Conversations with Search Engines: SERP-based Conversational Response Generation

Ren, P.; Chen, Z.; Ren, Z.; Kanoulas, E.; Monz, C.; de Rijke, M.

DOI
10.1145/3432726

Publication date
2021

Document Version
Author accepted manuscript

Published in
ACM Transactions on Information Systems

Conversations with Search Engines

PENGJIE REN, University of Amsterdam, The Netherlands
ZHUMIN CHEN, Shandong University, China
ZHAOCHUN REN, Shandong University, China
EVANGELOS KANOULAS, University of Amsterdam, The Netherlands
CHRISTOF MONZ, University of Amsterdam, The Netherlands
MAARTEN DE RIJKE, University of Amsterdam & Ahold Delhaize, The Netherlands

In this paper, we address the problem of answering complex information needs by conversing *conversations with search engines*, in the sense that users can express their queries in natural language, and directly receive the information they need from a short system response in a conversational manner. Recently, there have been some attempts towards a similar goal, e.g., studies on Conversational Agents (CAs) and Conversational Search (CS). However, they either do not address complex information needs, or they are limited to the development of conceptual frameworks and/or laboratory-based user studies.

We pursue two goals in this paper: (1) the creation of a suitable dataset, the Search as a Conversation (SaaC) dataset, for the development of pipelines for conversations with search engines, and (2) the development of a state-of-the-art pipeline for conversations with search engines, the **C**onvers**a**tions with **S**earch **E**ngines (CaSE), using this dataset. SaaC is built based on a multi-turn conversational search dataset, where we further employ workers from a crowdsourcing platform to summarize each relevant passage into a short, conversational response. CaSE enhances the state-of-the-art by introducing a supporting token identification module and a prior-aware pointer generator, which enables us to generate more accurate responses.

We carry out experiments to show that CaSE is able to outperform strong baselines. We also conduct extensive analyses on the SaaC dataset to show where there is room for further improvement beyond CaSE. Finally, we release the SaaC dataset and the code for CaSE and all models used for comparison to facilitate future research on this topic.

CCS Concepts: • **Information systems** → **Search interfaces**; **Question answering**; *Top-k retrieval in databases.*

Additional Key Words and Phrases: Conversational modeling, Search engine, Dataset, Neural model

ACM Reference Format:
Pengjie Ren, Zhumin Chen, Zhaochun Ren, Evangelos Kanoulas, Christof Monz, and Maarten de Rijke. 2020. Conversations with Search Engines. *ACM Transactions on Information Systems* 1, 1 (April 2020), 19 pages. https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

Authors' addresses: Pengjie Ren, University of Amsterdam, Amsterdam, The Netherlands, p.ren@uva.nl; Zhumin Chen, Shandong University, Jinan, China, chenzhumin@sdu.edu.cn; Zhaochun Ren, Shandong University, Jinan, China, zhaochun.ren@sdu.edu.cn; Evangelos Kanoulas, University of Amsterdam, Amsterdam, The Netherlands, e.kanoulas@uva.nl; Christof Monz, University of Amsterdam, Amsterdam, The Netherlands, c.monz@uva.nl; Maarten de Rijke, University of Amsterdam & Ahold Delhaize, Amsterdam, The Netherlands, derijke@uva.nl.

1046-8188/2020/4-ART $15.00
https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

1 INTRODUCTION

As we surround ourselves with a range of mobile devices, e.g., smartphones, smartwatches, which only have small screens available or even no screen at all, search is increasingly performed in a conversational manner.[1] Despite this development, for complex information needs where a user's intent may be unclear or where it is not obvious what the single direct answer should be (if any), complex search engine result pages (SERPs) are still the dominant format to present results to users. SERPs are typically characterized by a diverse set of snippets, usually grouped along vertical dimensions and/or by modality, which is far from our natural mode of communication through conversations. Hence, even when we interact with search engines, the more natural mode of interaction, instead of a complex SERP is conversational in nature [4, 5, 11, 28]. Figure 1 illustrates the difference given the information need *"The Bronze Age collapse and the transition into a dark age"*. In a *traditional SERP* scenario, we would use keywords to express our information need. For each query, we issue the keywords to a search engine and receive a SERP with a ranked list of results, possibly with snippets, in return. Then, we go through the list and find the information we need from the relevant SERP. If not, we reformulate our query and this process is repeated until our information need is satisfied. Alternatively, we can fulfill our information need through *conversations with search engines*. Here, we would express our need in natural language and we would directly receive the information we need in a short system response, that is a summary of relevant results listed in the SERP, in a conversational manner.

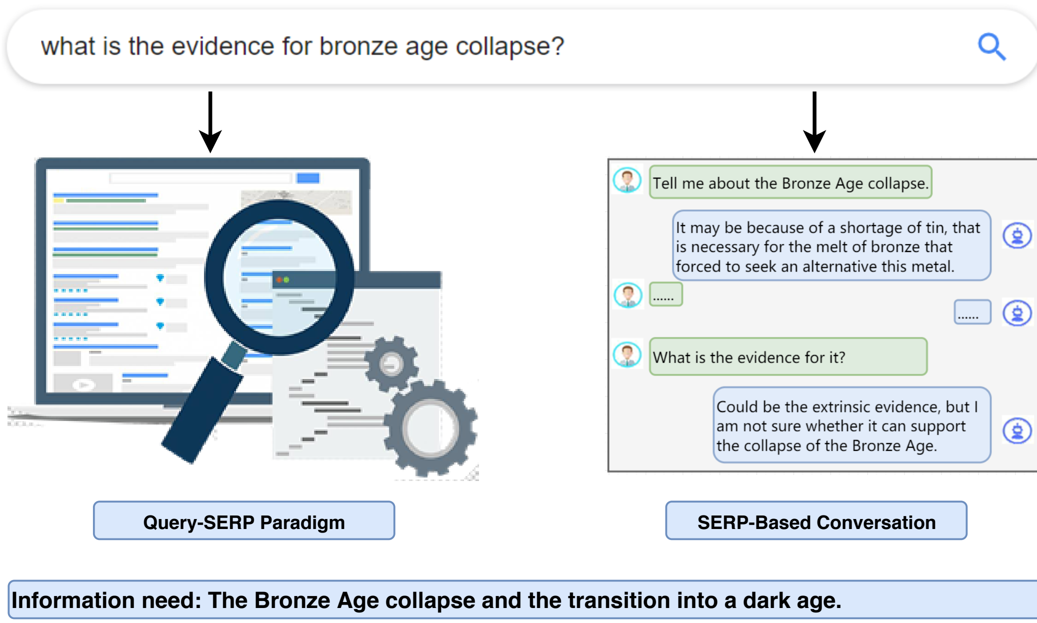

Fig. 1. Search using a traditional SERP vs. a conversation with a search engine, illustrated using the information need *The Bronze Age collapse and the transition into a dark age.*

Although conversational agents for connecting people to information have attracted a lot of attention, most studies focus on task-oriented dialogue systems (TDS) [8], question answering (QA) [43], or machine reading comprehension (MRC) [10, 29]. In these scenarios, users have well-specified and specific information needs and their queries are mostly simple questions that can be

[1] https://www.thinkwithgoogle.com/consumer-insights/personal-needs-search-trends/

answered by a relatively short text span (entity mentions, etc.) extracted from the given background knowledge (database, documents, etc.). But search engines cater for a much broader set of needs. Users' queries can be far more complex than a simple factoid question, and often cannot be answered by extracting a short span from a text snippet. Recently, there have been some studies on CS that target complex queries, however, most of them are limited to examining conceptual frameworks or laboratory-based user studies [28, 35]. Although there are some available datasets, they all have critical limitations. The CAsT dataset[2] only provides ground truth passages as answers and does not have conversational responses. The MS MARCO QA dataset [24] is single-turn and when there is no answer, it just leaves the system response blank, which is sufficient for training models but not suitable for evaluation. Hence, the conversational datasets that are available today are not sufficient to support the development of conversations with search engines.

