PECAN: A Platform for Searching Chat Conversations

Kunpeng Qin
kunpeng.qin@uq.net.au
The University of Queensland
Brisbane, Australia

Harrisen Scells
h.scells@uq.edu.au
The University of Queensland
Brisbane, Australia

Guido Zuccon
g.zuccons@uq.edu.au
The University of Queensland
Brisbane, Australia

ABSTRACT

Often, existing chat services that organisations and individuals use today provide a way to search through previously sent messages. However, many of these chat services provide far-limited search functionalities, typically exact matching on individual messages. In this paper, we introduce a new task for addressing this problem, called *searching for conversations*, whereby the aim is to retrieve and rank groups of related messages given a search query. We promote this task by providing a platform for research and development called PECAN. Our platform provides all the necessary functionality researchers need to conduct experiments on searching for conversations. Our system is also generic so as to support organisations and individuals who wish to search through their chat message archives.

We release PECAN to the wider community as an Open Source project available for download at https://github.com/ielab/pecan.

KEYWORDS

search interface, chat search, searching for conversations

ACM Reference Format:
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 (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3462791

1 PROBLEM, TARGET USERS & IMPORTANCE

Many search offerings of popular chat and messaging services of the likes of Slack, Discord, Signal, Whatsapp, etc., offer a sub-par experience: they mostly only allow exact matches between the user's query and their messages. However, this is not ideal because (1) the messages surrounding a targeted message often contain context to the information need, (2) semantic mismatch between the query and the messages severely impact retrieval quality. An example of this is when searching for an answer to a question but not remembering how the question was posed. To find their answer, one may search for a conversation they knew that happened within the same time period. Indeed, within the context of chat message search, one is often not interested in individual messages relevant to a search query, but a *conversation* of topically related messages. As search practitioners, we often want to understand how Information Retrieval systems work to improve them. Furthermore, we suggest that existing chat search systems can be drastically improved by addressing the task of 'search for conversations' (rather than messages). In this new task, search within chat messages should not be restricted to individual messages and conversations of messages are the ideal unit of retrieval. To support researchers investigate and experiment with this task, we propose **PECAN**: A **P**latform for s**E**arching **C**h**A**t co**N**versations.

PECAN targets teams and organisations of chat services that wish to provide their own search solution and Information Retrieval practitioners that wish to study how users search for conversations and improve the effectiveness of searching for conversation. PECAN provides all of the necessary tools to get started working on this task: a componentised framework, which can be further enhanced through the upgrade or implementation of new features, offline/batch retrieval for evaluation, and user study instrumentation (administration, query logging, interaction logging). An overview of PECAN's features and architecture is presented in Figure 1.

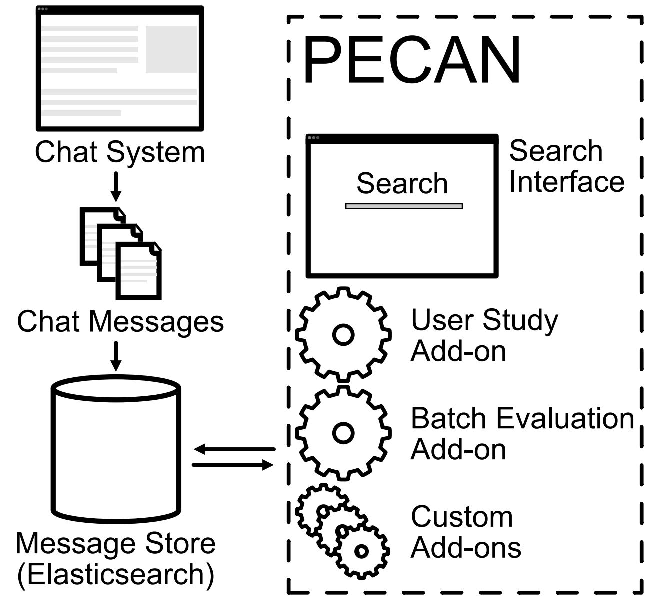


Figure 1: Overview of the architecture of the PECAN system.

2 SUPPORTED RESEARCH TASKS

To the best of our knowledge, PECAN is the only system currently available to enable research into this new task of 'searching for conversations'. There are many facets to this new research task that PECAN can be used for implementing new algorithms:

Conversation Aggregations: The challenge involved here is how to represent the 'unit of retrieval'. Two aggregation aspects can be addressed: Message Relevance and Conversation Boundaries. Message Relevance is the task of identifying which messages within a conversation are relevant and possibly discarding irrelevant messages from the presentation of the results. Conversation Boundaries is the task of identifying when a conversation begins and ends and merging retrieved conversations that overlap in time or topic.

