURN] What is the evidence for the Bronze Age collapse?
T5 Predicted Query	What are some of the possible causes for the Bronze Age collapse?
Target Query	What are some of the possible causes of the Bronze Age collapse?

Corefences are highlighted with underlines

4.2.3 BERT passage re-ranker

To perform re-ranking, we used the BERT model implementation from Huggingface (Wolf et al. 2020). Following the state-of-the-art (Nogueira and Cho 2019; Nogueira et al. 2019), we used the LARGE version of BERT with a classification layer (FFNN) on top, that takes as input the query–passage *CLS token* embeddings vector generated by BERT, and classifies the passage as relevant or non-relevant to that query. This model was trained following (Nogueira and Cho 2019) on the MS MARCO dataset (Nguyen et al. 2016) composed of 12.8 million non-conversational query–passage pairs. In testing, we truncate the concatenation of the query, passage, and separator tokens to a maximum of 512 tokens (the maximum number of tokens for the BERT model).

4.2.4 Transformer based answer generation

To generate the summarised answers, we employed the T5-BASE, BART-LARGE and PEGASUS models (Wolf et al. 2020). The T5-BASE has about 220 million parameters with 12 layers, 768 hidden-state size, 3072 feed-forward hidden-states and 12 heads. BART-LARGE holds about 406 million parameters, with a 12-layer, 1024 hidden state size and 16-head architecture. The PEGASUS model has the biggest number of parameters, 568 million, with 16 layers, 1024 hidden state size and 16-heads.

Table 5 Results of retrieval on the TREC CAsT training set

Queries	Retrieval model	Recall	P@3	MAP	MRR	nDCG@3
Original	BM25	0.480	0.083	0.091	0.140	0.079
Original	LMD	0.508	0.089	0.115	0.154	0.082
Original	LMJM	0.402	0.047	0.044	0.089	0.045
T5	BM25	0.779	0.186	0.207	0.326	0.185
T5	LMD	**0.790**	**0.194**	**0.251**	**0.338**	**0.195**
T5	LMJM	0.737	0.100	0.116	0.205	0.109

Bold indicates best performance

Table 6 Results of retrieval on the TREC CAsT evaluation set

Queries	**Re-ranker**	**Recall**	**P@3**	**MAP**	**MRR**	**nDCG@3**
Original	-	0.454	0.262	0.141	0.336	0.167
Original	BERT	0.454	0.385	0.181	0.456	0.272
T5	-	0.697	0.474	0.251	0.597	0.322
T5	BERT	0.697	0.632	**0.310**	**0.739**	**0.475**
TREC CAsT baselines						
clacBase	-	0.695	0.534	0.246	0.640	0.360
HistoricalQE	BERT	0.611	0.580	0.267	0.715	0.436
Manual baselines						
Manual	-	0.820	0.590	0.327	0.694	0.406
Manual	BERT	0.820	0.757	0.389	0.857	0.577

Bold indicates best performance

The HistoricalQE (Yang et al. 2019a) was the best performing model in TREC CAsT 2019

All models were fine-tuned on the summarising task with the CNN/Daily Mail dataset (Hermann et al. 2015). To generate the summary, we use 4 beams, restrict the n-grams of size 3 to only occur once, and allow for beam search early stopping when at least 4 sentences are generated. Additionally, we fix the maximum length of the summary to be of the same length of the input given to the models (which corresponds to 3 passages) and vary the minimum length from 20 to 120 words.

4.3 Experimental results

4.3.1 Conversation-aware query rewriting

In Table 3, we show the BLEU-4 scores obtained in CANARD's test set and in TREC CAsT's 2019 manually rewritten queries. The rows "Human" and "Raw" are from Elgohary et al. (2019), the row "T5-BASE" is from Lin et al. (2020). The last row corresponds to our implementation. Our results are on par with citet5conversational, being lower in the CANARD dataset but higher in TREC CAsT. We believe the minor differences in performance between our T5-BASE model and the T5-BASE from Lin et al. (2020) are due to the use of different input sequences, as the exact method of constructing the input is not specified in Lin et al. (2020).

From the analysis of the BLEU-4 scores and outputs, we can conclude that the model is performing both coreference and context resolution, approximating the queries in a

conversational format to context-independent queries. Examples of the inputs, targets, and predicted queries, are presented in Table 4. In TREC CAsT, the historical utterances do not depend on the responses of the system, so the answer is not provided as input to the model. As we can see, T5 is capable of resolving ambiguous queries by co-reference resolution, as in example 1, but sometimes mistakes similar co-references when multiple are involved, as evidenced in example 2 and in Lin et al. (2020), where the model predicts "throat cancer" instead of "lung cancer". We can also note that this model is more robust than just coreference resolution, as seen in example 3, where it includes the words "Bronze Age Collapse", even though there is no explicit mention (implicit coreference).

