SCC — A Test Collection for Search in Chat Conversations

Ismail Sabei*
The University of Queensland
St Lucia, Australia
i.sabei@uq.edu.au

Ahmed Mourad
The University of Queensland
St Lucia, Australia
a.mourad@uq.edu.au

Guido Zuccon
The University of Queensland
St Lucia, Australia
g.zuccon@uq.edu.au

ABSTRACT

We present SCC, a test collection for evaluating search in chat conversations. Chat applications such as Slack, WhatsApp and Wechat have become popular communication methods. Typical search requirements in these applications revolve around the task of known item retrieval, i.e. find information that the user has previously experienced in their chats. However, the search capabilities of these chat applications are often very basic. Our collection aims to support new research into building effective methods for chat conversations search. We do so by building a collection with 114 known item retrieval topics for searching over 437,893 Slack chat messages. An important aspect when searching through conversations is the unit of indexing (indexing granularity), e.g., it being a single message vs. an entire conversation. To support researchers to investigate this aspect and its influence on retrieval effectiveness, the collection has been processed with conversation disentanglement methods: these mark cohesive segments in which each conversation consists of messages whose senders interact with each other regarding a specific event or topic. This results in a total of 38,955 multi-participant conversations being contained in the collection. Finally, we also provide a set of baselines with related empirical evaluation, including traditional bag-of-words methods and zero-shot neural methods, at both indexing granularity levels.

CCS CONCEPTS

• **Information retrieval → Test collections**.

KEYWORDS

Search for Chat Conversations; Test Collection; IR baseline methods

ACM Reference Format:
Ismail Sabei, Ahmed Mourad, and Guido Zuccon. 2022. SCC — A Test Collection for Search in Chat Conversations. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ..., ..., ..., USA, 5 pages. https://doi.org/10.1145/3511808.3557692

*Also with Jazan University, Saudi Arabia.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557692

1 INTRODUCTION

Chat applications, such as WhatsApp, Slack and Wechat have become the primary digital communication method alongside emails[1]. A characteristic in common to most chat applications is that they support both one-to-one communication (direct messages) as well as many-to-many communication (e.g., channels in Slack and groups in WhatsApp and Wechat). As more and more information is exchanged and stored in such chat applications, the need for searching these vast repositories grows, be it for personal use ("At what time is the party next week?") or business use ("Where is the file with the latest results that was posted by Bill?").

A conversation can be defined as an interaction and exchange of information between a group of participants, humans or machines. Conversational search [1, 4, 9, 12] considers searching information among data archives by synchronously interacting with the search agent. Search in Chat Conversations (the setting considered in this paper), on the other hand, considers searching among archives of previous conversations – and the search does not necessarily occur in the form of a conversation between the information seeker and the search functionality.

The current search functionalities offered by most chat platforms, however, are often "primitive", consisting of simple keyword matching and with no notion of a conversation (and many conversations may be intertwined at the same time within the messaging stream). Similarly, research on how to most effectively support search in these conversations repositories is limited. In this context, the work of Qin et al. [8] is a precursor to ours in that it provides a system to index and search conversations from messaging applications using standard information retrieval tools.

In this paper we present SCC, a test collection we have created for evaluating information retrieval methods for search in chat conversations. We do this by enriching, with queries and relevance assessments, an existing dataset for the disentanglement of conversations [3]. The availability of annotations regarding disentangled conversations across the messaging stream means it is also possible to model the notion of an individual message vs. a cohesive conversation: an important aspect, we argue, that can influence the granularity of the indexing process. In fact, along with the collection, we also provide baselines by implementing a number of retrieval methods, spanning both bag-of-words models and neural models, and evaluating them using our collection. Interestingly, we evaluate the methods by comparing their effectiveness when the indexing granularity is a single message vs. an entire conversation (i.e. a related sequence of messages). We believe our collection can foster research in the emerging area of search in chat conversations.

[1]https://www.statista.com/statistics/1112966/us-workers-communication-methods-during-coronavirus-pandemic/ [last visited 1 June 2022].

Table 1: Statistics of the Slack channels [3].