Conversation Scoring: This is the task of scoring conversations given a query to provide a ranking of conversations in terms of relevance, for example. It is currently an open question for the most effective way to rank conversations and may depend on various factors from typical search scenarios, such as timeliness.

Query Suggestion: This is the task of suggesting alternative queries for searching conversations. We hypothesise that most searches for conversations are known-item retrieval (e.g., a user is searching to find an answer to a question they previously had but cannot remember how the question was asked or the answer). This scenario presents itself as a unique opportunity to apply query suggestion to known-item retrieval.

Conversation Summarisation: Conversations are rich sources of information that contain the history of questions and answers asked by a team or organisation. However, finding answers to complex questions within conversations can be tedious as one may need to read through many messages to find the answer. This task aims to summarise conversations to provide an answer to a question or a high-level overview of a conversation.

Related Conversations: This is the task of identifying similar conversations given a conversation. This task is useful in the case of, for example, showing that a topic of conversation has been discussed previously (i.e., preventing time-wasting, keeping track of important decisions).

3 OPERATION OF THE SYSTEM

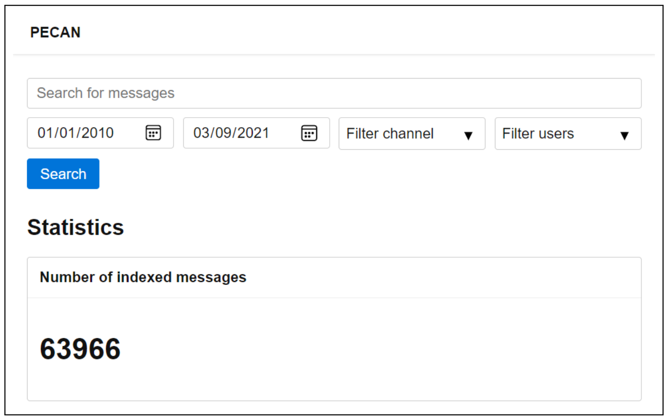
The architecture of PECAN comprises a search engine message store, and web-server back-end, and several add-ons that provide the functionality for running user studies and batch-style evaluation for the task of search for conversations. PECAN is made for bring-your-own-data: it provides the tools to search, present, and perform experiments on your data, but currently it does not provide a means to scraping chat messages. As such, an overview of the fields that are required to be indexed for the basic functioning of PECAN are presented in Table 1. We have chosen these fields as, in our opinion, they are the minimum required fields that almost all chat messaging systems today would use. PECAN however allows for thee customisation of this list.

3.1 Searching for Conversations

We first present a small case study that demonstrates some of the user interfaces that are built into PECAN. The homepage (Figure 2) of PECAN shows statistics about the number of messages that are indexed and several recent messages that have been indexed. One thing to note immediately is that it is possible for PECAN to (optionally) apply the same security considerations as the system that was originally used to send the messages. This allows private messages within a chat system to remain private. We achieve

Field	Description
Text	The textual content of a message.
Timestamp	The time and date that a message was sent.
User	The user (or user identifier) that sent a message.
Type	The type of message that is sent.
Channel	Where a message was sent.

Table 1: Necessary fields that must be included in the message index for the correct operation of PECAN. The purpose of the *Type* field is to visually present messages in different ways, e.g., quoted messages. The *Channel* field specifies the 'location' the message was sent in a chat system, e.g., the public/private spaces of services such as Slack or Discord, or the groups or direct messages of services such as Signal or Whatsapp (note that services such as Slack and Discord also have direct messages).