4.3.2 First-stage retrieval

In Table 5, we show the results of the first-stage retrieval step on the TREC CAsT training set. *Original* are the conversational queries (lower-bound) and *T5* are the queries generated by the developed query rewriting method. The results show that rewriting the utterances is essential to improve performance with a large increase in all metrics. When comparing the various retrieval models LMD showed the best results, confirming previous knowledge (Zhai and Lafferty 2001) and matching the shorter queries that we observe in a conversational search scenario. For this reason, LMD was the model chosen to perform retrieval for all following experiments.

4.3.3 Transformer-based passage reranking

Table 6 shows the results of retrieval on the TREC CAsT evaluation dataset. *Original* and *T5* are the same as in the previous section, *Manual* is a baseline where the queries were manually rewritten (upper-bound), and the other two lines are the results of baselines retrieved from Dalton et al. (2020a). *clacBase* (Clarke 2019) is a method that uses AllenNLP coreference resolution (Gardner et al. 2018) and a fine-tuned BM25 model with pseudo-relevance feedback, and HistoricalQE (Yang et al. 2019a) is a method that uses a query expansion algorithm based on session and query words together with a BERT LARGE model for re-ranking. The latter was the best performing method in terms of nDCG@3 in TREC CAsT 2019 (Dalton et al. 2020a).

Similarly to what happened in the training set with the various retrieval models the first observation that emerges from Table 6 is the clear need for a query rewriting method to maintain the conversational context, evidenced by the low scores on all metrics using the original conversational queries. Rewriting queries (with the *T5* model) outperforms the original conversational queries by a 5–20% margin (nDCG@3), thus showing the effectiveness of this approach. The second clear observation is again the considerable improvement when Transformers are used for re-ranking. In this case, the improvement is in the 10–15% range over standard retrieval metrics. This is due to the better understanding that the fine-tuned BERT model has of the interactions between the query and passage terms.

Finally, the largest gains emerge when we combine the two Transformers to deliver state-of-the-art results. With the proposed Transformers we outperform the best TREC CAsT 2019 baseline by 3.9% in terms of nDCG@3. We consider that this improvement is mainly due to the use of a better query-rewriting method that allows the retrieval model to retrieve passages given the conversational context, providing the re-ranker with more relevant passages.

Table 7 Query-type distribution and annotator agreement in TREC CAsT 2019 evaluation set

	# Queries	% Queries	Agreement (%)
Describe	97	56.07	92.0
Yes/No	15	8.67	94
List	34	19.65	83
Comparison and Connection	18	10.40	94
Compositional	9	5.2	82
All	173	100	90.2

4.3.4 Retrieval performance by query type

To obtain more details about the results obtained, we also performed an analysis of the results by query type considering the following five categories (Dalton et al. 2020b):

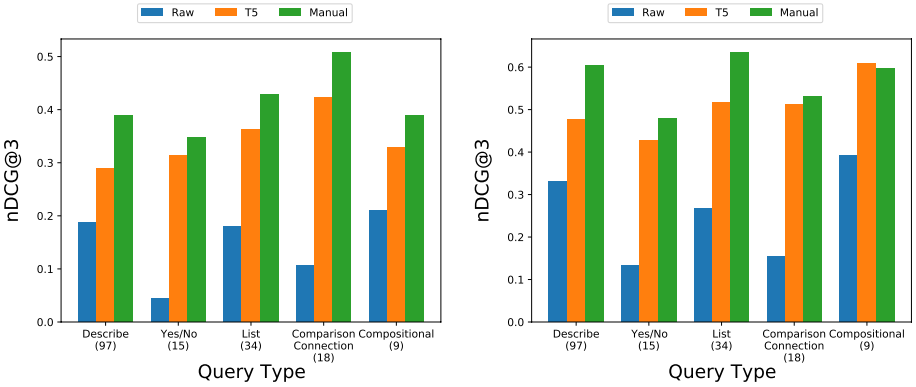
- *Describe* general description of the subject, e.g., "Tell me about the Bronze Age collapse."
- *Yes/No* answers the query and provides a brief justification, e.g., "Is the Coach Museum in Lisbon free?"
- *List* the answer is a list, e.g., "Who are the members of the Avengers?"
- *Comparison and Connection* the query contains comparisons or connections between concepts, e.g., "How does Netflix compare to Amazon Prime Video?"
- *Compositional* combination of two questions in a single query, e.g., "What is the Galileo system and why is it important?"

To categorise the dataset queries, we gathered five annotators that classified all of the queries. After this, we chose the mode of the category annotations as the final label for each query. We observed an average agreement of 90.2%, indicating that this task is not particularly difficult for humans. In the disagreeing cases, the most common case was a query having *Describe* category annotations, when the mode was of category *List*. We can also observed that more than half of the queries (56.07%) are of the type *Describe*, followed by 19.65% *List* queries. More details about the query types and their distribution over the dataset are available in Table 7.