Channel	#conversations	#messages
pythondev#help	8,887	106,262
clojurians#clojure	7,918	72,973
elmlang#beginners	13169	168,689
elmlang#general	8,981	89,969
Total	38,955	437,893

2 SEARCHING THROUGH CHAT ARCHIVES

We consider a chat archive $M = \{m_1, m_2..., m_n | n = |M|\}$ where each m_i is a message and is defined as a tuple of $\langle id, content, conversation\_id \rangle$ where id is a unique message identifier and $content$ is the textual content of a message and $conversation\_id$ is a label that relates a message to a conversation c_k. We also consider an equivalent chat archive $C = \{c_1, c_2, ..., c_k | n = |C|\}$ where each c_i is a conversation and is defined as a tuple of $\langle id, contents \rangle$ where id is a unique conversation identifier and $content$ is the set of messages in this conversation, each represented as $\langle messages\_id : content \rangle$. The message and conversation ids in M and C are aligned. Current chat search functionalities are restricted to only finding messages. Our collection supports this setting, allowing for the retrieval of message-based units. However, our collection also supports an alternative view of this task, where the unit of retrieval is a conversation, rather than a message; this is done by integrating conversation disentanglement in order to retrieve conversation-based units. Formally, given a query set $Q = \{q_1, q_2, ...q_i\}$ and two collections M and C, an information retrieval system can return either the ranked lists M' or C'.

3 TEST COLLECTION

3.1 Conversations

SCC employs a collection of disentangled Slack conversations related to software development questions[2] [3]. The collection is sourced from four open Slack channels as indicated in Table 1. The selected chat channels are akin to community question answering threads, and thus may not be fully representative of other uses of chat tools, e.g., private communications between friends. However, we believe these channels still display the key characteristics of conversations within chat applications. It comprises 38,955 conversations and 437,893 messages, spanning two years (from July 2017 to Jun 2019), contributed by 12,171 users[3]. Further breakdown of the number of conversations and messages for each of the four channels is shown in Table 1.

3.2 Topics and Relevance Assessments

Given that SCC is related to software Q&A, the creation of the topics and relevance assessments were conducted by a group of computer science researchers from our lab[4]. We randomly sampled 300 conversations from C. Then, for each conversation, we asked the assessors to provide: *user's natural language question*, *keyword search query* and *relevant message ids*, thus effectively simulating a known item retrieval task. Note that not all conversations were cohesive; some conversations might be intertwined depending on the quality of disentanglement.

```
<?xml version='1.0' encoding='utf-8'?>
<Disentangled_Slack_Conversations>
  <Conversation id="1">
    <Message id="2018-12-31T00:08:47.720400">Ok</Message>
  </Conversation>
  <Conversation id="2">
    <Message id="2018-12-31T00:21:49.721700">
    Hello Guys i need help
    The issue is: I want to set up "AdminLTE-2.4.5" theme+CRUD in
    ↪  Django App at front-side, I get many answers like direct
    ↪  cmd through installing it and I do it but i don't know how
    ↪  to set up for the view
    Thanks
    </Message>
  </Conversation>
  <Conversation id="3">
    <Message id="2018-12-31T08:37:28.734200">
    Hi everyone :smile:
    say you have a list `_list = ['hello', 'world', 'pikachu']` and
    ↪  a string `greeting = 'hello, my name is pikachu'`
    how would you assert that any of the elements in the list are
    ↪  in the string ? Thanks for your help !</Message>
  </Conversation>
  <Conversation id="3">
    <Message id="2018-12-31T08:38:51.735000">
    You're looking for Time/Space complexity optimised solutions?
    </Message>
  </Conversation>
  <Conversation id="2">
    <Message id="2018-12-31T06:01:25.730500">
    To change 400 to 0.4, you divide by 1000. If you then want to
    ↪  format 0.4 as 0.40, that is something you do in the string
    ↪  formatting stage
    </Message>
  </Conversation>
</Disentangled_Slack_Conversations>
```

Listing 1: An example of disentangled conversations.

Looking at the nature of the specific chat corpus, a conversation is usually initiated with a question. The assessors were recommended to avoid relying on the conversation initial question and to generate keyword search queries from the middle of conversations based on their understanding of the context of conversations. However, a conversation initial question more often had the most representative keywords for a search query. After analysis of the natural language questions and queries, this yielded two keyword search query sets derived from: *Conversation Initial Question (CIQ)* and *Conversation Context (CC)*.

Lastly, the assessors were asked to select the messages that answer the question within the same conversation. Each conversation is judged by a single assessor. A total of 114 topics were created out of 300 conversations; the rest were discarded because of the low quality of disentanglement.