PECAN

Search for messages

01/01/2010 03/09/2021 Filter channel Filter users

Search

Statistics

Number of indexed messages

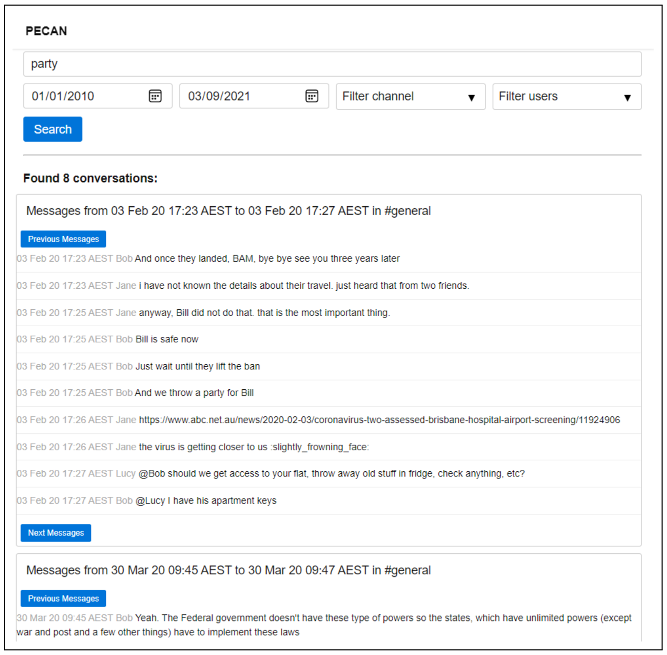
63966

Figure 2: The homepage of PECAN. Here, users can enter their search query and apply any search filters to refine their search. Statistics about the number of messages that are indexed are also shown, alongside several recently indexed messages (omitted for space reasons).

this by providing a pluggable authentication system. [1] Within the homepage, users may enter queries and filter their query to certain time periods, channels, and other users (depending on the original chat system). PECAN can also be configured to record query logs and fine-grained user interactions with the system, for improving searching for conversations effectiveness and for studying user behaviour in a user study setting.

Once users issue a query to PECAN, they are presented with a conversations results page (CORP), as presented in Figure 3. On this page, users can explore conversations that are relevant to their query. Conversations are ranked according to their relevance to a given query. We describe our approach to scoring conversations in Section 3.2.2. We choose to select five messages before and after a retrieved message to form a conversation. Note that this may result in some conversations with overlapping messages. To address this, we propose to aggregate conversations with overlapping messages. As a result, the CORP may contain conversations with more than ten messages. Our approach to aggregating conversations is described

[1]For more information about this, please refer to the GitHub repository.



PECAN

party

01/01/2010 | 03/09/2021 | Filter channel | Filter users

Search

Found 8 conversations:

Messages from 03 Feb 20 17:23 AEST to 03 Feb 20 17:27 AEST in #general

Previous Messages

03 Feb 20 17:23 AEST Bob And once they landed, BAM, bye bye see you three years later

03 Feb 20 17:23 AEST Jane i have not known the details about their travel. just heard that from two friends.

03 Feb 20 17:25 AEST Jane anyway, Bill did not do that. that is the most important thing.

03 Feb 20 17:25 AEST Bob Bill is safe now

03 Feb 20 17:25 AEST Bob Just wait until they lift the ban

03 Feb 20 17:25 AEST Bob And we throw a party for Bill

03 Feb 20 17:26 AEST Jane https://www.abc.net.au/news/2020-02-03/coronavirus-two-assessed-brisbane-hospital-airport-screening/11924906

03 Feb 20 17:26 AEST Jane the virus is getting closer to us :slightly_frowning_face:

03 Feb 20 17:27 AEST Lucy @Bob should we get access to your flat, throw away old stuff in fridge, check anything, etc?

03 Feb 20 17:27 AEST Bob @Lucy I have his apartment keys

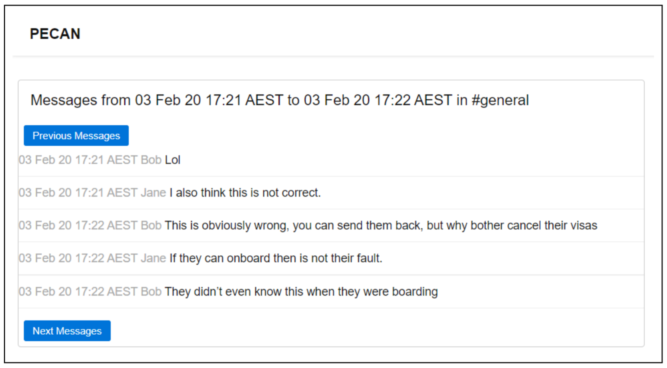
Next Messages

Messages from 30 Mar 20 09:45 AEST to 30 Mar 20 09:47 AEST in #general

Previous Messages

30 Mar 20 09:45 AEST Bob Yeah. The Federal government doesn't have these type of powers so the states, which have unlimited powers (except war and post and a few other things) have to implement these laws

Figure 3: The CORP page of a response to an example query 'party'. Each conversation is ranked in terms of relevance to the input query, similar to contemporary SERPs. Conversations can be explored further by interacting with the 'Previous Messages' and 'Next Messages' buttons at the start and end of each conversation.