To address this gap we pursue two goals: (1) the creation of a suitable dataset for the development of pipelines for conversations with search engines, and (2) the development of state-of-the-art baseline components that make up such a pipeline using this dataset. The dataset, called *Search as a Conversation* (SaaC), is developed in a Wizard-of-Oz fashion [6]. We simulate users based on conversational queries from the CAsT dataset. Then, we employ online workers (a.k.a., "wizards") to play the role of the system. The wizards have access to a SERPs from which they can get useful information to respond to the user queries. We ask the wizards to find supporting sentences from results/snippets on a SERPs and summarize them into short conversational responses. When there is no direct answer, we ask the wizards to generate something that is likely relevant, e.g., "It could be the extrinsic evidence, but I am not sure ...", or interesting to the user, e.g., "I have no idea about how melatonin was discovered. But I can tell you that ...".

As to our second main goal in this paper, using the SaaC dataset for the development of pipelines to support conversations with search engines, we devise a modularized multi-task learning framework, called ***C**onvers**a**tions with **S**earch **E**ngines* (CaSE). CaSE decomposes conversations with search engines into four sub-tasks: (1) conversation & passage understanding (CPU), (2) relevant passage selection (RPS), (3) supporting token identification (STI), and (4) response generation (RG). CPU is a module aiming at understanding and encoding conversations and passages. RPS then finds relevant passages based on the encoded representations from the CPU. STI further identifies supporting tokens that are eventually used in the responses. Finally, RG generates the responses based on the outputs from above three modules.

Because there are no ground truth labels for STI to define a supervised learning loss, we present a weakly-supervised Confidence-Critical Cross Entropy (CCCE) learning loss based on the intuition that the overlapping tokens between the ground truth responses and the passages are more likely supporting tokens than the non-overlapping ones, especially overlapping larger rare tokens. In order to make CaSE generate more accurate responses, we propose a Prior-aware Pointer Generator (PPG) to implement RG by considering the passage and token probabilities from RPS and STI as priors so that the generated responses are expected to be more accurate by including supporting tokens from relevant passages. We conduct experiments to: (1) compare the performance of state-of-the-art methods from related tasks to our CaSE, (2) understand the contribution of the four sub-tasks in CaSE, and (3) identify room for further improvement on SaaC beyond CaSE.

To sum up, the contributions of this work are as follows:

- We introduce the task of conversations with search engines and build a new dataset, the SaaC dataset.

[2] http://www.treccast.ai/

- We decompose the task into four sub-tasks (CPU, RPS, STI, RG), and propose a modularized CaSE model that uses a weakly-supervised CCCE loss to identify the supporting tokens and PPG to encourage generating more accurate responses.
- We conduct extensive experiments to show the effectiveness of CaSE and identify room for further improvement on conversations with search engines.

2 DATASET

We next describe the stages involved in creating the SaaC dataset.

2.1 Collecting Conversational Queries

TREC CAsT [12] has already built a collection of conversational query sequences, so we reuse their data to reduce development cost. Here, we briefly recall the process used in collecting the TREC CAsT data. The topics in CAsT are collected from a combination of previous TREC topics (Common Core [9], Session Track [18], etc.), MS MARCO Conversational Sessions, and the CAsT organizers [12]. The organizers ensured that the information needs are complex (requiring multiple rounds of elaboration), diverse (across different information categories), open-domain (not requiring expert domain knowledge to access), and mostly answerable (with sufficient coverage in the passage collection). A description of an example topic is shown in Figure 2. Then, the TREC CAsT organizers

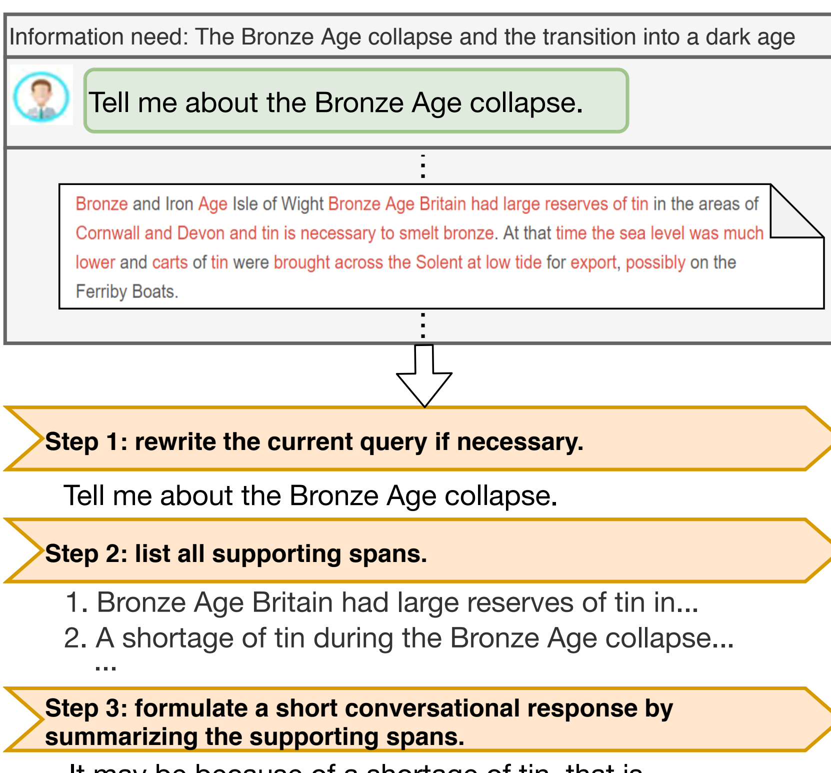

Fig. 2. The process of building the SaaC dataset.

created sequences of conversational queries for each turn. They started with the description of the topic and manually formulated the first conversational query. After that, they formulated follow-up

Table 1. Comparison of conversational datasets on search, chitchat and question answer.

Dataset	Multiple turns	Complex query	Open domain	Conversational response
CAsT [12]	✓	✓	✓	✗
Holl-e [23], WoW [13]	✓	✓	✗	✓
MS MARCO [24]	✗	✓	✓	?
QuAC [10], CoQA [30]	✓	✗	✗	✗
SaaC (this paper)	✓	✓	✓	✓

conversational queries by introducing coherent transitions, e.g., coreference and omission. For example, "Tell me about the Bronze Age collapse. … What is the evidence for it? (introducing coreference).", or "What is a physician's assistant? … What's the average starting salary in the UK? … What about in the US? (introducing omission)." There is a constraint that later conversational queries only depend on the previous queries, but not on system responses. We will discuss this constraint later. The reader is referred to Dalton et al. [12] for a more detailed account.

2.2 Collecting Candidate Passages

For our dataset creation, we combine three standard TREC collections: MARCO Ranking passages, Wikipedia (TREC CAR), and News (Washington Post) as the passage collection. To introduce more complex passages and meanwhile achieve higher recall for the current query, we follow Voskarides et al. [38] to extend the current query by extracting words that capture relevant information from previous turns and add them to the query of the current turn. Next, we use standard query likelihood with Dirichlet smoothing and RM3 relevance feedback as the ranking model to retrieve the top 10 candidate passages (if the ground truth passage is within the top 10, otherwise, we retrieve the top 9 and manually add the ground truth passage). Note that although we rewrote the current query to make it self-contained (which will be detailed in the next subsection), we did not use the rewritten queries when preparing the candidate passages in order to stay close to practical search engines. Finally, we randomly shuffle the top 10 candidate passages to eliminate position bias.