Figure 4 shows the results of the different methods without and with BERT re-ranking.

Similarly to what we saw in Table 6 using the rewritten queries and a re-ranking step greatly improve performance in all query types indicating that these methods are generalizable to all types of queries. Perhaps unexpectedly, one of the best performing query types after re-ranking was *Compositional*, which in theory, are the most complex ones, so, after further analysis we concluded that 80% of these queries were not conversational and therefore are less susceptible to query-rewriting mistakes. Interestingly, *Yes/No* was the query type that achieved the lowest scores in most formulations, showing that retrieving just a *Yes/No* type of answer from a dataset containing passages can be a difficult task.

In summary, these results leave the door open for future work in methods that treat the various types of queries more specifically which in turn can improve the overall performance of the system.



(a) Query-types without re-ranking. **(b)** Query-types with BERT re-ranking.

Fig. 4 Query type results on TREC CAsT 2019 evaluation set. Values in parentheses indicate the number of queries of that type

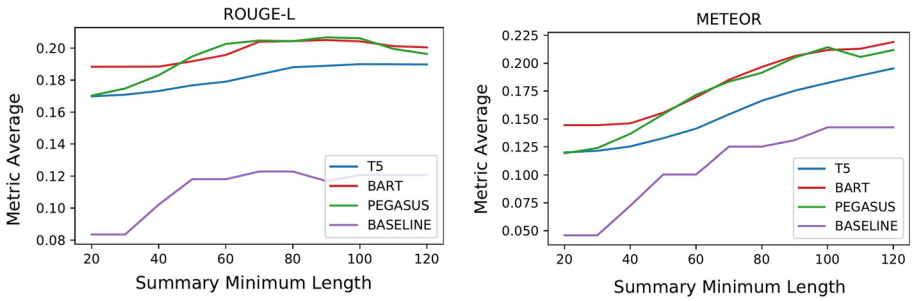


Fig. 5 Performance of the answer generation results under different metrics

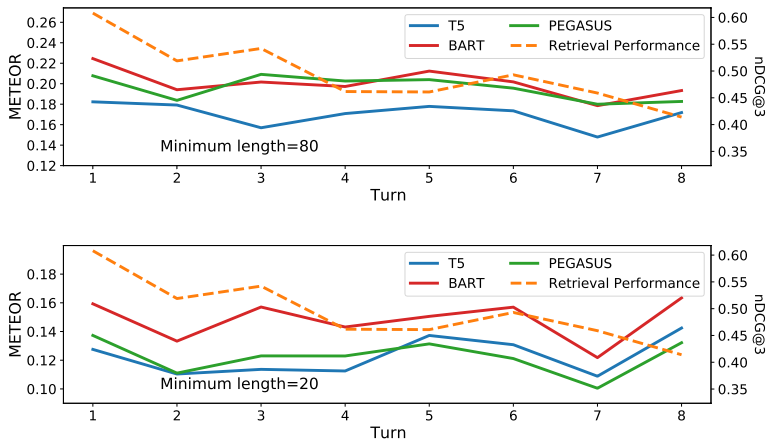


Fig. 6 Answer generation versus retrieval performance per conversation turn. The minimum length is 80 and 20 in the top and bottom graphs respectively

Table 8 Answer generation example for the turn "What was the first artificial satellite?"

Method	Answer
Retrieval Passage 1	The first artificial satellite was Sputnik 1, launched by the Soviet Union on October 4, 1957, and initiating the Soviet Sputnik program, with Sergei Korolev as chief designer (there is a crater on the lunar far side which bears his name). This in turn triggered the Space Race between the Soviet Union and the United States
Retrieval Passage 2	The first artificial Earth satellite was Sputnik 1. Put into orbit by the Soviet Union on October 4, 1957, it was equipped with an on-board radio-transmitter that worked on two frequencies: 20.005 and 40.002 MHz. Sputnik 1 was launched as a step in the exploration of space and rocket development. While incredibly important it was not placed in orbit for the purpose of sending data from one point on earth to another. And it was the first artificial satellite in the steps leading to today's satellite communications
Retrieval Passage 3	The first artificial satellite was Sputnik 1. It was the size of a basketball and was made by the USSR (Union of Soviet Socialist Republics) or Russia. It was launched on October 4, 1957
T5	*The first artificial satellite was Sputnik 1, launched by the* ***US or Russia***. **It was the size of a basketball and launched on October 4, 1957**. The satellite **was equipped with a radio-transmitter that worked on two frequencies**. *Incredibly important it was not placed in orbit for sending data from one point on earth to another. In turn, it triggered the space race between the united states and the Soviet Union*
BART	*The first artificial satellite was Sputnik 1, launched by the Soviet Union on October 4, 1957. It was equipped with an on-board radio-transmitter that worked on two frequencies: 20.005 and 40.002 MHz. This in turn triggered the Space Race between the Soviet Union and the United States*. **The size of a basketball, it was not placed in orbit for the purpose of sending data from one point on earth to another**. And it was the first Artificial satellite in the steps leading to today's satellite communications
PEGASUS	*The first artificial satellite was Sputnik 1, launched by the Soviet Union on October 4, 1957. Sputnik 1 was launched as a step in the exploration of space and rocket development. It was not placed in orbit for the purpose of sending data from one point on earth to another. This in turn triggered the Space Race between the* **USSR and the US**. ***There is a crater on the lunar far side which bears his name***