3.3 Characteristics of the Collection

Figure 1 shows the distribution of the number of messages per conversation; the majority of the conversations have 10 or less messages, with an average of 11.24. Figure 2 shows the conversation, message and query length in number of tokens as extracted by the monoBERT Tokenizer. Most of the conversations have 728 tokens (mean=635.57), messages have 34 tokens (mean=29.42) and queries have 11 tokens (mean=9.53).

[2]https://github.com/preethac/Software-related-Slack-Chats-with-Disentangled-Conversations/tree/v1.0.0 [last visited 1 June 2022]
[3]Users identities are anonymized in the original collection.
[4]http://ielab.io/

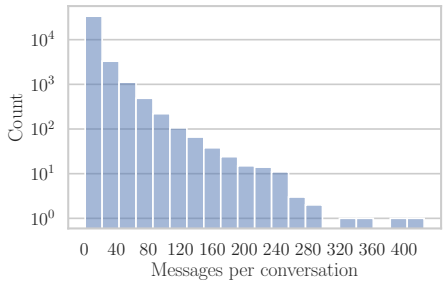

Figure 1: Number of messages per conversation.

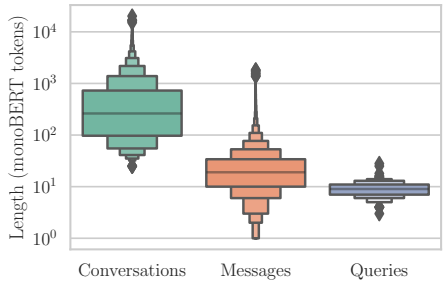

Figure 2: Length of conversations, messages and queries.

The original format of the collection was in XML as shown in Listing 1 where each message has a conversation id, a timestamp, an anonymised participant name and the message text. We convert the collection to JSON to facilitate processing with existing IR toolkits. For each message, we combine the conversation id and timestamp as a unique identifier for the messages and include the message text; we ignore the participant name. This creates the *M* collection. Similarly, we generate a unique id for each conversation and include a list of all message id and message text pairs in this conversation.

4 UTILITY OF THE COLLECTION

SCC provides a resource for a number of different research areas:

Conversations Search. SCC is developed to introduce the new task of ad-hoc conversations search, which has different characteristics than traditional search as mentioned in Sec. 1. It allows to explore questions such as: (1) what is the most effective indexing granularity (messages or conversations)? (2) what are the most effective retrieval approaches? (3) what are the challenges placed by the input size limits of pre-trained language models methods?

Conversations Disentanglement. The quality of conversations search relies on the quality of conversation disentanglement. SCC provides a resource for evaluating the impact of different disentanglement methods on the effectiveness of conversations search.

Query variations. Topics in SCC contain a natural language question and a corresponding keyword search query. This allows to measure the impact of these two different information need representations on the effectiveness of conversations search.

Conversational search. While conversations search is different from conversational search, SCC can be used to evaluate conversational agents. Online Q&A forums are popular sources of conversational datasets although they don't realistically simulate instant conversations. On the other hand, conversations disentanglement resolves this issue by constructing conversations from instant chat applications. SCC further provides the relevance assessments for the messages that provide answers in the conversation, which enables using it for conversational search as well.

5 EXPERIMENTS USING SCC

In this section, we aim to (1) provide a set of common baselines for the SCC collection on the task of ad-hoc search through conversations archives, (2) use the SCC collection and the baseline methods to investigate the impact of indexing granularity on the effectiveness of the retrieval methods. The latter demonstrates the utility of our collection to promote research on the emerging area of search through conversations archives.

5.1 Retrieval Methods

The considered methods, implemented using Pyserini [5], were:

(1) **BM25 [10]:** A simple bag-of-words retrieval baseline.
(2) **monoBERT [7]:** A neural reranker involving a first stage BM25 initial retrieval of 1000 documents, followed by a fine-tuned monoBERT reranking of the the top k=10 documents (without interpolation with BM25). We used a monoBERT model pre-trained on the MS MARCO [2] without further fine-tuning (i.e. zero-shot).
(3) **ANCE [11]:** A RoBERTa-based [6] dense retriever that selects negative training instances from an Approximate Nearest Neighbor (ANN) index of the corpus. We used an ANCE model pre-trained on MSMARCO (i.e. zero-shot).

