SCSS-LIE: A Novel Synchronous Collaborative Search System with a Live Interactive Engine

Junyan Wang[1], Peng Zhang[1*], Cheng Zhang[1], Dawei Song[2]
[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
1079158605@qq.com,pzhang@tju.edu.cn,zccode@gmail.com,dwsong@bit.edu.cn

ABSTRACT

Synchronous collaborative search systems (SCSS) refer to systems which support two or more users with similar information need to search together simultaneously. Generally, SCSS provide a social engine to enable users to communicate. However, when the number of users in the social engine is insufficient to collaborate on the search task, the social engine will encounter the cold start problem and can not perform collaborative search well. In this paper, we present a novel Synchronous Collaborative Search System with a Live Interactive Engine (SCSS-LIE). SCSS-LIE proposes to apply a ring topology to add an intelligent auxiliary robot, **Infobot**, into the social engine to support real-time interaction between users and the search engine to address the cold start problem of the social engine. The reading comprehension model BiDAF (Bi-Directional Attention Flow) is employed in the Infobot in the process of interacting with the search engine to obtain answers to facilitate the acquisition of information. SCSS-LIE can not only allow users with similar information need to be grouped into one chat channel to communicate, but also enable them to conduct real-time interaction with the search engine to improve search efficiency.

CCS CONCEPTS

• **Information systems → Collaborative search**;

KEYWORDS

Synchronous Collaborative Search Systems; Social Engine; Infobot; Interactive Engine; Machine Reading Comprehension

ACM Reference Format:
Junyan Wang[1], Peng Zhang[1*], Cheng Zhang[1], Dawei Song[2]. 2019. SCSS-LIE: A Novel Synchronous Collaborative Search System with a Live Interactive Engine. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331407

*Corresponding author: Peng Zhang (pzhang@tju.edu.cn).

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331407

1 INTRODUCTION

Web search is generally considered as a solitary activity; browsers and search systems are designed to support single-user scenarios. Whereas, studies of search strategies in knowledge workers [6] point out that users often desire to collaborate on search tasks.

Collaborative search systems exploit collective intelligence to enable users to combine their efforts in Information Retrieval (IR) activities and share information resources collaboratively. Most of the work at present in collaborative search systems refers to systems which support an asynchronous, passive, implicit form of collaboration between remote users. These systems are referred to as Asynchronous Collaborative Search Systems (ACSS) in this paper. However, users in these systems often do not know that they are collaborating or do not know who to collaborate with. In essence, the purpose of these systems is equivalent to the traditional information retrieval systems to provide individuals with effective retrieval information.

Another collaborative search method is to perform collaborative search synchronously, that is, two or more users come together in a focused group to perform real-time collaborative search to satisfy a shared information need. These systems are referred to as Synchronous Collaborative Search Systems (SCSS) [2]. Recent attempts on collaborative search systems are CoZapce [5] and SearchX [7]. However, most of these systems leverage collaborative search between small groups of users such as specific people, colleagues and friends rather than for a general-purpose web search. SECC [1] adds a social engine to allow users to communicate through multiple chat channels. However, when there are not enough people searching for similar queries, the number of people in the chat channel is insufficient to conduct collaborative search, which results in the cold start problem of the social engine.

To address the cold start problem of the social engine, we propose to employ a ring topology to add an intelligent auxiliary robot Infobot into the social engine to support multiple rounds of interaction between the social engine and the search engine. When there is only one user in the chat channel, the user can still accomplish the search task by interacting with the search engine. Furthermore, the BiDAF model [8] is implemented in the Infobot to get the answers to the users' questions to obtain information quickly.

2 MOTIVATION

In this section, we introduce our motivation of implementing the ring topology with the Infobot from two perspectives: solving the cold start problem of SCSS, leveraging the advantages of QA systems to obtain information quickly. We first illustrate the overall differences of ACSS, SCSS and SCSS-LIE. Then we present the improvement of our demo based on the two perspectives.

Table 1: Comparison between ACSS, SCSS and SCSS-LIE in terms of topology and the manner of collaborative search.

Types	Topology	Collaborative Search Manner
ACSS	single-center	asynchronous, passive
SCSS	multi-center	synchronous, positive, independent
SCSS-LIE	ring	synchronous, positive, interactive

