

## RESEARCH ARTICLE

WILEY

# Toward an eclectic and malleable multiagent educational assistant

Aaron Briel 

Georgia Institute of Technology, Atlanta,  
Georgia, USA

**Correspondence**

Aaron Briel, Georgia Institute of  
Technology, Atlanta, GA 30332, USA.  
Email: [abriel3@gatech.edu](mailto:abriel3@gatech.edu)

**Abstract**

Conversational agents are systems capable of processing and responding to human language. They have evolved over the years from a means to pass the Turing Test to chatbots that fulfill a utilitarian purpose. Closed-domain chatbots are specialized in a specific knowledge base and are often used in an attempt to assist users in an educational context. Existing open-source, educational assistant chatbots are narrow in immediate functionality, thus limiting the content and services that can be provided to students and educators. To address this limitation, a novel multiagent framework is proposed, providing diverse capabilities and component flexibility to better meet varied educational requirements. The version presented in this experiment can not only answer questions in different styles but is also able to provide content summaries and links. The solution is tested by presenting participants with a lesson containing information related to coronavirus disease 2019 (COVID-19), followed by engagement with the chatbot system, a subsequent evaluation of its responses, and a quiz to quantify its pedagogical efficacy. COVID-19 was chosen as the knowledge base due to its current relevance in society. The chatbot framework's knowledge base is comprised of two data sets containing facts related to the virus, one which is used to provide longer, frequently asked questions-type responses, and another used to provide short answers. The resulting participant evaluations indicate a preference for more informative responses in the experimental context and showcase the benefit of the framework's malleability in not only fulfilling but discovering varying needs in educational assistance.

**KEYWORDS**

chatbot, COVID-19, educational Assistant, natural Language Processing, transformers

## 1 | INTRODUCTION

Conversational agents are systems capable of processing and responding to human language. They have evolved over the years from a means to pass the Turing Test to chatbots that fulfill a utilitarian purpose, such as customer support [8] or tutoring [37]. A distinction can be

made between open- and closed-domain agents. Open-domain variational are capable of conversation related to multiple subjects. Others are more task-oriented and are able to return information for a variety of requests.

Closed-domain instances are specialized for certain topics. Educational chatbots encapsulate this variation, as the agents' underlying knowledge bases are usually

domain-specific. For example, they have been developed to teach a foreign language [13] and increase the engagement of engineering students [7]. Closed-domain, educational chatbots often play the role of an assistant or Frequently Asked Questions (FAQ) portal. Instances like this can regularly be observed fulfilling a student support role on university websites, offering a range of services from administrative support [18] to being a full-fledged teaching assistant [15].

The conversational and task-oriented differentiation also exists within this closed-domain realm. Closed-domain chatbots are often limited in functionality, whether it be domain-specific conversation [2], simple question answering [30], or document retrieval [20]. Despite inherent limitations on their own, there exists an extensive variety of possible capabilities singular agents can offer. Aside from providing conversation, answers, and documents, certain chatbots can ask questions, fill a role of contact to an instructor or teaching assistant, and even give learning material recommendations [14].

A system that is able to execute all of these capabilities concurrently would clearly be superior to any of the single function agents on their own. Emergent, multifaceted educational assistant properties would be realized as the eclectic functionality of the system is expanded. Student needs, whether it be specific interaction styles that can better facilitate individual learning, different functions such as providing links or summaries, or content spanning various knowledge bases, would be better met by this broader range of capabilities. To fulfill these needs, we propose a multiagent chatbot framework with a focus on the diversity of chatbot capability not only in closed-domain knowledge but in the type of responses. User evaluations provide insight into the efficacy of the architecture and its underlying components. Although the experimental iteration of this framework consists of information related to coronavirus disease 2019 (COVID-19), the knowledge base is not limited to a single subject or even a single knowledge base. Instances could contain multiple knowledge bases containing information related to various subjects from Math to English that even span grades.