2.3 Collecting Conversational Responses

We employ online workers from Amazon Mechanical Turk (MTurk)[3] to collect conversational responses in a Wizard-of-Oz fashion, where we ask the workers to play the role of the system and write responses based on the provided passages, as shown in Figure 2. Specifically, we first rewrite the queries if necessary and we require that the rewritten queries should be self-contained (step 1). Then, the workers need to list all supporting spans from the passages that contain facts to help generate the responses (step 2). The supporting spans are kept the same as they are in the passages. Finally, we ask the workers to summarize the supporting spans into short, conversational responses. Other requirements include: (1) make sure the responses are case sensitive and grammatically correct; (2) avoid using the spans directly in the responses without summarization; (3) talk about secondary relevant information or information that could be interesting to the users, when no answer can be found in the passages. To guarantee that the collected data meets our requirements, we only employed high-quality master workers of MTurk and we manually checked each annotation ourselves.

Table 1 shows that while SaaC shares some characteristics with existing datasets, it also has its own unique characteristics, tailored to support research on conversations with search engines. out

[3]https://www.mturk.com/

Table 2. Some examples from the SaaC dataset. For comparison, we also include examples from the MS MARCO dataset.

	Queries	Responses
SaaC	Tell me about the Bronze Age collapse.	It may be because of a shortage of tin, that is necessary for the melt of bronze that forced to seek an alternative of this metal.
	What is the evidence for it?	Could be the extrinsic evidence, but I am not sure whether it can support the collapse of the Bronze Age.
	What are some of the possible causes?	One of the possible causes of the Bronze Age collapse is the invasion of the Sea Peoples.
	Who were the Sea Peoples?	The Sea Peoples are Greek mercenaries of the Pelesets (persians).
MS MARCO	causes of achalasia	An infection, heredity or an abnormality of the immune system
	why did the sumerians develop writing?	An ancient writing system developed by the Sumerians, made up of wedge-shaped markings.
	ursa energy	Ursa Energy, LLC is a Domestic Limited Liability Company, located in North Haven, Connecticut and was formed on Nov 03, 2009.

of all the datasets available at the time of writing, MS MARCO is the closest to SaaC; Table 2 list some instances from both datasets to show their differences.

2.4 The SaaC dataset

The SaaC dataset has 80 topics (with a total of 748 queries, and 7 − 12 queries per topic) from CAsT. Almost all of the queries are with complex information need, among which 359 are "what" queries, 144 are "how" queries, and 47 are "why" queries.

We report some descriptive statistics on the SaaC dataset in Table 3; for comparison we include the same information for the MS MARCO dataset. In the table, "#pairwise passage similarity" denotes

Table 3. Further statistics about the SaaC dataset. For comparison, we include the same information for the MS MARCO dataset.

	SaaC	MS MARCO
#query length	7.21	6.08
#answer length	28.19	15.90
#passage length	155.25	67.11
#pairwise passage similarity	31.14%	26.77%
#1-gram overlap	80.75%	90.93%
#2-gram overlap	47.68%	72.92%
#3-gram overlap	34.85%	63.43%
#4-gram overlap	27.85%	58.01%
#query common words ratio	70.80%	56.15%
#answer common words ratio	60.21%	53.61%

the average TF-IDF dot similarity of each pair of candidate passages. "#n-gram overlap" denotes the average n-gram overlap ratio between the answer and candidate passages. "query/answer common words ratio" denotes the ratio of common words (word frequency ≥100,000) in the query and answer, respectively. We can see from Table 3 that SaaC is more complex and conversational than MS MARCO. SaaC is more complex in that (1) the "#query/answer/passage length" is larger, which means the queries are more complex to understand, and the passages contain more noise; and (2) the "#pairwise passage similarity" is larger, which means it has more confusing candidate passages, making it hard to find the correct ones. SaaC is also more conversational because: (1) the "n-gram overlap" between the answers and passages is much smaller, which means the answers are more abstractive; and (2) the "#query/answer common words ratio" is larger, which means the queries and answers are more in spoken languages which are more informal.

3 METHOD

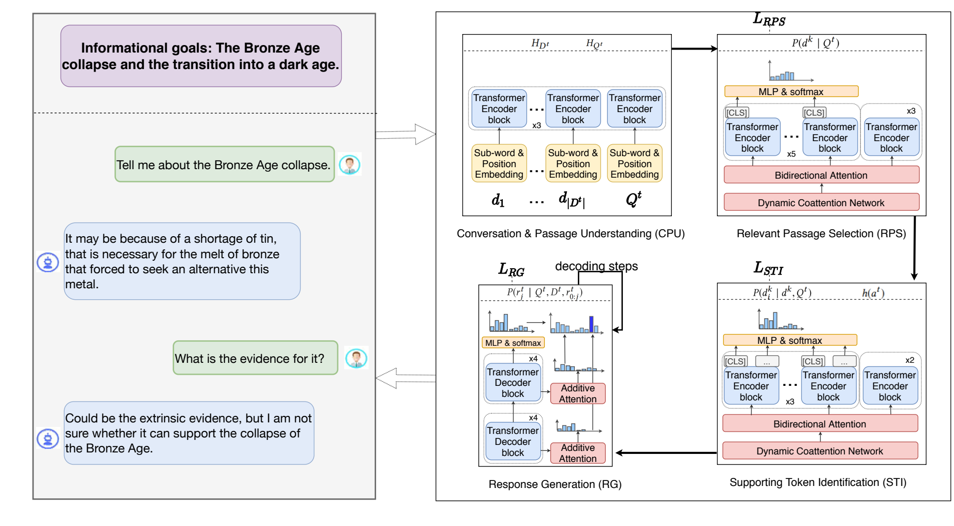

Fig. 3. An overview of **C**onvers**a**tions with **S**earch **E**ngines (CaSE). Section 3 contains a walkthrough of the model.

Formally, given a series of previous user queries $Q^{t-1} = [q^{t-m}, q^{t-m+1}, \ldots, q^{t-1}]$ in natural language, the user query at the current turn q^t, and a list of candidate passages $D^t = [d^1, d^2, \ldots, d^k]$ that have been retrieved by a search engine and potentially contain the answers (or are at least relevant to q^t),[4] the SaaC task is to generate a short response r^t for q^t by finding the relevant passages and summarizing the supporting spans from these passages into a conversational response.

In this work, we decompose the SaaC task into four sub-tasks: (1) conversation & passage understanding (CPU), (2) relevant passage selection (RPS), (3) supporting token identification (STI), and (4) response generation (RG), as shown in Figure 3. We devise a modularized framework, **C**onvers**a**tions with **S**earch **E**ngines (CaSE), to operationalize the sub-tasks in an end-to-end way.

Specifically, the CPU module first encodes each query q and passage d into a sequence of hidden representations, i.e.,

$$H_{q^t} = [\mathbf{h}_{q^t_1}, \ldots, \mathbf{h}_{q^t_{|q^t|}}] \text{ for } q^t, \text{ and} \tag{1}$$

$$H_{d^k} = [\mathbf{h}_{d^k_1}, \ldots, \mathbf{h}_{d^k_{|d^k|}}] \text{ for } d^k. \tag{2}$$

[4]While creating effective queries from multi-turn questions is a challenging task in itself, we assume that the candidate documents are provided in advance so as to simplify the experimental design and facilitate reproducibility.

Then, based on the query and passage representations, the RPS module selects the relevant passage by estimating the passage relevance probability $P(d^k \mid Q^t)$ for each passage d^k in the candidate pool D^t. After that, the STI module estimates the probability of each passage token to be a supporting token, $P(d_i^k \mid d^k, Q^t)$, where a token is supporting if it contributes to the final responses. Finally, the RG module takes the outputs from the above three modules into consideration and generates a short response, token by token. In particular, $P(d^k \mid Q^t)$ and $P(d_i^k \mid d^k, Q^t)$ are modeled as priors in the RG module.

In the following subsections, we will describe our proposed solution for each module in detail.