2.1 Solving the cold start problem of SCSS

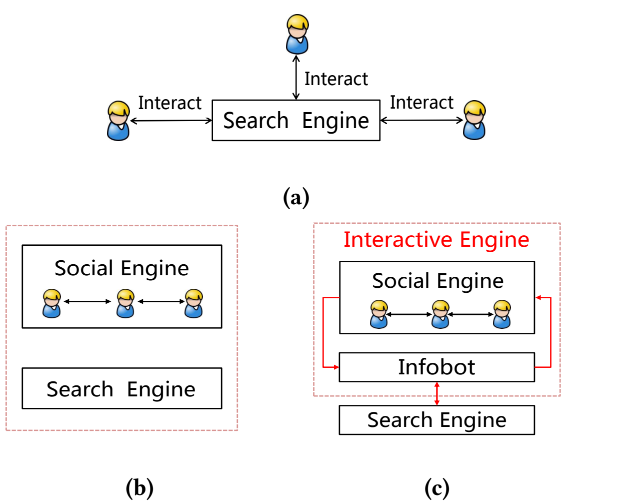


Figure 1: The topology of ACSS, SCSS and SCSS-LIE. (a) Single-center topology. (b) Multi-center topology. (c) Ring topology. Users in the social engine put forward a search query. The Infobot receives the query, retrieves it in the search engine and gets the answer. Then it returns the answer to the social engine, forming the ring topology.

Table 1 shows the overall differences between ACSS, SCSS and SCSS-LIE. ACSS share a single-center topology, which supports an asynchronous, passive form of collaborative search between multiple users. Figure 1 (a) shows the single-center topology of ACSS.

SCSS improve the search activities from an individual to a group process by providing a social engine to enable multiple users to communicate to perform synchronous, positive collaborative search. Figure 1 (b) shows this kind of multi-center topology. However, when there are not enough people searching for similar queries, or when there are some deviations in the grouping of users, the social engine will face the cold start problem.

To address the issue, we propose to move from the multi-center topology to a ring topology. Figure 1 (c) shows the ring topology of SCSS-LIE. We add an intelligent robot Infobot into the social engine as the proxy of the search engine to support multiple rounds of interaction between the social engine and the search engine. Users can directly complete search tasks in this interactive engine, which increases user experience and search efficiency.

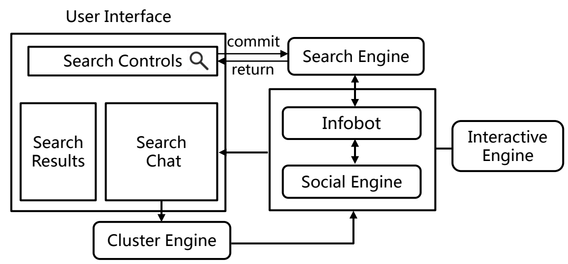


Figure 2: Framework of SCSS-LIE. The left side of the figure is the user interface, and the right side is the main engines of SCSS-LIE. On the left side of the user interface is the search results retrieved by the search engine. On the right side of the interface is the chat interface of the interactive engine.

2.2 Leveraging the advantages of QA systems to obtain information quickly

QA (Question Answering) systems provide a user-friendly interface to navigate through knowledge and information, which makes it easy for users to obtain information without having to read the entire retrieved document. The Machine reading comprehension (MRC) tasks require the machine to answer a question by reading and comprehending the question and related documents. To better utilize the advantages of the QA system to access information quickly, we plan to employ the method of MRC in the Infobot to get answer directly. Traditional MRC tasks share a single-turn setting of answering a single question without considering the context. However, in our system, users usually seek answers via conversation. Thus, traditional MRC models are not suitable to be directly applied in our system. Inspired by SDNet [9], we exploit the idea of Conversational Question Answering (CQA) in the Infobot to comprehend the related documents and conversation history and get answers to the search query.

The task of CQA can be illustrated as follows: given a passage C, and conversation history utterances $Q_1, A_1, Q_2, A_2, ..., Q_{k-1}, A_{k-1}$, the task is to generate response A_k given the latest question Q_k. To incorporate conversation history into response generation, we prepend the latest N rounds of utterances in the conversation history to the current question Q_k. So the reformulate question is $Q_k = \{Q_{k-N}; A_{k-N}; ..., Q_{k-1}; A_{k-1}; Q_k\}$. The updated question and related documents retrieved by the Infobot are then put into the traditional MRC model to obtain answers. In this paper, we employ the BiDAF model as the MRC model.

3 FRAMEWORK AND IMPLEMENTATION

In this section, we demonstrate the framework of SCSS-LIE and its implementation details.

3.1 Framework

As is shown in Figure 2, our system mainly consists of three engines, i.e., the search engine, the cluster engine and the interactive engine. The search engine provides search functions, e.g., query completion, query suggestion and webpage ranking based on an

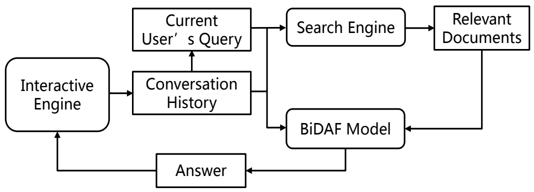


Figure 3: Working process of the Infobot module.

indexed corpus. The cluster engine groups multiple users dynamically according to the keywords of the queries they search. It compares the keywords of the current user's query with the keywords of the existing groups in the social engine. When most of the keywords of the query are the same as one of the groups, the user is divided into the group. When users in the group change search queries, the cluster engine can automatically collect users' new queries and regroup them according to the new queries. The interactive engine consists of two modules, i.e., the social engine and the Infobot. The social engine provides multiple chat channels for grouped users to communicate to exploit collaborative search. The Infobot is added in our demo to facilitate the functions of interaction with the search engine.

