A Self-learning Conversational Agent Framework for Team Collaboration

Barbaros Özdemir\textsuperscript{1,2} and Gerald Quirchmayr\textsuperscript{2}

\textsuperscript{1} IBM Austria, Obere Donaustraße 95, 1020 Vienna, Austria, barbaros.oezdemir@at.ibm.com
\textsuperscript{2} University of Vienna, Währinger Straße 29, 1090 Vienna, Austria, gerald.quirchmayr@univie.ac.at

Abstract. The conversational agent framework proposed in this paper employs data extraction algorithms to extend the conversational agents’ knowledge base with new information retrieved from goal-oriented human-to-human conversations without the need for manual intervention. When utilized as a team collaboration tool, this framework will help prevent the loss of internally generated information and make it available for the whole team. A demo case illustrates how the method works in supporting service teams in a collaborative way. The paper finishes with an outlook on future work.

Keywords: conversational agent, information retrieval, chatbot, collaboration, watson assistant, ibm, goal-oriented, conversation logs, unsupervised learning, human-in-the-loop, feedback, self-learning, team support

1 Introduction

Conversational agents are employed for a wide range of use-cases [1]. They let organizations interact with more users than the capacity of their human resources would allow. They also provide service recipients with easy access to information. However, it is critical to maintain conversational agents over time by evaluating their efficiency and improvement of their content, as mentioned in [2]. Such maintenance activities are bound with high costs due to the employment of human labor [3].

This paper introduces a framework based on the human-in-the-loop conversational agent [4] design pattern. The framework development has been guided by the information systems research framework introduced in [5]. The loops have been distilled to definition the goal, analysis of related work, design, and implementation, similar to [6].

The framework differentiates itself from other approaches with a feedback loop that utilizes the information generated during internal goal-oriented conversations between team members and makes this information available for the rest of the team without the need for manual interference (i.e., maintenance work). Each user starts interacting with the conversational agent as a client. Clients can ask questions and evaluate the answer they received as positive or
negative based on its correctness and relevance. A positive evaluation requires that the answer is both relevant and correct. Users should evaluate all other cases as negative. A non-helpful answer indicates the necessity for new knowledge to be introduced. A helpful but wrong answer may reveal a scope creep or change of an existing answer. Both cases are to be addressed by the data extraction algorithm. Any user can switch its role from client to subject-matter expert by requesting it from the conversational agent. If the conversational agent cannot answer a question, it will allow the client to talk to the next available subject-matter expert. If the client accepts, the conversational agent will redirect the conversation accordingly. Depending on the availability of gathered human-to-human conversation logs covering a particular topic, the algorithms employed by the framework are expected to trigger an information retrieval process and add new information to the conversational agent’s knowledge base.

2 Related Work

Research in information retrieval goal-oriented conversation logs can be grouped based on text corpora (e.g., documents, forum entries, conversations) and, when applicable, its participants (i.e., human, machine) used as the data source.

In [7], [8], [9], [10] and [11] studies on data extraction from online documents, conversation corpora and verification have been presented. [7] presents a custom trust model and an "Automated Knowledge Extraction Agent (AKEA)". The proposed solution in this paper also makes use of an agent component with a similar goal. The research in [12] introduces a cascaded hybrid model to extract information from online discussion forums, which can be considered similar to human-to-human conversation logs. Similar algorithms can also be applied to conversation logs. The research in [13] applies data mining techniques on human-to-human conversation logs to extract client requests. The research in [14] also uses the same type of conversation logs as the data source. The system described in [14] is the closest approach to the framework introduced in this paper. It is semi-automated and uses different algorithms for intent detection. It applies sentence encoders to generate context vectors. Through vector-based nearest neighbor search, similar utterances are detected and marked as intents. Finally, the solution provides the maintenance personnel of the conversational agent with a list of intents and their corresponding potential answers. This way, the maintenance of the conversational agent can be conducted more efficiently while still requiring manual interference.

The main inspiration for the framework proposed in the next chapter is the model introduced in [14]. The proposed framework extends existing work by replacing the k-means clustering with the Damerau-Levenstein algorithm and removing the Bot designer’s manual intervention. The mature contribution of this paper can therefore be seen as having introduced increased automation that improves the reduction of manual labor in bot creation.
3 Proposed Framework

This chapter introduces a framework for a conversational agent capable of supporting team collaboration by retaining internally generated information and making it available to all team members. The proposed framework consists of four component groups shown in Fig. 1: Frontend and Backend components for interaction between the conversational agent and human users, Agent component for data extraction, IBM Watson Assistant and Databases for persistence. The Emulator component is designed for testing purposes only and does not belong to the framework. This framework differentiates itself from human-in-the-loop constructs e.g. in [4] by not requiring the operator (i.e., subject-matter expert) to intervene in the adjustment of the training data.