3.1 Conversation and Passage Understanding

We employ a Transformer model to perform conversation and passage understanding, which relies on self-attention to extract important information to represent conversations and passages. Specifically, for the conversational queries $Q^t = Q^{t-1} \cup \{q^t\}$, we first concatenate the tokens as one sequence, and then input it into a stack of Transformer encoder blocks [37] to obtain representations for each query and each token $H_{Q^t} = [H_{q^{t-m}}, H_{q^{t-m+1}}, \ldots, H_{q^t}]$, where each H_{q^t} is defined as in Eq. 1. Note that we put a special token "[CLS]" at the start, formally referred to as $\mathbf{h}_{q^t_{CLS}}$, which is considered to represent the conversations up to the current conversation turn t. Similarly, we obtain representations H_{d^k} as in Eq. 2 for each passage.

3.2 Relevant Passage Selection

In order to model the relevance to conversational queries of each passage, we first need to model the interaction between them. Here, we employ a similar bi-directional attention flow as proposed by Seo et al. [33] to do MRC, which is also used by Nishida et al. [25] to do QA. Specifically, we first obtain the interaction matrix $M^{Qd_k} \in \mathbb{R}^{|Q_t| \times |d_k| \times N}$ between the conversation tokens H_{Q^t} and each passage H_{d^k} [33] as follows:

$$M^{Qd_k} = \begin{bmatrix} f^{Qd_k}(\mathbf{h}_{q_1^{t-m}}, \mathbf{h}_{d_1^k}) & f^{Qd_k}(\mathbf{h}_{q_1^{t-m}}, \mathbf{h}_{d_2^k}) & \cdots & f^{Qd_k}(\mathbf{h}_{q_1^{t-m}}, \mathbf{h}_{d_{|d^k|}}) \\ f^{Qd_k}(\mathbf{h}_{q_2^{t-m}}, \mathbf{h}_{d_1^k}) & f^{Qd_k}(\mathbf{h}_{q_2^{t-m}}, \mathbf{h}_{d_2^k}) & \cdots & f^{Qd_k}(\mathbf{h}_{q_2^{t-m}}, \mathbf{h}_{d_{|d^k|}}) \\ \cdots & \cdots & \ddots & \cdots \\ f^{Qd_k}(\mathbf{h}_{q^t_{|q^t|}}, \mathbf{h}_{d_1^k}) & f^{Qd_k}(\mathbf{h}_{q^t_{|q^t|}}, \mathbf{h}_{d_2^k}) & \cdots & f^{Qd_k}(\mathbf{h}_{q^t_{|q^t|}}, \mathbf{h}_{d^k_{|d^k|}}) \end{bmatrix}, \tag{3}$$

where f^{Qd_k} is the cross-correlation function, which is modeled as:

$$f^{Qd_k}(\mathbf{h}_{q_i}, \mathbf{h}_{d_j^k}) = {\mathbf{v}^{Qd_k}}^\top [\mathbf{h}_{q_i} \oplus \mathbf{h}_{d_j^k} \oplus (\mathbf{h}_{q_i} \odot \mathbf{h}_{d_j^k})], \tag{4}$$

where $\mathbf{v}^{Qd_k} \in \mathbb{R}^{3N \times 1}$ is the parameter vector; $\oplus$ denotes the concatenation operation and $\odot$ denotes the Hadamard product. Then, a Dynamic Coattention Network as proposed in [40] is used to obtain the dual attention representations for the conversational queries $H_{Q^t \leftharpoondown D^t}$, and each passage $H_{d^k \leftharpoondown Q^t}$ as follows:

$$\begin{aligned} H_{Q^t \leftharpoondown D^t} &= [H_{Q^t} \oplus H^1_{Q^t \leftharpoondown D^t} \oplus H^2_{Q^t \leftharpoondown D^t} \oplus (H^1_{Q^t \leftharpoondown D^t} \odot H_{Q^t}) \oplus (H^2_{Q^t \leftharpoondown D^t} \odot H_{Q^t})] \\ H_{d^k \leftharpoondown Q^t} &= [H_{d^k} \oplus H^1_{d^k \leftharpoondown Q^t} \oplus H^2_{d^k \leftharpoondown Q^t} \oplus (H^1_{d^k \leftharpoondown Q^t} \odot H_{d^k}) \oplus (H^2_{d^k \leftharpoondown Q^t} \odot H_{d^k})], \end{aligned} \tag{5}$$

where

$$H^2_{Q^t \leftharpoondown D^t} = \max(\{M_d H^1_{d^k \leftharpoondown Q^t}\}_{k=1,\ldots,|D^t|}) \tag{6}$$

$$H^1_{Q^t \leftharpoondown D^t} = \max(\{M_{d_k} H_{d^k}\}_{k=1,\ldots,|D^t|}) \tag{7}$$

$$H^2_{d^k \leftharpoondown Q^t} = M_Q^\top H^1_{Q^t \leftharpoondown D^t}, \;\; H^1_{d^k \leftharpoondown Q^t} = M_Q^\top H_{Q^t} \tag{8}$$

$$M_{d_k} = \text{softmax}_d(M^{Qd_k}), \;\; M_Q = \text{softmax}_Q(M^{Qd_k}). \tag{9}$$

Here, max denotes max pooling; softmax_Q and softmax_d denote the softmax over M^{Qd_k} along the query and passage dimension, respectively. Then, a stack of the Transformer encoder blocks are used to reduce the dimension of $H_{Q^t \leftharpoondown D^t}$ and $H_{d^k \leftharpoondown Q^t}$.

To estimate the passage relevance score, we use an MLP to get the passage relevance score by taking the first token representation of each passage $H_{d^k_0 \leftharpoondown Q^t}$ (corresponding to the "[CLS]" token in §3.1) as input. The relevance score is normalized with a sigmoid to obtain the relevance probability $P(d^k \mid Q^t)$. We use binary cross entropy to supervise the learning of this module:

$$L_{RPS} = \sum_{d^k \in D^t} y_{d^k} \log P(d^k \mid Q^t) + (1 - y_{d^k}) \log(1 - P(d^k \mid Q^t)), \tag{10}$$

where $y_{d^k} = 1$ if d is relevant otherwise $y_{d^k} = 0$.

3.3 Supporting Token Identification

Previous methods directly generate the responses after passage selection, which we hypothesize can be improved by incorporating a dedicated supporting token identification (STI) module. The core idea is that, besides learning to select the relevant passage, the model could also learn to identify supporting tokens, which might be used to generate the response.

To do so, we use a similar architecture as in the RPS module (parameters are not shared) to get updated representations $H_{Q^t \leftharpoondown D^t}$ and $H_{D^t \leftharpoondown Q^t}$. But, instead of estimating the passage relevance score based on the "[CLS]" representation, we estimate the probability of each passage token as a supporting token $P(d_i^k \mid d^k, Q^t)$. Specifically, we use an MLP to get the supporting token likelihood score for each token with $h_{d_i^k \leftharpoondown Q^t} \in H_{D^t \leftharpoondown Q^t}$ as input, which is normalized with a sigmoid to obtain $P(d_i^k \mid d^k, Q^t)$.