Figure 3 shows the working process of the Infobot module. The Infobot first collects conversation history of users in the same chat channel. Then it finds out the current user's query based on the order of utterances in the conversation history and some symbols such as '?'. Thirdly, it searches the query in the search engine and returns relevant documents. Then the current query is fused with the conversation history. The fused query and related documents are then put into the BiDAF model to get the corresponding answer and return it to the interactive engine. This process is referred to as a single-turn QA interaction between the interactive engine and the search engine. The Infobot then performs multi-turn QA interaction to accomplish the search task. Users in the interactive engine can input the symbol word 'info' to start the Infobot. Moreover, we design a switch for the Infobot to allow user to control the startup state of the Infobot. The Infobot works when the number of 'info' that the user inputs is odd, and the Infobot does not work when the number of 'info' is even.

When a user inputs a query in the interactive engine, the cluster engine will automatically run in background to dynamically group users with similar queries into the same group. Each group will provide a chat channel to enable users in the group to communicate with each other. When one user in the group inputs 'info', the Infobot will work as illustrated above.

3.2 Implementation

Figure 4 shows the main modules and technical dependencies in the system. Our system consists of five modules, including user interface, web server, search engine, cluster engine and interactive engine.

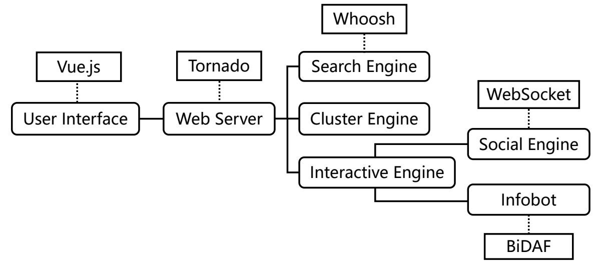


Figure 4: Modules and related technical dependencies.

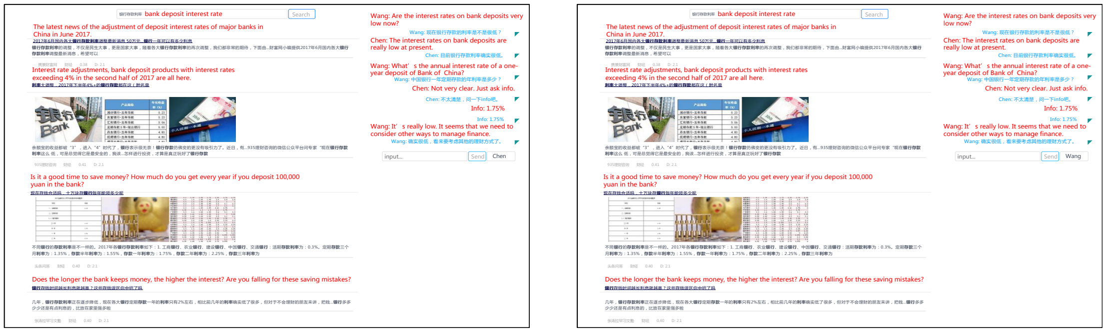
SCSS-LIE is coded with Python 3.5[1]. The user interface is implemented with Vue.js[2]. Vue.js is a widely used progressive framework for building user interface. The web server is supported by Tornado[3], which has the property of asynchronous and non-blocking I/O. Whoosh[4], a fast full-text indexing and searching library, is utilized to develop the search engine. The Chinese word segmentation tool - jieba[5] is implemented in the cluster engine to cut users' queries into tokens and extract keywords to facilitate the grouping of users. The interactive engine consists of two modules, namely the social engine and the Infobot. WebSocket[6] is implemented in the social engine to store session data and enable users in the same channel to communicate. In the implementation of the Infobot, we cut the related documents, conversation history and query into tokens by jieba and encode each token into a fixed-length vector (300-dim embedding). Then we choose two rounds of conversations as conversation history (N=2) and concatenate them with the current query. Then we use the bidirectional RNNs (BiLSTMs [4]) to integrate the conversation history into the current query. We take the last vector output from the BiLSTM as the query fusion vector. The query fusion vector and related document vector are then put into the BiDAF model to conduct reasoning and obtain answers. The DuReader dataset[7] [3], which is a large-scale open-domain Chinese dataset for MRC, is utilized to train the BiDAF model.

In our demo, we crawl about 250,000 pieces of data from Toutiao[8] to construct a search system and illustrate the feasibility of the proposed framework.