PECAN

Messages from 03 Feb 20 17:21 AEST to 03 Feb 20 17:22 AEST in #general

Previous Messages

03 Feb 20 17:21 AEST Bob Lol

03 Feb 20 17:21 AEST Jane I also think this is not correct.

03 Feb 20 17:22 AEST Bob This is obviously wrong, you can send them back, but why bother cancel their visas

03 Feb 20 17:22 AEST Jane If they can onboard then is not their fault.

03 Feb 20 17:22 AEST Bob They didn't even know this when they were boarding

Next Messages

Figure 4: The conversation exploration page that users are presented with after clicking one of the 'Previous Messages' or 'Next Messages' buttons at the start and end of each conversation on the CORP. Users can continue to load more messages in the conversation indefinitely.

in Section 3.2.1. PECAN allows full customisation of these settings and the associated algorithm.

Finally, each conversation in the CORP can be further explored by viewing messages that occur before or after a conversation. Each conversation in the CORP contains buttons to view messages that occur before or after a conversation. Users are able to cycle through more messages indefinitely in order to fulfil their information need. The interface that users can use to explore these messages is presented in Figure 4.

Algorithm 1: High-level overview of how conversations are retrieved, scored, and ranked given a query.

```
   inputs : Search query $q$
   output : List of conversations $C$
1  Initialise messages $M$ as the result of executing $q$
2  Initialise $C \leftarrow \emptyset$
3  foreach $m \in M$ do
4      Add to $C$ the list of $k$ surrounding messages of $m$
5  end
6  Merge any overlapping $c \in C$ using Algorithm 2
7  return $C$
```

3.2 System Implementation

PECAN is implemented using the Go programming language. Go was chosen as it has been demonstrated to be an efficient server programming language. This is important as advanced algorithms that need to be executed at retrieval time may pose considerable overheads (e.g., complex conversation aggregation or scoring algorithms). For the underlying message storage engine, we use Elasticsearch 7.x. Elasticsearch is a highly scalable search engine that is capable of ingesting and retrieving documents (in our case messages) in real time.

The current PECAN release provides baseline implementations of two of the conversation search related research tasks mentioned in Section 2: conversation aggregation and conversation scoring. The following two sections provide a description of how we implement these methods in PECAN. These implementations are part of a pluggable system, where new implementations are designed to be easy to add and extend. Please refer to the GitHub repository for details on how to implement new methods.

3.2.1 Conversation Aggregation. There are many ways to retrieve conversations instead of messages, for example, one could decide to group messages together based on time, topicality, among others. The default behaviour we provide in PECAN is to aggregate messages into conversations based on the time messages were sent. A high-level overview of this algorithm is provided in Algorithm 1. To form conversations, we first retrieve individual messages given a query. Next, we retrieve the $k = 5$ messages that were sent before and after the original message (including the original message). This is a fast operation within Elasticsearch, and can be done at retrieval time. All $2 \cdot k$ messages are then combined into a single conversation. One implementation detail we note is that for chat systems that have multiple 'channels', we do not merge messages from different channels. This is because we believe that a conversation is tied to the context within which it occurred.

Following the creation of the conversations, we proceed to merge conversations that contain overlapping messages. This process is detailed in Algorithm 1.

3.2.2 Conversation Scoring. After merging conversations, we rank these conversations based on their scores. Our conversation scoring method exploits the fact that individual messages are scored at retrieval time (i.e., line 1 in Algorithm 1). By default, each conversation inherits the score assigned to the original message that formed the conversation. However, when a conversation is composed by merging two conversations together, we sum the scores of the

Algorithm 2: High-level algorithm for merging and scoring conversations.

```
inputs : List of conversations $C$
output : Merged conversations $\hat{C}$
1  Initialise $\hat{C} \leftarrow \emptyset$
2  foreach $c_i \in C$
3  do
4      if $c_i \in \hat{C}$ then
5          foreach $c_j \in C$ do
6              if $c_i \cap c_j > 0$ then
7                  $c_i \leftarrow c_i \cup c_j$
8                  Set score of $c_i$ to score($c_i$)+score($c_j$)
9                  goto 13
10             end
11         end
12     else
13         Add $c_i$ to $\hat{C}$
14     end
15 end
16 return $\hat{C}$
```

two messages that overlap (line 8 in Algorithm 2). We predict that there are many more signals for relevance within conversations, for example the time-frame a conversation spans or the length of a conversation, that can greatly improve ranking effectiveness. We leave this investigation for future work.

3.3 Add-ons

PECAN is designed to be a flexible research tool for conducting experiments involving the task of searching for conversations. As such, we provide two add-ons that cover many of the types of research that can be performed using the tool.

3.3.1 Running a User Study. In addition to the common use-case of searching for conversations, as any organisation or individual may want to use PECAN for, we also provide research-grade user study facilities in the form of an add-on. This add-on allows one to study aspects of searching for conversations such as user behaviour. In particular, the added functionality consists of:

Query Logging. Queries that users issue to the system, and the filters that they apply can be logged directly to a file.