Unfortunately, there are no ground truth labels to define a supervised learning loss to train $P(d_i^k \mid d^k, Q^t)$. To this end, we design a weak supervision signal based on the following intuitions: (1) If a passage token d_i^k is a supporting token, it must exist in the ground truth response; (2) If the surrounding tokens of d_i^k are also in the ground truth response, d_i^k is more likely to be a supporting token; (3) Rare passage tokens that exist in the ground truth response are more likely to be supporting tokens than frequent ones. Specifically, we devise a *Confidence-Critical Cross Entropy* (CCCE) loss as follows:

$$L_{STI} = -\sum_{d^k \in D^t} \sum_{d_i^k \in d^k} \left[c(d_i^k) \hat{y}_{d_i^k} \log P(d_i^k \mid d^k, Q^t) + (1 - \hat{y}_{d_i^k}) \log(1 - P(d_i^k \mid d^k, Q^t)) \right], \tag{11}$$

where $\hat{y}_{d_i^k}$ is a weak label indicating whether d_i^k is a supporting token; $\hat{y}_{d_i^k} = 1$ if $d_i^k \in r^{t*}$, and 0 otherwise, where r^{t*} is the ground truth response; and $c(d_i^k)$ is a coefficient indicating the confidence of d_i^k as a supporting token, which is defined as:

$$c(d_i^k) \propto \frac{1}{\log \mathit{freq}(d_i^k)} \cdot \prod_n |d^k_{i-n:i+n} \cap r^{t*}|, \tag{12}$$

where $\mathit{freq}(d_i^k)$ is the token frequency in the data collection. The first term models how "rare" d_i^k is, the second term models how likely d_i^k and its n surrounding tokens are supporting tokens (overlapped with ground truth). Finally, $c(d_i^k)$ ensures that rare and more often overlapped tokens get more opportunities to be identified as supporting tokens.

3.4 Response Generation

We propose a Prior-aware Pointer Generator (PPG) to implement the RG module, which is able to generate tokens from a predefined vocabulary and copy tokens from both the queries and passages. Especially when estimating the copy probability, PPG takes the passage relevance and supporting token likelihood into consideration.

Given the previous decoded tokens $r^t_{0:j} = [r^t_0, \ldots, r^t_{j-1}]$ (r^t_0 is set to a special token "[BOS]" indicating the beginning of decoding), we first use a stack of Transformer decoder blocks [37] to obtain the hidden representations

$$H^Q_{r^t_{0:j}} = [h^Q_{r^t_0}, \ldots, h^Q_{r^t_{j-1}}],$$

which takes $r^t_{0:j}$ and $H_{Q^t \leftharpoondown D^t}$ as inputs. Then, we use another stack of Transformer Decoder blocks to obtain hidden representations

$$H^D_{r^t_{0:j}} = [h^D_{r^t_0}, \ldots, h^D_{r^t_{j-1}}], \tag{13}$$

which takes $H^Q_{r^t_{0:j}}$ and $H_{D^t \leftharpoondown Q^t}$ as inputs. Afterwards, we estimate the token probability from three modes: generating from the vocabulary g, copying from queries c_{Q^t}, and copying from passages c_{D^t}.

Vocabulary generator. The probability of generating a token from the predefined vocabulary is estimated as:

$$P(r^t_j \mid Q^t, D^t, r^t_{0:j}, g) = P(g \mid r^t_{0:j}) \operatorname{softmax}(\operatorname{mlp}([r^t_{j-1} \oplus h^D_{r^t_{j-1}} \oplus h(a^t)])), \tag{14}$$

where $P(g \mid r^t_{0:j})$ denotes the probability of the generation mode g; $h(a^t)$ is the answer representation from the STI module, which is estimated as follows:

$$h(a^t) = \sum_{d^k \in D^t} P(d^k \mid Q^t) \sum_{d^k_i \in d^k} P(d^k_i \mid d^k, Q^t) h_{d^k_i \leftharpoondown Q^t}, \tag{15}$$

where $P(d^k \mid Q^t)$ is the passage relevance probability; $h_{d^k_i \leftharpoondown Q^t}$ is the i-th token representation from $H_{d^k \leftharpoondown Q^t}$. Both are from the RPS module. $P(d^k_i \mid d^k, Q^t)$ is the supporting token probability from the STI module.

Query pointer generator. We use another additive attention to estimate the probability of copying a token from the conversational queries:

$$P(r^t_j = Q^t_i \mid Q^t, D^t, r^t_{0:j}, c_{Q^t}) = P(c_{Q^t} \mid r^t_{0:j}) P(Q^t_i \mid Q^t, D^t, r^t_{0:j}), \tag{16}$$

where $P(c_{Q^t} \mid r^t_{0:j})$ is the query copying mode probability; and $P(Q^t_i \mid Q^t, D^t, r^t_{0:j}) = \operatorname{attention}(query : h^Q_{r^t_{j-1}}, key : H_{Q^t \leftharpoondown D^t})$.

Prior-aware passage pointer generator. The probability of copying a token from the passages is

$$P(r^t_j = d^k_i \mid Q^t, D^t, r^t_{0:j}, c_{D^t}) = P(c_{D^t} \mid r^t_{0:j}) \sum_{d^k \in D^t} P(d^k \mid Q^t) \cdot \sum_{d^k_i \in d} P(d^k_i \mid d^k, Q^t) P(d^k_i \mid Q^t, D^t, r^t_{0:j}), \tag{17}$$

where $P(c_{D^t} \mid r^t_{0:j})$ is the passage copying mode probability; $P(d^k \mid Q^t)$ is the passage prior from the RPS module and $P(d^k_i \mid d^k, Q^t)$ is the supporting token prior from the STI module; and $P(d^k_i \mid Q^t, D^t, r^t_{0:j}) = \operatorname{attention}(query : h^D_{r^t_{j-1}}, key : H_{D^t \leftharpoondown Q^t})$.

To coordinate the probabilities from different modes, we learn a mode coordination probability:

$$[P(g \mid r^t_{0:j}), P(c_{Q^t} \mid r^t_{0:j}), P(c_{D^t} \mid r^t_{0:j})] = W^\top [h_{r^t_{j-1}} \oplus h^{att}_{Q^t} \oplus h^{att}_{D^t}], \tag{18}$$

where $W \in 3N \times 3$ is the parameter matrix; $h^{att}_{Q^t}$ and $h^{att}_{D^t}$ are the attended query and passage representations from the two attentions in the query and passage pointer generators, respectively. The final probability at the j-th decoding step $P(r^t_j \mid Q^t, D^t, r^t_{0:j}) = P(r^t_j \mid Q^t, D^t, r^t_{0:j}, g) + P(r^t_j \mid Q^t, D^t, r^t_{0:j}, c_{Q^t}) + P(r^t_j \mid Q^t, D^t, r^t_{0:j}, c_{D^t})$. If a token is absent from a mode, its corresponding probability from that mode is set to zero.

We use the negative log likelihood loss to train PPG as follows:

$$L_{RG} = -\sum_{r^t_j \in r^t} \log P(r^t_j \mid Q^t, D^t, r^t_{0:j}). \tag{19}$$

4 EXPERIMENTAL SETUP

We seek to answer the following research questions:

(**RQ1**) What is the performance of CaSE compared to other methods? Does CaSE outperform the state-of-the-art methods in terms of response generation and passage ranking performance? What, if any, are the performance differences on the MS MARCO and SaaC datasets?

(**RQ2**) What is the effect of different components of CaSE on its overall performance?

(**RQ3**) Where does CaSE fail? That is, is there any room for further improvement on the SaaC dataset?

4.1 Datasets and Evaluation Metrics

As pointed out in §2.4, SaaC contains 80 topics (with 748 queries) from CAsT, which is too small to train neural models in an end-to-end fashion. Hence, we train all the models that we consider on the MS MARCO 2.1 Q&A + Natural Language Generation training set (MS MARCO train) [24]. Although this is a single-turn dataset and the queries and responses are less conversational than in the SaaC dataset (see Table 2), it does have passage relevance labels and human written answers, which is sufficient as a training set. MS MARCO is the sole dataset with real queries from search engines and human written answers. We only keep data samples where the "wellFormedAnswers" field is not empty. This ensures that these well-formed answers are true natural language answers and not just span selections.[5] We randomly split the original development set into two sets with roughly equal size, one as our development set (MS MARCO dev) and the other as our test set (MS MARCO test). The sizes of the MS MARCO train, MS MARCO dev, MS MARCO test, and SaaC test are 524,105, 32,345, 32,225, and 1,008, respectively.