5.2 Indexing Granularity

As part of our empirical efforts in providing baseline methods for the SCC collection, we aim to investigate the impact on search effectiveness of indexing conversations vs. indexing individual messages. To achieve this, we create four separate indexes based on the indexing granularity (conversation vs. message) and data representation (sparse for BM25 and monoBERT vs. dense for ANCE).

For BM25, we employed standard stop-word lists and Porter stemming. For monoBERT, the input is either a conversation or a message based on the index granularity used for the initial BM25 retrieval. For ANCE, we created two separate Faiss dense indexes: one obtained by encoding entire conversations, i.e. each dense vector represents a conversation, the other by encoding individual messages, i.e. each dense vector represents a message.

The length of the entire conversation may be larger than the maximum input allowed by either BERT or RoBERTa based encoders (512 tokens minus other required tokens). When this occurs, part of the conversation will not be encoded — while this amount is different between monoBERT and ANCE (e.g., due to different tokenizers and RoBERTa's vocabulary being larger than BERT's), these differences are minor. What is important is to instead understand how much of the input is not encoded and what the impact may be on effectiveness. We analyse this aspect further in Sec. 5.4.

5.3 Evaluation Measures

The nature of the ad-hoc search through chat archives task modelled by the SCC collection is similar to that of a known item retrieval task. In our SCC collection each query has only one target relevant conversation — the one used to construct the query topic. If the unit of retrieval then is a conversation, the SCC collection contains exactly

Table 2: Retrieval effectiveness of different models and Indexing Granularity (IG) using the SCC test collection. Statistical significance (p-value < 0.05) is measured using a two-tailed paired t-test with Bonferroni correction.

IG	Model	MRR@10	nDCG@10	R@10
Conv	[a]BM25	0.615	0.663	0.816
	[b]monoBERT	0.672	0.706	0.816
	[c]ANCE	$0.382^{a,b,d,e}$	$0.407^{a,b,d,e}$	$0.509^{a,b,d,e}$
Msg	[d]BM25	0.568^{b}	$0.591^{a,b}$	$0.693^{a,b}$
	[e]monoBERT	$0.542^{a,b}$	$0.568^{a,b}$	$0.675^{a,b}$
	[f]ANCE	$0.405^{a,b,d,e}$	$0.429^{a,b,d,e}$	$0.535^{a,b,d,e}$

Table 3: Retrieval effectiveness of different models and Indexing Granularity (IG) per query set (CIQ vs. CC).

IG	Model	CIQ-Topics			CC-Topics		
		MRR@10	nDCG@10	R@10	MRR@10	nDCG@10	R@10
Conv	[a]BM25	0.709	0.753	0.88	0.439	0.501	0.692
	[b]monoBERT	0.694	0.729	0.88	0.522	0.563	0.692
	[c]ANCE	$0.453^{a,b}$	$0.495^{a,b}$	$0.627^{a,b}$	$0.127^{a,b,d,e}$	$0.157^{a,b}$	$0.256^{a,b,d,e}$
Msg	[d]BM25	0.644	$0.633^{a,b}$	$0.747^{a,b}$	0.398	0.431^{a}	0.539
	[e]monoBERT	0.649^{b}	$0.676^{a,b}$	$0.773^{a,b}$	0.337^{b}	$0.364^{a,b,d}$	$0.487^{a,b}$
	[f]ANCE	$0.508^{a,b,d,e}$	$0.537^{a,b,d,e}$	$0.627^{a,d,e}$	$0.177^{a,b,d,e}$	$0.220^{a,b,d,e}$	$0.359^{a,b,d,e}$

one relevant document (conversation) per query. To evaluate this, we use MRR@10, Recall@10 and nDCG@10. For nDCG@10 we are interested in its rank-discounting behaviour, compared to the sharp one of MRR — our collection does not contain graded relevance so all relevant documents provide the same amount of gain. We use the rank cut-off 10 for evaluation because, in our experience, most users of chat applications do not go beyond the first few rank positions. Cut-off of 10 is also in line with the current evaluation efforts in passage retrieval, and with common information retrieval practice.

We further note that if messages were used as the unit of retrieval, in place of conversations, then SCC contains one or more relevant messages for each query. The measures identified above would still be suitable for this evaluation. However, although it is possible to use messages as the unit of retrieval and evaluation, we do not believe that this is a good modelling of the actual reality. We do observe that some chat applications do return individual messages as search results, e.g. Slack. However these are a "hook" into, or snippet of, a conversation: when users click on such a message, they are brought to the whole conversation (or stream of previous and subsequent messages). We thus believe that using conversations as unit of evaluation, rather than messages, is a more appropriate modelling of reality. To achieve this, we map the retrieved messages to their respective conversations followed by pruning the ranked list of conversations to remove duplicates.