Summary minimum length is set to 90. Bold sentences illustrate abstractive, italics sentences illustrate extractive, and bold italics sentences illustrate incorrect summaries

4.3.5 Conversational answer generation

Figure 5 shows the result of the answer generation step according to the ROUGE-L and METEOR metrics. The baseline is composed by the concatenation of the top 3 passages, cropped to the maximum length of the passage according to the "Summary Minimum Length" value, respecting sentence endings. In Fig. 5, all answer generation models were better than the retrieval baseline method. According to ROUGE-L, the top performance is achieved around 60–90 word length answers. Since the goal is to generate short and informative answers, we were not interested in answers longer than 100 words. According to these results we observe that BART was the best answer generation method.

In Fig. 6 we analyse the retrieval and the answer generation performance over conversation turns. We see that peak performance is achieved on the first turn, which was expected since that it is the first turn that establishes the conversation topic. As the conversation progresses, retrieval performance decreases, but surprisingly, answer generation performance

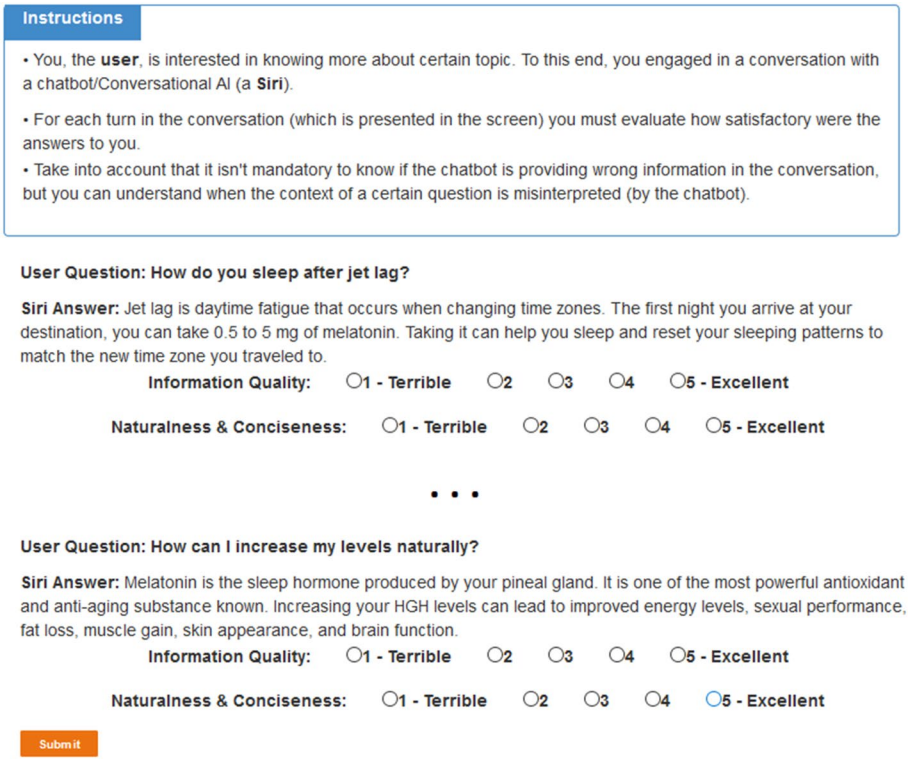


Fig. 7 Interface of the Amazon MTurk HIT which shows first and last conversation turn

is stable until the 6th turn. We also observed that the decrease in performance is linked to sub-topic shifts within the same conversation topic.

An interesting observation from Fig. 6 is that PEGASUS is the method that is most affected by retrieval performance. We believe this is related to its generation process that has a behaviour closer to extractive summarisation, while BART and T5 demonstrate a more abstractive behaviour.

4.3.6 Qualitative analysis

To complement the quantitative experiment based on automatic metrics, i.e. ROUGE-L and METEOR, we analysed the generated results and observed several patterns. Table 8 provides a good example of the answer generation behaviour with all three Transformers. This table further confirms the abstractive versus extractive summarisation behaviours of the different Transformer-based architectures. In this example we see that T5-BASE tries to generate new sentences by combining different sentences and PEGASUS makes use of verb synonyms not seen in text in order to convey the same message but with fewer words.