![UML Component diagram illustrating interfaces of the framework's components.](image)

The Frontend component is responsible for creating user sessions. To send a message, the Frontend sends a REST Call to the ”publish” endpoint, and to receive a message; it applies long polling at the ”subscribe” endpoint of the Messaging component. The Messaging component relays all incoming messages from the Frontend to the Adapter component, while it stores all incoming messages from the Adapter component in the Message DB. This way, asynchronous communication capability is achieved. The Adapter component is responsible for the management of user sessions. It matches each user id with the relevant conversation id. The Adapter is also responsible for managing user roles. All users start their interaction with the conversational agent in the client role and may switch to the role of subject-matter expert as shown in Fig. 2. To switch the role, a user only needs to write that he/she is a subject-matter expert into the input field. The conversational agent will then mark that user respectively.

It informs Orchestrator when required about the next available user in the role of the subject-matter expert. The Orchestrator component retrieves the
context of Watson Assistant from Conversation Session DB based on the conversation id it received from the Adapter component and forwards it with the user input to Watson Assistant. If there is no such id, a new conversation will be started. This component is also responsible for changing the message’s recipient (i.e., conversational agent or another user) and logging all interactions. The Agent (i.e., Knowledge Extraction Agent, not to be confused with the conversational agent) component runs in a loop. It extracts new information from human-to-human conversation logs once enough data is available. As stated in [15] Watson Assistant has a higher F1-score than other platforms and therefore contains comparatively less inconsistency in its answer generation process. That is why the designed framework utilizes IBM’s Watson Assistant technology to store the retrieved information and make it available for future use through a conversational agent.

4 Prototype Implementation

The prototype’s goal is to prove that the framework can be utilized as a team collaboration tool to preserve information and make it available to other team members.

Conversation logs are generated using the Emulator component’s help by using an existing conversational agent’s knowledge base to test the prototype. Using the intent (i.e., question) examples, the Emulator started conversations with the conversational agent with an empty knowledge base. Parallel to the initiated conversation, the Emulator registered itself as another user with the role of a subject-matter expert and used the existing conversational agent to answer incoming questions. To make emulated conversations more human-like, we introduced handshake messages (i.e., greeting, goodbye). For testing purposes, we only used two questions that the Emulator could rephrase in many different ways. Once the conversation logs similar to in Fig. 3 are generated, the Agent component starts running the algorithm for information extraction. For test purposes, an evaluation of the correctness of answers is not considered but planned be added in future.

The algorithm utilized by the Agent component consists of two parts; data preparation and data extraction, as shown in Fig. 4.

In the final step, all questions and the respective answers are grouped as ”intent”s similar to the two examples as shown in Fig. 5. Lastly, the Agent component creates the extended list of intents and their respective dialog nodes.
Hi, this is the Team Collaboration Chatbot. How can I help you?

How can we reimburse a subscription?

Sorry, I couldn’t understand your question. Would you like to talk to a subject-matter expert?

Yes

Subscriptions can be reimbursed as Non-Travel Expenses.

thank you

no problem

bye

so long

Fig. 3. A snippet from conversation logs generated by the emulator component with indicators on left pane showing messages group, e.g. All, human-to-human (i.e H2H), without handshake (i.e. wohs), question and answer (i.e. Q&A).

Fig. 4. The Algorithm utilized to extend conversational agent’s knowledge base with information extracted from human-to-human conversation logs.

These dialog nodes are activated when a specific intent has been detected in user input and deliver a particular answer for a given intent. The newly created knowledge base is then uploaded with an incremented version number to Watson Assistant and is ready to be used by the conversational agent in all future conversations.

5 Initial Results

The prototype has been tested with 732 Human-to-human conversation logs fed to the Agent component. Around 30 examples for each of the two Question & Answer sets could be retrieved from 61 conversation threads. The number of logs generated in total (i.e., All) and the number of valuable data extracted (i.e., Q&A), and the amount of data at each data preparation step has been put into
perspective in Fig. 6. The algorithm correctly identified intents and relevant answers from logs to create the same behavior as the original conversational agent.

![Figure 6](image_url)

**Fig. 6.** Number of All Logs, Logs from emulated Human-to-Human interactions (i.e. H2H), Logs after filtering handshake messages, Q&A Logs and conversation Threads.

When deployed as a team collaboration tool, the solution can catch specific knowledge on interaction with tools to conduct daily business activities, which would otherwise have to be figured out by all team members individually. The most significant contribution to teamwork is structured, collection documentation of the team’s knowledge, which all team members can access via a chatbot.

### 6 Summary and Outlook

The main goal of this paper was to introduce a framework that can be utilized as a team collaboration tool to retain information generated during internal communication and make it available to the whole team without the need for manual maintenance. This has been achieved through the introduction of the Agent component with a custom data extraction algorithm. The prototype has shown that automatic knowledge base extension through self-learning is possible.

However, the current algorithm is only capable of extracting data represented as Question & Answer tuples. It cannot represent data from multi-turn conversations and take user feedback (i.e., positive, negative) on answers into consideration. Future work will include improving the information extraction algorithm to gain new information from complex conversations and introduce capabilities to deal with the detection and binding of dynamic data into the extended knowledge base.

As early results have shown, the approach based in this paper shows a promising way to improve the support provided by conversational agents to teams, in our case to service teams.
References