5.4 Results

Table 2 reports the effectiveness of baselines for all 114 queries.

Term-based vs neural model effectiveness. There are mixed results between the term-based BM25 and neural models (monoBERT and ANCE). monoBERT is more effective, yet not statistically significant, than BM25 for the conv-based index (same R@10 as it is a re-ranker). Conversely, BM25 is more effective than monoBERT for the msg-based index. ANCE consistently achieves the worst effectiveness over all metrics and indexes with statistical significance.

Indexing granularity. There is a large difference in effectiveness between the conv-based index and the msg-based index. BM25 and monoBERT are statistically significant far more effective for the conv-based index than their counterparts for the msg-based index across all evaluation metrics. On the contrary, ANCE for the msg-based index is more effective , yet not statistically significantly, than for the conv-based index.

Query sets. Table 3 shows the effectiveness of our models and indexing granularity per query set: CIQ and CC (See Sec. 3.2). For the conv-based index, the term-based BM25 tends to outperform the neural models for CIQ topics, while monoBERT is more effective for the CC topics. On one hand, the overlap in terms between the conversation initial question and the generated search query can explain the effectiveness of BM25 over CIQ topics, while the semantic matching power of monoBERT can justify the effectiveness over CC topics that do not have much overlap in keywords with the conversation. On the other hand, the latter argument does not hold for ANCE, as it consistently performs worse than other models. As for the influence of indexing granularity, our retrieval models for the conv-based index are consistently more effective than their counterparts for the msg-based index.

Is the evaluation reliable? Our collection and the associated evaluation has relied on the premise that there is only one relevant conversation for a query – the known item used at collection creation. To investigate the limitations of this assumption (and thus whether the evaluation provided above is reliable), we sampled 5 queries from each of CIQ and CC topics. For each query, we assessed the relevance of the top k=10 retrieved conversations (conv-based index only). We found that in two cases there was a consistent number of additional relevant conversations other than the known item one (qid: "41" had 4 additional relevant documents, "4" had 7), and in other 2 cases there were only one additional relevant document. Upon further analysis, we identified these cases to be associated with broad and under-specified queries (e.g., "decode JSON object"). However, for the remaining queries in the sample, no additional relevant documents were found. The analysis of the non-sampled information needs highlighted broad topics were rare. We conclude that the large majority of information needs in SCC are indeed true known item retrieval topics.

6 CONCLUSION AND FUTURE WORK

In this paper we have contributed the SCC collection for evaluation of information retrieval systems for searching through chat conversations. The collection contains over 38K conversations from 437K Slack chat messages and 114 known item retrieval topics with associated relevance assessments. As part of the collection, we provide an evaluation of common baselines and observations regarding the effectiveness of different indexing granularities. The collection is available at https://github.com/ielab/SCC. Our future work will extend this collection with further queries and assessments; and by considering retrieval models specifically designed for this task. Specifically, it would be interesting to consider neural models capable of handling longer conversations without truncation.

Acknowledgement. We thank Dr. Essa Alhazmi (Jazan University) for sharing initial result visualisation code.

REFERENCES

[1] Avishek Anand, Lawrence Cavedon, Hideo Joho, Mark Sanderson, and Benno Stein. 2020. Conversational search (dagstuhl seminar 19461). In *Dagstuhl Reports*, Vol. 9. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[3] Preetha Chatterjee, Kostadin Damevski, Nicholas A Kraft, and Lori Pollock. 2020. Software-related slack chats with disentangled conversations. In *Proceedings of the 17th International Conference on Mining Software Repositories*. 588–592.

[4] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. TREC CAsT 2019: The conversational assistance track overview. *arXiv preprint arXiv:2003.13624* (2020).

[5] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. *Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations*. Association for Computing Machinery, New York, NY, USA, 2356–2362. https://doi.org/10.1145/3404835.3463238

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[7] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[8] Kunpeng Qin, Harrisen Scells, and Guido Zuccon. 2021. PECAN: A Platform for Searching Chat Conversations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2610–2614.

[9] 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*. 117–126.

[10] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[11] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

[12] Hamed Zamani, Johanne R Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational Information Seeking. *arXiv preprint arXiv:2201.08808* (2022).