We use BLEU (up to 4-grams using uniform weights), and ROUGE-1, ROUGE-2, and ROUGE-L to evaluate the response generation performance, which are commonly used in natural language generation tasks, e.g., QA, MRC [19, 25]. We also report MAP, Recall@5 and NDCG for passage ranking performance.

4.2 Methods Used for Comparison

We collect and implement state-of-the-art methods from various related tasks.

- **S2SA**. S2SA is a sequence-to-sequence with attention model; it is commonly used as a baseline for natural language generation tasks [22, 23].

[5] https://github.com/microsoft/MSMARCO-Question-Answering

- **GTTP** [32]. GTTP improves S2SA by incorporating a pointer mechanism that enables it to copy tokens from the input during generation. GTTP achieves state-of-the-art performance on many natural language generation tasks [26, 30].
- **TMemNet** [13]. TMemNet was first introduced for the knowledge grounded dialogue task. It combines a Memory Network and Transformer to do knowledge selection and dialogue response generation.
- **GLKS** [31]. GLKS is a state-of-the-art method for Background-Based Conversations (BBC) (the best performing method on the Holl-E[6] dataset at the time of writing). It introduces a mechanism to combine global and local knowledge selection for dialogue response generation.
- **Masque** [25]. Masque is the best performing method on the MS MARCO Q&A + Natural Language Generation task at the time of writing. Because we only use "wellFormedAnswers" to train the models, we remove the answer possibility classifier and style token.
- **CaSE**. CaSE is proposed in this paper.

4.3 Implementation Details

For a fair comparison, we implement all models in our experiments based on the same code framework to ensure that they share the same codes apart from the model itself. We set the word embedding size and hidden size to 256. We use the BERT vocabulary[7] for all methods but we avoid using any extra resources for all methods, including pre-trained embeddings. The vocabulary size is 30,522. The learning rate was increased linearly from zero to 2.5×10^{-4} in the first 6,000 steps and then annealed to 0 by using a cosine schedule. We use gradient clipping with a maximum gradient norm of 1. We use the Adam optimizer ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$). An exponential moving average was applied to all trainable variables with a decay rate of 0.9995. For all models, we combine multiple losses linearly if there is more than one. We train all models on four TITAN X (Pascal) GPUs. The batch size is chosen from (32, 64, 128) according to the GPU memory. We select the best models based on performance on the development set.

5 EXPERIMENTAL RESULTS

5.1 How Does CaSE Perform?

To answer RQ1, we report the results of all methods on both MS MARCO[8] and SaaC; see Table 4. S2SA, GTTP and GLKS do not perform passage ranking, so there are no MAP, Recall@5 and NDCG results. Given the results, there are three main observations.

Table 4. Overall performance (%). **Bold face** indicates the best result in terms of the corresponding metric. R-1: ROUGE-1; R-2: ROUGE-2; R-L: ROUGE-L. * and ** indicate CaSE is significantly better than Masque (p-value < 0.05 and p-value < 0.01 with t-test, respectively).

Methods	MS MARCO test							SaaC test						
	R-1	R-2	R-L	BLEU	MAP	Recall@5	NDCG	R-1	R-2	R-L	BLEU	MAP	Recall@5	NDCG
S2SA	46.75	25.17	38.79	18.61	–	–	–	24.71	09.04	17.69	07.23	–	–	–
GTTP	47.41	25.74	39.64	19.32	–	–	–	28.89	10.51	20.61	08.80	–	–	–
GLKS	47.74	27.51	40.01	20.75	–	–	–	35.46	13.46	25.31	12.38	–	–	–
TMemNet	48.01	29.56	40.29	23.23	53.43	82.56	64.83	35.29	12.72	25.05	09.60	14.12	11.31	29.25
Masque	55.33	37.15	47.75	30.01	63.75	90.77	72.82	35.44	13.48	25.74	11.97	16.13	12.00	31.77
CaSE	**57.44****	**39.26****	**49.91****	**32.08****	**65.75****	**92.20****	**74.34****	**37.34***	**14.58**	**27.24***	**13.26**	**17.22**	**13.72**	**32.67**

[6]https://github.com/nikitacs16/Holl-E
[7]https://github.com/huggingface/transformers
[8]The performance is not comparable to the leaderboard of MS MARCO at http://www.msmarco.org/leaders.aspx as different training, test sets and evaluation scripts are used.

First, CaSE achieves the best results on both datasets in terms of all metrics. Especially, CaSE can outperform the best performing model Masque on the MS MARCO leaderboard and the best model GLKS from the BBC task (at the time of writing). Generally, CaSE improves over Masque by around 2%pt in terms of generation evaluation metrics and around 1–2%pt in terms of passage ranking metrics. Part of the improvement is from the proposed STI and PPG modules, which we will analyze in more detail in §5.2.

Second, the results on the MS MARCO dataset are much higher than those on the SaaC dataset. This is the case for all methods, including CaSE. For example, the BLEU score of CaSE is 18.82%pt higher on MS MARCO for response generation, and the MAP score is 48.53%pt higher. This confirms that SaaC is a more challenging and more suitable dataset than MS MARCO for research on conversations with search engines. SaaC is more challenging because (1) the queries and passages are more complex, which is clear from Table 3. (2) SaaC has greater query, passage and answer lengths and the passages are more similar. SaaC is more suitable for conversations with search engines because (1) it has multi-turn conversations, which introduces the requirements of modeling context from historical turns. (2) the responses are more abstractive and conversational, which is closer to real conversation scenarios. See §5.3 for further details.

Third, modeling passage selection is necessary. On the one hand, we can see that the methods with passage selection modules (TMemNet, Masque and CaSE) are generally much better than those without (i.e., S2SA, GTTP and GLKS). On the other hand, we also notice that the improvements for passage ranking are consistent with the improvements for response generation.

5.2 What Do the Components of CaSE Contribute?

To answer RQ2 and analyze the effects of the RPS, STI and RG modules in CaSE, we conduct an ablation study. We do not separately study the modeling and learning of the STI module, as removing the L_{STI} will result in no direct supervision to guide the learning of STI, which does not make sense. The results are shown in Table 5.

Table 5. Ablation study (%). CaSE-X denotes CaSE with the component X left out. CaSE-RG: replacing the RG component with a traditional pointer generator. CaSE-STI: CaSE without the STI component. CaSE-PS: CaSE without the RPS module. There is no CaSE-CPU because all other components rely on the CPU component.

CaSE variants	MS MARCO test							SaaC test						
	R-1	R-2	R-L	BLEU	MAP	Recall@5	NDCG	R-1	R-2	R-L	BLEU	MAP	Recall@5	NDCG
CaSE	**57.44**	**39.26**	**49.91**	**32.08**	**65.75**	**92.20**	**74.34**	37.34	14.58	27.24	13.26	**17.22**	**13.72**	**32.67**
CaSE-RG	57.21	38.91	49.62	31.58	65.16	91.95	73.90	**38.32**	**15.20**	**28.21**	**14.45**	16.54	12.47	32.51
CaSE-STI	55.68	36.94	47.92	29.98	64.02	91.10	73.03	37.74	14.51	27.62	13.01	16.70	12.68	32.39
CaSE-RPS	56.36	38.37	48.75	31.22	–	–	–	37.83	15.02	26.87	14.39	–	–	–

Generally, removing any module will result in a drop in performance in terms of both response generation and relevant passage selection. Specifically, the results drop by more than 2%pt in terms of BLEU and more than 1.5%pt in terms of MAP on the MS MARCO dataset by removing the STI module. This is also true for the RPS and RG modules, although the drops are not as large as for STI. The generation performance of CaSE-STI is even worse than Masque, which confirms the effectiveness of STI. Interestingly, although we found that modeling passage selection is very important (Table 4) for the other models, removing the RPS module does not much influence the overall performance of CaSE. We think the reason is that CaSE incorporates the STI module which has some overlapping effects with RPS to some extent, as when a passage contains more supporting tokens, it is more relevant in general. To sum up, even if the RPS, STI and RG share some common effects, they are also complementary to each other as combining them will bring further improvements.