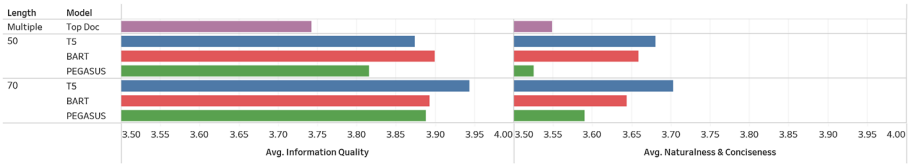


Fig. 8 Averaged values of the human evaluation in terms of Information Quality, Naturalness and Conciseness per combination of Length and Model

4.4 User study

We posit that in a conversational scenario, to ensure a rich interaction between an user and the agent, answers are expected to be informative, on point and complete, while being natural. To assess how the different proposed methods capture these aspects, we chose to manually evaluate them with regards to: information quality, conciseness and naturalness of the answers given in the conversations.

Accordingly, we conducted a human evaluation experiment on Amazon Mechanical Turk. In this experiment, to compare the answer generation methods and the top retrieved passage, we asked each Worker to rate each conversation turn on a 1–5 Likert scale, with higher being better, according to the following dimensions (Fig. 7):

- **Information Quality (IQ)** which aims to evaluate how well an answer addressed the query of the present turn, taking into account the context of the conversation.
- **Naturalness and Conciseness (NC)** which aims to evaluate if the answers can be perceived as being created by human beings—in such a manner that the flow of the different phrases, possibly from different sources, is coherent—with answers not including too much extraneous information.

Each task was independently performed by 4 different Workers which, to be able to partake in the task, had to present a minimum approval rate of 95% and had to at least have completed 100 Human Intelligence Tasks (HITs) already. Additionally, all HITs were inspected to the best of our ability. When for a single user a continuous session of HITs submission took place, the first and last submissions time and number of HITs performed were taken to calculate the average time spent per HIT. Users that showed an average value of less than 20 s had their submissions rejected and those HITs were re-submitted by other users. For each turn in a HIT, the final value was calculated as the average of the evaluation performed by 4 distinct Workers. Each HIT was completed for a reward of 0.04$. HITs were assigned to a total of 65 Workers, resulting in a total of 560 HITs. Since some Workers performed more HITs than others, to assess the impact of skewness in the HITs per Worker distribution, we experimented removing annotations from the most productive workers, and the conclusions remained the same.

In Fig. 8 we can see the obtained evaluation of IQ and NC that were averaged per row. Each row comprises of 20 different conversations with 8 turns each, that were individually evaluated by 4 different Workers, totalling in 160 ratings per conversation per method.

The overall IQ of the answers were rated higher than the NC. Moreover, we can easily see that the top retrieved passage was rated by users as the worst method in terms of IQ, and the best result, both in terms of IQ and NC was achieved with the combination

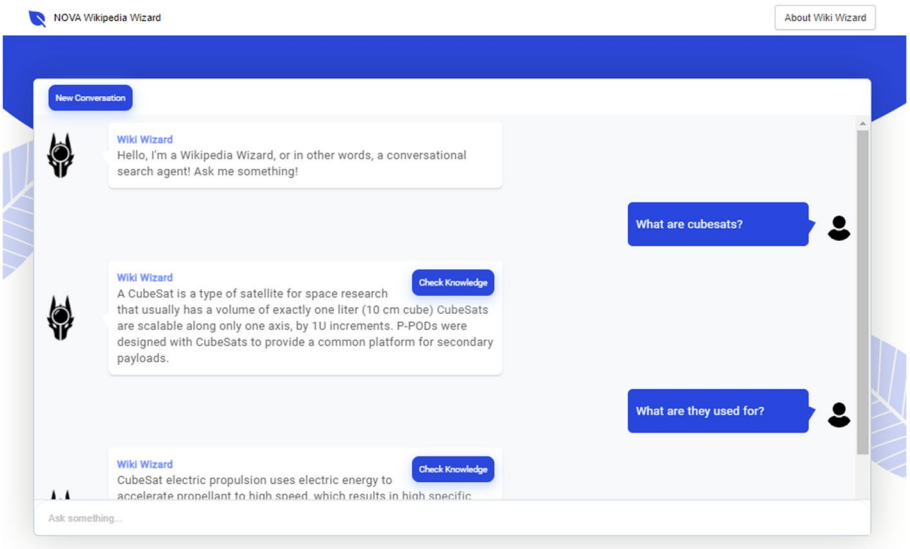


Fig. 9 Interface of our end-to-end open-domain conversational system prototype. A video demonstrating the Conversational Agent Prototype is provided as supplementary material in https://youtu.be/VE_rSuNiiXg

of the T5 model with answers of 70 words length. However, for shorter answers both BART and T5 were rated similarly. This leads to the important conclusion that retrieval results do effectively feed answer generation methods, enabling the generation of a summarised answer that better addresses the user's information need.