Interaction Logging. Fine-grained user interactions with the web browser, such as mouse movements or keyboard events, can be logged directly to a file.

Screen Recording. Live screen recording of users interacting with the system, including their cursor, can be logged directly to a directory containing a series of images for each capture.

Administration. Design searching for conversations user experiments, specify task rotations, and administer users.

3.3.2 Batch Evaluation. Another add-on that we provide is the ability to run and evaluate batch-style evaluation for the task of searching for conversations. This add-on exposes a RESTful service where one may upload a set of queries and receive a TREC formatted run file as a response. In addition, one may specify an implementation of the research tasks listed in Section 2. This could be to, for example, compare the effectiveness of two or more conversation scoring algorithms.

4 COMPARISON WITH EXISTING SYSTEMS

In our survey of the literature, we found no such research for the explicit task of searching for conversations. More so, we did not identify any such research-oriented systems for searching chat conversations, or messages for that matter. However, in terms of the research tasks related to searching for conversations put forward in Section 2, we have identified a number of works that have addressed these tasks, and would have benefited from the PECAN ecosystem.

For the task of conversation scoring, Magnani et al. [4] propose a conversation ranking framework for micro-blogging (e.g., Twitter). For this task, there are many signals of relevance that do not exist in our task, for example, popularity metrics such as likes. This task also models conversations with explicit replies to posts. Our task differs in that multiple, different, overlapping conversations may occur all at the same time.

In terms of conversation aggregation, Khan et al. [3] identify this as a problem for studying tasks involving chat messages and propose a rule-based classification system to detect when conversations begin and end. However, their results highlight that rule-based methods are not scalable, especially in multi-lingual contexts. Shen et al. [6] takes a different approach and instead attempt to detect topically related conversations in chat messages using a novel single-pass clustering algorithm. Such methods are possible to be implemented directly into PECAN, allowing research into this task.

While we did not identify any works on query suggestion for chat messages or chat conversations, we did identify research in related domains. For example, Mishne et al. [5] present an architecture of Twitter's real-time related query suggestion, where relevance signals from query logs are used to suggest related queries to users. Carmel et al. [2] investigate demographical applications to query suggestion in email search, where one could consider similarities between email and chat messages. Their results indicate that personalisation is important for suggesting effective queries. PECAN's query logging may be used to capture data related to these works, allowing research into this task.

For the task of conversation summarisation, Bengel et al. [1] propose an interface for intelligence agencies to monitor chat participants that can identify latent topics of conversations in channels or of individual users. They also provide a basic interface that allows agents to search for messages. Our idea of conversation summarisation differs from this in that we seek to provide a succinct summary of the conversation that took place, rather than trying to classify the topics of a conversation.

5 IMPACT OF THE SYSTEM

It is our opinion that the task of searching for conversations is highly important for organisations with many members in unlocking the information that is trapped in conventional chat systems. We believe that PECAN will spur new research into this emerging task of searching for conversations. With PECAN, we provide the infrastructure for others to build upon and begin work on developing new methods for this task.

Acknowledgements. Dr Guido Zuccon is the recipient of an Australian Research Council DECRA Research Fellowship (DE180101579) and Google Faculty Award.

REFERENCES

[1] Jason Bengel, Susan Gauch, Eera Mittur, and Rajan Vijayaraghavan. 2004. Chattrack: Chat room topic detection using classification. In *International Conference on Intelligence and Security Informatics*. Springer, 266–277.

[2] David Carmel, Liane Lewin-Eytan, Alex Libov, Yoelle Maarek, and Ariel Raviv. 2017. The demographics of mail search and their application to query suggestion. In *Proceedings of the 26th International Conference on World Wide Web*. 1541–1549.

[3] Faisal M Khan, Todd A Fisher, Lori Shuler, Tianhao Wu, and William M Pottenger. 2002. Mining chat-room conversations for social and semantic interactions. *Computer Science and Engineering, Lehigh University* (2002).

[4] Matteo Magnani, Danilo Montesi, and Luca Rossi. 2012. Conversation retrieval for microblogging sites. *Information retrieval* 15, 3 (2012), 354–372.

[5] Gilad Mishne, Jeff Dalton, Zhenghua Li, Aneesh Sharma, and Jimmy Lin. 2013. Fast data in the era of big data: Twitter's real-time related query suggestion architecture. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 1147–1158.

[6] Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread Detection in Dynamic Text Message Streams. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, Washington, USA) *(SIGIR '06)*. Association for Computing Machinery, New York, NY, USA, 35–42. https://doi.org/10.1145/1148170.1148180