Table 6. Response generation and passage ranking performance of CaSE on the SaaC dataset w.r.t different conversational turns (%).

Turn	R-1	R-2	R-L	BLEU	MAP	Recall@5	NDCG
1	**47.97**	**28.13**	**38.22**	**33.16**	**0.2117**	**0.1909**	**0.3521**
2	38.05	16.27	27.00	15.56	0.1607	0.1043	0.3205
3	42.11	17.25	30.54	13.95	0.1904	0.1289	**0.3541**
4	37.84	13.61	26.08	13.45	0.1635	0.1234	0.3127
5	36.28	11.80	26.74	10.39	0.1406	0.0944	0.2992
6	31.73	08.33	21.92	04.53	0.1361	0.0930	0.2950
7	30.42	08.32	21.95	05.04	0.1451	0.1120	0.3000
8	33.05	11.03	24.39	08.46	0.1060	0.0936	0.2415
$\geq$ 9	35.16	13.88	26.06	11.00	0.1672	0.1088	0.3445

One exception is that CaSE-RG achieves better response generation performance than CaSE on SaaC. We believe that the reason for this behavior is that although CaSE can generate more accurate responses by leveraging the outputs from the RPS and STI modules as priors in PPG (which can be verified by the better performance on MS MARCO), this will influence the abstractiveness of the response, because CaSE is encouraged to put more emphasis on the tokens in the relevant passages with PPG.

5.3 Is there Room for further Improvement?

To answer RQ3, we explore the room for further performance improvements on the SaaC dataset by conducting additional experiments and/or case studies.

First, more effort is needed to properly model the context, i.e., the conversational history, of the current turn query. To illustrate this, we show the performance of the response generation and passage ranking of each turn in Table 6. We see that the performance is much higher in the first turn because there is no context needed to be considered. Performance drops dramatically for the following turns including the 2nd turn, and performance tends to get worse as the number of turns increases. There is an exception for the ≥9th turns, which are better than the 8th turn in terms of passage ranking. This may be because that hard queries do not go after the 8th turn in CAsT. We analyzed the queries from the ≥9th turns and found that there are a lot of "what" queries in these two turns like "What type does chemical energy belong to?", which generally needs less modeling of missing context. A deeper understanding of the current query is challenging because it is not just a matter of coreference resolution [1]. It is common that people omit information to keep the conversations natural, which is well reflected in the SaaC dataset.

Second, more effort is needed to obtain a better understanding of the current turn query. To illustrate this, we conduct a comparison of using the original current query (OQ), the context queries and the current original query (CQ+OQ), and the reformulated current query (RQ), as shown in Table 7. RQ is a manually rewritten version of the current original query, where we make sure that RQ is self-contained and does not need to rely on context. We can see that using RQ improves the performance, which is not surprising. But we should also note that even when using RQ, the performance is far from perfect as it does not even outperform the 1st turn in Table 6. This indicates that the model cannot create a understanding of the current query even though all needed information is provided. This is confirmed by the fact that CQ+OQ performs worse than OQ: the model is doing worse by involving more contextual information.

Third, more effort is needed to perform better at passage selection and support identification. We illustrate this by showing the results of CaSE when the ground truth passages (GP) or sentences (GS)

Table 7. Response generation performance of CaSE on the SaaC dataset (%). OQ: Using the original query of the current turn; CQ+OQ: Using context queries + original query of the current turn; RQ: Using reformulated ground truth query of the current turn; GP: Using ground truth passages; GS: Using ground truth supporting sentences.

	R-1	R-2	R-L	BLEU
OQ	39.24	18.71	29.49	18.05
CQ+OQ	37.34	14.58	27.24	13.26
RQ	42.79	22.17	32.75	22.05
GP	46.19	25.78	33.37	27.05
GS	48.74	29.39	36.54	32.34

are provided in Table 7. We can see that although CaSE has used effective and complex mechanisms to perform relevant passage selection and supporting token identification, there is still a lot of room to improve in this direction.

Fourth, more effort is needed to investigate how to generate more conversational and abstractive responses. Due to limitations of the MS MARCO data, the models are rarely trained to learn to generate tokens that address the conversational nature of responses. For instance, given the query "What is arnica used for?," the current model will just list the answers, "arnica , trauma , pain and shock." The human response is like "Arnica is a plant based remedy used to relief pain. It is also used to speed injury and trauma healing as well as to reduce bruising." Besides, there are some cases where there is no answer or only a partial answer is available in the passages, the current model will either generate a wrong answer or just leave it blank. However, in practical scenarios, the system should indicate it does not know the answer or only knows part of the answer and reply accordingly, e.g., "Sorry, I don't know much about the largest tiger shark ever to have lived on Earth or caught. But I do know the largest great white shark ever caught on camera, it was a seven metre-long female, called Deep Blue." We can build suitable datasets to address this. Alternatively, we can investigate how to leverage datasets from related tasks, e.g., chitchat datasets, where there are more natural and conversational human responses [13, 23].

6 RELATED WORK

We briefly present an overview of related work on Conversational Search (CS) and on CAs.

6.1 Conversational Search

The concept of search as a conversation has been around since the 1980s [4, 11]. Until recently, the idea did not attract a lot of attention due to limitations in data and computing resources at the time.

Now, the topic is back in the spotlight. One branch of work conducts user studies on CS. Vtyurina et al. [39] conduct a user study, where they ask 21 participants to solve 3 information seeking tasks by conversing with three agents: an existing commercial system, a human expert, and a perceived experimental automatic system, backed by a human "wizard behind the curtain". They show that existing conversational assistants cannot be effectively used for complex information search tasks. Vakulenko et al. [36] argue that existing studies neglect exploratory search when users are unfamiliar with the domain of their goal. They investigate interactive storytelling as a tool to enable exploratory search within the framework of a conversational interface. Trippas et al. [35] conduct a laboratory-based observational study for CS, where pairs of people perform search tasks communicating verbally. They conclude that CS is more complex and interactive than traditional search.

Another line of work has proposed theoretical frameworks concerning CS. Radlinski and Craswell [28] present a theoretical framework of information interaction in a chat setting for CS, which highlights the need for multi-turn interactions. Azzopardi et al. [3] propose a conceptual framework that outlines the actions and intents of users and agents in order to enable the user to explore the search space and resolve their information need. The work listed above studies CS either in a theoretical or a user study environment. The theoretical/conceptual frameworks have made requirements about the data annotations more demanding, often going beyond currently available datasets.

Zhang et al. [45] devise a System Ask-User Respond paradigm for CS, and design a memory network for product search and recommendation in e-commerce. Following this line, Aliannejadi et al. [2] and Zamani et al. [44] formulate the task of asking clarifying questions in information retrieval. Bi et al. [7] propose a conversational paradigm for product search, and an aspect-value likelihood model to incorporate both positive and negative feedback on non-relevant items. To advance research on CS, Dalton et al. [12] organize a TREC Conversational Assistance Track (CAsT), which establishes a concrete and standard collection of data with information needs to make systems directly comparable. In the first year, they only focus on candidate passage retrieval: Read the dialogue context and perform retrieval over a large collection of passages. Although the studies listed above propose concrete datasets or methods for CS, none of them targets directly generating conversational system responses by modeling CS in a conversational scenario like we do.

6.2 Conversational Agents

Conversational modeling has long been a hot research topic in natural language processing [16, 20]. Most research falls into three groups: task-oriented dialogue systems (TDS) agents, social bots, and QA agents [14]. TDS aims to achieve a specific task for users, e.g., booking a flight [8], while social bots aim to satisfy the human need for communication and so on [13, 22]. These goals are quite different from people's goals in search scenarios where user information needs can be more complex and exploratory.