5 Conclusions

In this paper we investigated the conversational search paradigm in a truly conversational setting. In general, there are many successes across the full IR/NLP pipeline that leads to the success of the proposed pipeline. We demonstrated how Transformer architectures can address different tasks in open-domain conversational search, with particular emphasis on the search-answer generation task. The described system was implemented as a functioning prototype,[7] Fig. 9, and the results of the system were evaluated with crowd-workers. In summary, the key findings are as follows:

– **Conversational search paradigm.** The key takeaway from this paper is that the conversational search paradigm can be implemented as a summarisation approach. The user study clearly showed that users prefer this approach when searching for information in a conversational setting. Moreover, the implemented prototype demonstrated its viability with existing state-of-the-art algorithms.

[7] https://kwiz.ai.

- **Transformers for IR tasks.** Transformers can solve a number of tasks in conversational search, leading to new state-of-the-art results by outperforming the best TREC-CAsT 2019 baseline by 3.9% in terms of nDCG@3. This result is rooted on a fine-tuned bi-directional Transformer model (Raffel et al. 2020) for conversational query re-writing, which attained an improvement of 5–20% (nDCG@3) over raw conversational queries. Similarly, the re-ranking task using a fine-tuned BERT LARGE model (Nogueira and Cho 2019) improved results by 10–15% (nDCG@3) over an LMD model.
- **Search-Answer Generation.** Experiments showed that search systems can be improved with agents that abstract the information contained in multiple documents to provide a single and informative search answer. In terms of ROUGE-L we concluded that all answer generation models (Lewis et al. 2020; Raffel et al. 2020; Zhang et al. 2020) performed better than the retrieval baseline.
- **Abstractive vs. Extractive Answer Generation.** The examined answer generation Transformers revealed different behaviours. While the three models revealed an extractive behaviour, with input sentence fragments being included in the output summary, BART and T5 performed abstractive summarisation more often, by combining and rewriting answers with information from different passages. This approach turned out to be better than extractive methods that copy and paste sentences from different passages.

As future research, we plan to improve the conversational query rewriting methods and develop re-rankers with a notion of the context of the conversation. Another interesting topic is the mining of possible conversation paths to steer the answer generation process towards further helping the user in exploring alternative aspects of the searched topic.

Funding This work has been partially funded by the iFetch Project, Ref. 45920 co-financed by ERDF, COMPETE 2020, NORTE 2020, the Carnegie Mellon University Portugal Project GoLocal Ref. CMUP-ERI/TIC/0046/2014 and by the Project NOVA LINCS Ref. UID/CEC/04516/2013.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Belkin, N. J. (1980). Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science, 5*(1), 133–143.

Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W., Choi, Y., Liang, P., & Zettlemoyer, L. (2018). QuAC: Question answering in context. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, Brussels, Belgium, October 31–November 4, 2018 (pp. 2174–2184). Association for Computational Linguistics.

Clarke, C. L. A. (2019). WaterlooClarke at the TREC 2019 conversational assistant track. In E. M. Voorhees & A. Ellis (Eds.), *Proceedings of the twenty-eighth Text REtrieval Conference, TREC 2019*, Gaithersburg, Maryland, USA, November 13–15, 2019. NIST Special Publication (Vol. 1250). National Institute of Standards and Technology (NIST).

Croft, W. B., & Thompson, R. H. (1987). I3R: A new approach to the design of document retrieval systems. *JASIST, 38*(6), 389–404.

Culpepper, J. S., Diaz, F., & Smucker, M. D. (2018). Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in Lorne (SWIRL 2018). *SIGIR Forum, 52*(1), 34–90.

Dai, Z., Xiong, C., Callan, J., & Liu, Z. (2018). Convolutional neural networks for soft-matching n-grams in ad hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining, WSDM 2018*, Marina Del Rey, CA, USA, February 5-9, 2018 (pp. 126–134). ACM.

Dalton, J., Xiong, C., & Callan, J. (2020a). TREC CAsT 2019: The conversational assistance track overview. *CoRR*, abs/2003.13624.

Dalton, J., Xiong, C., & Callan, J. (2020b). The TREC conversational assistance track (CAsT).

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). 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. Long and Short Papers (Vol. 1, pp. 4171–4186). Association for Computational Linguistics.

Dietz, L., Gamari, B., & Dalton, J. (2018). TREC CAR 2.1: A data set for complex answer retrieval.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., & Weston, J. (2019). Wizard of Wikipedia: Knowledge-powered conversational agents. In *7th International conference on learning representations, ICLR 2019*, New Orleans, LA, USA, May 6–9, 2019. OpenReview.net.