4 DEMONSTRATION

Figure 5 shows the overall interface of the demo we plan to show off at the conference. Figure 5 (a) and (b)respectively represent the interface of user Chen and user Wang. Chen and Wang are grouped into one chat channel to perform collaborative search when searching for similar query 'bank deposit interest rate' in the interactive engine. On the left side of the two figures are the search results returned by the search engine when the user searches for a query. On the right side of the two figures are the chat interface between Chen, Wang and the Infobot. They communicate with each other

[1]https://www.python.org/
[2]https://cn.vuejs.org/
[3]http://www.tornadoweb.org/en/stable/
[4]https://whoosh.readthedocs.io/en/latest/
[5]https://pypi.org/project/jieba/
[6]http://www.websocket.org/
[7]https://ai.baidu.com//broadsubordinate?dataset=dureader
[8]https://www.toutiao.com/



(a) (b)

Figure 5: Demonstration of SCSS-LIE. (a) The interface of user Chen. (b) The interface of user Wang. The contents of the light blue font on the right side of these two figures represent the conversation between Chen, Wang and the Infobot. The contents of the red font in the figures represent the translation of the conversation and some search results.

Table 2: Conversation between Chen, Wang and the Infobot.

Wang: 现在银行存款的利率是不是很低?
Wang: Are the interest rates on bank deposits very low now?
Chen: 目前银行存款利率确实很低。
Chen: The interest rates on bank deposits are really low at present.
Wang: 中国银行一年定期存款的年利率是多少?
Wang: What's the annual interest rate of a one-year deposit of Bank of China?
Chen: 不太清楚， 问一下 info 吧。
Chen: Not very clear. Just ask info.
Info: 1.75%
Wang: 确实很低，看来要考虑其他的理财方式了。
Wang: It's really low. It seems that we need to consider other ways to manage finance.

to share information, and then Chen inputs the symbol word 'info' to start the Infobot. The Infobot finds out the current user's query ' 中国银行一年定期存款的年利率是多少? (What's the annual interest rate of a one-year deposit of Bank of China?)' and gets the answer '1.75%' through BiDAF model and returns it to the chat interface. The conversation between them is illustrated in Table 2.

5 CONCLUSION AND FUTURE WORK

In this paper, we present a new intelligent search pattern for the synchronous collaborative search system. We exploit a novel ring topology to address the cold start problem of the social engine and improve the efficiency of collaborative search. Furthermore, our system provides users with a more intelligent interface to allow users to perform multi-turn QA interaction with the search engine to accomplish the search task and improve user satisfaction.

The current version of the demo is designed to verify our initial ideas, and there is still considerable room for further improvements. Firstly, we plan to employ more diverse grouping approach in the cluster engine. We will exploit some semantic understanding methods to further understand the user's query and make grouping more effective. Secondly, the performance of the Infobot relies on the capability of the reading comprehension model to understand natural language. It is necessary to design a more powerful reading comprehension model to understand natural language better. Finally, we will increase user satisfaction experiments to better test the performance of the Infobot and further optimize it through better interfaces and algorithms.

Overall, we hope that SCSS-LIE will be a good start for the research on intelligent synchronous collaborative search systems.

6 ACKNOWLEDGEMENTS

This work is sponsored in part by the state key development program of China (grant No. 2017YFE0111900), Natural Science Foundation of China (grant No. 61772363, U1636203), and the European Unions Horizon 2020 research and innovation programme under the Marie SkodowskaCurie grant agreement No. 721321.

REFERENCES

[1] Zhang Cheng, Zhang Peng, Jingfei Li, and Dawei Song. 2016. SECC:A Novel Search Engine Interface with Live Chat Channel. (2016).
[2] Colum Foley and Alan F. Smeaton. 2010. Division of labour and sharing of knowledge for synchronous collaborative information retrieval. *Information Processing & Management* 46, 6 (2010), 762–772.
[3] Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, and Qiaoqiao She. 2017. DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications. (2017).
[4] S Hochreiter and J Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[5] Hannarin Kruajirayu, Ake Tangsomboon, and Teerapong Leelanupab. 2014. Cozpace: a proposal for collaborative web search for sharing search records and interactions. In *Student Project Conference (ICT-ISPC), 2014 Third ICT International*. IEEE, 165–168.
[6] Meredith Ringel Morris. 2007. Collaborating alone and together: Investigating persistent and multi-user web search activities. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007). Amsterdam, Netherland: ACM*. 23–27.
[7] Sindunuraga Rikarno Putra, Felipe Moraes, and Claudia Hauff. 2018. SearchX: Empowering Collaborative Search Research. In *International Acm Sigir Conference*.
[8] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional Attention Flow for Machine Comprehension. (2016).
[9] Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. SDNet: Contextualized Attention-based Deep Network for Conversational Question Answering. (2018).