Efforts to build QA agents come in two main flavors: KB-QA and text-QA, which study how to query a KB interactively with natural language, and generate an answer to an input query based on a set of passages, respectively. Early studies focus on extraction-based methods [15, 29, 42], which try to retrieve an entity from a KB, or extract a span from a given passages as direct answers. Later, generation-based methods, which can generate natural language answers, attract more attention [34, 43]. Recently, QA has been extended to multi-turn conversational scenarios [10], which introduce more challenges related to conversational understanding. However, all aforementioned studies mostly target simple (factoid) questions which are relatively easy to answer [17]. On many benchmark datasets, the best models are approaching human performance [10] or even have outperform humans [30]. Some approaches target complex questions [19, 27]. In particular, Nguyen et al. [24] collect a large-scale dataset, MS MARCO, from Bing usage logs, where the answers are written by real humans to ensure that they are in natural language. However, the work listed above focuses on single-turn QA, which has not been extended to multi-turn conversational scenarios where query understanding, relevant passage finding and response generation, etc. are more challenging.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose *conversations with search engines* as task for the community to consider and we contribute two types of result: First, we release a new test set, SaaC, which is more suitable and challenging for this research than existing resources. Second, we propose an end-to-end neural model, CaSE, to advance the state-of-the-art. We implement state-of-the-art methods from related tasks and conduct extensive experiments to show that: (1) our CaSE can achieve leading performance; (2) the proposed STI and PPG modules can bring large improvement; (3) SaaC is more challenging and there is significant room for further improvements.

As to future work, on the one hand, we plan to further improve the performance of CaSE by proposing transfer learning methods in order to leverage more multi-turn conversational datasets (e.g., conversational QA [30, 41], conversational MRC [10] and chitchat [21]). On the other hand, we plan to study how to address the cases in SaaC when there is no correct answer or only a partially correct answer. We will also develop ways to lift the restriction of SaaC that later conversational queries only depend on previous queries and not on system responses by introducing mixed initiatives [44].

CODE AND DATA

The SaaC dataset and the code of all methods used for comparison in this paper are shared at https://github.com/PengjieRen/CaSE-1.0.

REFERENCES

[1] Mohammad Aliannejadi, Manajit Chakraborty, Esteban Andrés Ríssola, and Fabio Crestani. 2020. Harnessing Evolution of Multi-Turn Conversations for Effective Answer Retrieval. In *Proceedings of 2020 Conference on Human Information Interaction and Retrieval (CHIIR'20)*.

[2] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'19)*. 475–484.

[3] Leif Azzopardi, Mateusz Dubiel, Martin Halvey, and Jeffery Dalton. 2018. Conceptualizing Agent-human Interactions during the Conversational Search Process. In *The 2nd Workshop on Conversational Approaches to Information Retrieval*.

[4] Nicholas J Belkin. 1980. Anomalous States of Knowledge as A Basis for Information Retrieval. *Canadian Journal of Information Science* 5, 1 (1980), 133–143.

[5] Nicholas J Belkin, Colleen Cool, Adelheit Stein, and Ulrich Thiel. 1995. Cases, Scripts, and Information-seeking Strategies: On the Design of Interactive Information Retrieval Systems. *Expert Systems with Applications* 3 (1995), 379–395.

[6] Niels O Bernsen, Hans Dybkjær, and Laila Dybkjær. 2012. *Designing Interactive Speech Systems: From First Ideas to User Testing*. Science & Business Media.

[7] Keping Bi, Qingyao Ai, Yongfeng Zhang, and W. Bruce Croft. 2019. Conversational Product Search Based on Negative Feedback. In *Proceedings of the 28th ACM International Conference on Information & Knowledge Management (CIKM'19)*. 359–368.

[8] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 5016–5026.

[9] Ben Carterette, Virgil Pavlu, Hui Fang, and Evangelos Kanoulas. 2009. Million Query Track 2009 Overview. In *TREC 2009*.

[10] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. Association for Computational Linguistics, 2174–2184.

[11] W. Bruce Croft and R. H. Thompson. 1987. I3R: A New Approach to The Design of Document Retrieval Systems. *Journal of the Association for Information Science and Technology* 38, 6 (1987), 389–404.

[12] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2019. CAsT 2019: The Conversational Assistance Track Overview. In *TREC 2019*.

[13] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered Conversational Agents. In *The International Conference on Learning Representations (ICLR'19)*.

[14] Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural Approaches to Conversational AI. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*. 2–7.

[15] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Proceedings of the 29th International Conference on Neural Information Processing Systems (NIPS'15)*. 1693–1701.

[16] Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. 2014. Towards An Open-domain Conversational System Fully Based on Natural Language Processing. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'14)*. 928–939.

[17] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*. 1601–1611.

[18] Evangelos Kanoulas, Carterette Ben, Mark Hall, Paul Clough, and Mark Sanderson. 2011. Overview of the TREC 2011 Session Track. In *TREC 2011*.

[19] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA Reading Comprehension Challenge. *Transactions of the Association for Computational Linguistics* 6 (2018), 317–328.

[20] James Lester, Karl Branting, and Bradford Mott. 2004. Conversational Agents. *The Practical Handbook of Internet Computing* (2004), 220–240.

[21] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL'15)*. 285–294.

[22] Chuan Meng, Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. RefNet: A Reference-aware Network for Background based Conversation. In *The 34th AAAI Conference on Artificial Intelligence (AAAI'20)*.

[23] Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards Exploiting Background Knowledge for Building Conversation Systems. In *The 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 2322–2332.

[24] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

[25] Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019. Multi-style Generative Reading Comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*. 2273–2284.

[26] Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *International Conference on Learning Representations (ICLR'18)*.

[27] Chen Qu, Liu Yang, W. Bruce Croft, Johanne R. Trippas, Yongfeng Zhang, and Minghui Qiu. 2018. Analyzing and Characterizing User Intent in Information-seeking Conversations. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'18)*. ACM, 989–992.

[28] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval (CHIIR'17)*. 117–126.

[29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*. 2383–2392.

[30] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* 7 (2019), 249–266.

[31] Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. Thinking Globally, Acting Locally: Distantly Supervised Global-to-local Knowledge Selection for Background based Conversation. In *The 34th AAAI Conference on Artificial Intelligence (AAAI'20)*.

[32] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*. 1073–1083.

[33] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. In *International Conference on Learning Representations (ICLR'17)*.

[34] Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2018. S-Net: From Answer Extraction to Answer Synthesis for Machine Reading Comprehension. In *The 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*. 5940–5947.

[35] Johanne R. Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the Design of Spoken Conversational Search: Perspective Paper. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval (CHIIR'18)*. 32–41.

[36] Svitlana Vakulenko, Ilya Markov, and Maarten de Rijke. 2017. Conversational Exploratory Search via Interactive Storytelling. In *1st International Workshop on Search-Oriented Conversational AI*.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 6000–6010.

[38] Nikos Voskarides, Dan Li, Andreas Panteli, and Pengjie Ren. 2019. ILPS at TREC 2019 Conversational Assistant Track. In *TREC 2019*.

[39] Alexandra Vtyurina, Denis Savenkov, Eugene Agichtein, and Charles L. A. Clarke. 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)*. 2187–2193.

[40] Caiming Xiong, Victor Zhong, and Richard Sochern. 2017. Dynamic Coattention Networks For Question Answering. In *International Conference on Learning Representations (ICLR'17)*.

[41] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*. 2369–2380.

[42] Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The Value of Semantic Parse Labeling for Knowledge Base Question Answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16)*. 201–206.

[43] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural Generative Question Answering. In *Proceedings of the Workshop on Human-Computer Question Answering*.

[44] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating Clarifying Questions for Information Retrieval. In *Proceedings of the 29th International Conference on World Wide Web (WWW'20)*.

[45] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *Proceedings of the 27th ACM International Conference on Information & Knowledge Management (CIKM'18)*. 177–186.