Dou, Z., Liu, P., Hayashi, H., Jiang, Z., & Neubig, G. (2021). GSum: A general framework for guided neural abstractive summarization. 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 (pp. 4830–4842). Association for Computational Linguistics.

Elgohary, A., Peskov, D., & Boyd-Graber, J. L. (2019). Can you unpack that? Learning to rewrite questions-in-context. In K. Inui, J. Jiang, V. Ng & X. Wan (Eds.), *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 (pp. 5917–5923). Association for Computational Linguistics.

Ferreira, R., Leite, M., Semedo, D., & Magalhães, J. (2021). Open-domain conversational search assistant with transformers. In *Advances in information Retrieval—43rd European conference on IR research, ECIR 2021, virtual event, Proceedings, Part I*, March 28–April 1, 2021. Lecture notes in computer science (Vol. 12656, pp. 130–145). Springer.

Gao, J., Galley, M., & Li, L. (2019). Neural approaches to conversational AI. *Foundations and Trends in Information Retrieval, 13*(2–3), 127–298.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., & Zettlemoyer, L. (2018). AllenNLP: A deep semantic natural language processing platform. In *Proceedings of workshop for NLP open source software (NLP-OSS)*, Melbourne, Australia (pp. 1–6). Association for Computational Linguistics.

Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Han, S., Wang, X., Bendersky, M., & Najork, M. (2020). Learning-to-rank with BERT in TF-ranking. *CoRR*, abs/2004.08476.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems* (pp. 1693–1701).

Humeau, S., Shuster, K., Lachaux, M., & Weston, J. (2020). Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *8th International conference on learning representations, ICLR 2020*, Addis Ababa, Ethiopia, April 26–30, 2020. OpenReview.net.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics, 8,* 64–77.

Khattab, O., & Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen & Y. Liu (Eds.), *Proceedings of the 43rd international ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, virtual event*, China, July 25–30, 2020 (pp. 39–48). ACM.

Lee, K., Chang, M., & Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. In A. Korhonen, D. R. Traum & L. Màrquez (Eds.), *Proceedings of the 57th conference of the Association for Computational Linguistics, ACL 2019*, Florence, Italy, July 28–August 2, 2019. Long papers (Vol. 1, pp. 6086–6096). Association for Computational Linguistics.

Lee, K., He, L., Lewis, M., & Zettlemoyer, L. (2017). End-to-end neural coreference resolution. In *Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017*, Copenhagen, Denmark, September 9–11, 2017 (pp. 188–197). Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In D. Jurafsky, J. Chai, N. Schluter & J. R. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the Association for Computational Linguistics, ACL 2020, online*, July 5–10, 2020 (pp. 7871–7880). Association for Computational Linguistics.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., & Gao, J. (2016). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, Austin, Texas (pp. 1192–1202). Association for Computational Linguistics.

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., & Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, Copenhagen, Denmark (pp. 2157–2169). Association for Computational Linguistics.

Lin, S., Yang, J., Nogueira, R., Tsai, M., Wang, C., & Lin, J. (2020). Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *CoRR*, abs/2004.01909.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Mazaré, P., Humeau, S., Raison, M., & Bordes, A. (2018). 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 (pp. 2775–2779). Association for Computational Linguistics.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., & Deng, L. (2016). 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. CEUR workshop proceedings (Vol. 1773). CEUR-WS.org.

NIST. (2019). *TREC Washington Post corpus*.

Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Nogueira, R., Jiang, Z., Pradeep, R., & Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. In *Proceedings of the 2020 conference on empirical methods in natural language processing: Findings, EMNLP 2020, online event*, November 16–20, 2020 (pp. 708–718). Association for Computational Linguistics.

Nogueira, R., Yang, W., Cho, K., & Lin, J. (2019). Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424.

Oddy, R. N. (1977). Information retrieval through man–machine dialogue. *Journal of Documentation, 33*(1), 1–14.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA (pp. 311–318). Association for Computational Linguistics.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., & Zhou, M. (2020). ProphetNet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 conference on empirical methods in natural language processing: Findings, EMNLP 2020, online event*, November 16–20, 2020 (pp. 2401–2410). Association for Computational Linguistics.

Qu, C., Yang, L., Chen, C., Qiu, M., Croft, W. B., & Iyyer, M. (2020). Open-retrieval conversational question answering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, SIGIR '20*, New York, NY, USA (pp. 539–548). Association for Computing Machinery.

Qu, C., Yang, L., Qiu, M., Croft, W. B., Zhang, Y., & Iyyer, M. (2019a). BERT with history answer embedding for conversational question answering. In B. Piwowarski, M. Chevalier, É. Gaussier, Y. Maarek, J. Nie & F. Scholer (Eds.), *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, SIGIR 2019*, Paris, France, July 21–25, 2019 (pp. 1133–1136). ACM.

Qu, C., Yang, L., Qiu, M., Zhang, Y., Chen, C., Croft, W. B., & Iyyer, M. (2019b). Attentive history selection for conversational question answering. In W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He & J. X. Yu (Eds.), *Proceedings of the 28th ACM international conference on information and knowledge management, CIKM 2019*, Beijing, China, November 3–7, 2019 (pp. 1391–1400). ACM.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog, 1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*, 140.1–140.67.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval, 3*(4), 333–389.

Song, Y., Li, C.-T., Nie, J.-Y., Zhang, M., Zhao, D., & Yan, R. (2018). An ensemble of retrieval-based and generation-based human–computer conversation systems. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence, international joint conferences on artificial intelligence organization, IJCAI-18* (pp. 4382–4388).

Tian, Z., Bi, W., Li, X., & Zhang, N. L. (2019). Learning to abstract for memory-augmented conversational response generation. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, Florence, Italy (pp. 3816–3825). Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017*, December 4–9, 2017, Long Beach, CA, USA (pp. 5998–6008).

Voskarides, N., Li, D., Ren, P., Kanoulas, E., & de Rijke, M. (2020). Query resolution for conversational search with limited supervision. *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*.

Vtyurina, A., Savenkov, D., Agichtein, E., & Clarke, C. L. A. (2017). Exploring conversational search with humans, assistants, and wizards. In *Proceedings of the 2017 CHI conference extended abstracts on human factors in computing systems, CHI EA '17*, New York, NY, USA (pp. 2187–2193). Association for Computing Machinery.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). SuperGLUE: A stickier benchmark for general-purpose language understanding systems. 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 (pp. 3261–3275).

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the workshop: Analyzing and interpreting neural networks for NLP, BlackboxNLP@EMNLP 2018*, Brussels, Belgium, November 1, 2018 (pp. 353–355). Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, EMNLP 2020, demos, online* November 16–20, 2020 (pp. 38–45). Association for Computational Linguistics.

Xiong, C., Dai, Z., Callan, J., Liu, Z., & Power, R. (2017). 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 (pp. 55–64). ACM.

Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P. N., Ahmed, J., & Overwijk, A. (2021). 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.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019b). XLNet: Generalized autoregressive pretraining for language understanding. 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 (pp. 5754–5764).

Yang, P., Fang, H., & Lin, J. (2017). Anserini: Enabling the use of Lucene for information retrieval research. In N. Kando, T. Sakai, H. Joho, H. Li, A. P. de Vries & R. W. White (Eds.), *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, Shinjuku, Tokyo, Japan, August 7–11, 2017 (pp. 1253–1256). ACM.

Yang, Z., Lan, Q., Guo, J., Fan, Y., Zhu, X., Lan, Y., Wang, Y., & Cheng, X. (2018). A deep top-k relevance matching model for ad hoc retrieval. In *Information retrieval—24th China conference, CCIR 2018, proceedings*, Guilin, China, September 27–29, 2018. Lecture notes in computer science (Vol. 11168, pp. 16–27). Springer.

Yang, J., Lin, S., Wang, C., Lin, J., & Tsai, M. (2019a). Query and answer expansion from conversation history. In E. M. Voorhees & A. Ellis (Eds.), *Proceedings of the twenty-eighth Text REtrieval*

Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13–15, 2019. NIST special publication (Vol. 1250). National Institute of Standards and Technology (NIST).

Yu, S., Liu, J., Yang, J., Xiong, C., Bennett, P. N., Gao, J., & Liu, Z. (2020). Few-shot generative conversational query rewriting. 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 (pp. 1933–1936). ACM.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'01*, New York, NY, USA (pp. 334–342). Association for Computing Machinery.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th international conference on machine learning, ICML 2020, virtual event*, 13–18 July 2020. Proceedings of machine learning research (Vol. 119, pp. 11328–11339). PMLR.

Zhuang, Y., Wang, X., Zhang, H., Xie, J., & Zhu, X. (2017). An ensemble approach to conversation generation. In X. Huang, J. Jiang, D. Zhao, Y. Feng & Y. Hong (Eds.), *Natural language processing and Chinese computing—6th CCF international conference, NLPCC 2017, proceedings*, Dalian, China, November 8–12, 2017. Lecture notes in computer science (Vol. 10619, pp. 51–62). Springer.

Authors and Affiliations

Rafael Ferreira[1] · Mariana Leite[1] · David Semedo[1] · Joao Magalhaes[1]

Mariana Leite
me.leite@campus.fct.unl.pt

David Semedo
df.semedo@fct.unl.pt

Joao Magalhaes
jmag@fct.unl.pt

[1] NOVA LINCS, NOVA School of Science & Technology, Universidade NOVA de Lisboa, Lisbon, Portugal