## 2 | RELATED WORK

The use of chatbots as educational assistants shows promise, particularly in scenarios demanding scalability, such as Massive Open Online Courses (MOOC). Agents have been demonstrated to improve the customization of individualized assignments [3], advance the learning process through voice-based peer to peer (P2P) assessments [27], and provide meaningful question answering

capability [19]. All of these assistant-based chatbot solutions free up the time of teaching assistants and instructors, which is an essential prerequisite to the fundamental MOOC objective of learning at scale.

Educational assistant chatbots are often restricted to a singular underlying architecture, whether it be rules-based logic to process student enquiries [35] or a transformers-based model for handling conversation [38]. Isolated chatbots may be able to provide satisfactory responses to queries specific to their specialization, but optimizing a solution requires the capability to address all necessary interactions [28]. For example, a system such as Lin and Chang's [21] thesis development assistant might be able to provide an answer to a question regarding the grammatical correctness of a sentence, but it will not be able to process subsequent requests for the overall feeling of the phrase, along with the link to a research paper discussing sentiment analysis with transformers.

In addition to lack of diversity in response type, a system's endemic technology may end up hindering its advancement. Chatbots strongly coupled to certain models like BERT [1] may be state-of-the-art for specific behaviors one day, only to become outdated as improved and more specialized models like MTSI-BERT [34] are released.

One possible solution to these limitations would be to present students with a chat system containing multiple isolated chatbot assistants with which they can communicate. A Wizard of Oz study by Chavez and Gerosa [6] in which users believed they were interacting with actual chatbots, however, found that this configuration may not be preferable. Increased confusion was reported among participants who believed they were engaged in a conversation with multiple chatbots as opposed to interacting with one.

A multiagent system capable of orchestrating the responses of multiple chatbots would address this issue. Makhkamova et al. [23] proposed a multichatbot broker where each agent is specialized with specific content, demonstrating better scores than baselines and "competing artefacts" on various metrics. The underlying chatbots, however, were homogeneous in architecture and thus response type, and a human evaluation of the system was absent from the study. Another solution following the multiagent approach demonstrated higher quality responses than those obtained from singular chatbot modules on their own [39]. Notable again was the absence of feedback from actual chatbot users and any emphasis on variability of responses. Neither solution was implemented specifically for use in education.

An eclectic, multichatbot system will benefit closed-domain users in the educational space by providing a

more seamless solution to their need for data that goes beyond simple domain-specific answers or even data limited to specific subjects or levels of education. Regarding the experimental approach, directly measuring the satisfaction of actual users against the multiagent system as well as with the underlying chatbots on their own, along with assessing their testing performance on domain-specific knowledge with or without a chatbot assistant, would contribute to quantification in a manner novel to multichatbot evaluations thus far, and highlight the benefits of framework malleability.

### 3 | NOVEL MULTIAGENT SOLUTION

We propose JuggleChat—a multiagent framework consisting of multiple underlying chatbot or non-chatbot components with varied functionality. This solution is primarily closed-domain, but not limited to a specific domain, with immediate application in an educational learning space. It needs to not only be capable of answering questions relevant to the learning material, but able to retrieve pertinent sections of texts, summarize content, provide URLs for further information, or execute any other functions deemed necessary in its deployment, including the ability to deliver content across subjects and grade levels. Its malleability will be evidenced by seamless support for state-of-the-art open-source chatbot solutions. The development of a multiagent framework was proven necessary to achieve this objective due to the singular-function nature of existing open-source chatbots and is novel to the educational technology space as far as the author is aware. In addition, the added behavioral capabilities of the system expand beyond the functional and knowledge base constraints of existing multiagent offerings.

#### 3.1 | Architecture

The JuggleChat framework contains a user interface, intent extraction module, chatbot allocator, and underlying chatbots. These components are illustrated in Figure 1.

User input is processed through a model specialized for intent extraction, which is detailed in Section 3.3. The intent and input are then passed to a chatbot allocator module that determines which chatbot to route the input to, based on said intent. The respective chatbot then processes the input and generates a response, which, in turn, is transmitted back to the user.

#### 3.2 | Datasets

The specialized, closed-domain knowledge base of the system in its instantiation for this experiment consists of questions and answers related to COVID-19 that were extracted from two sources. The first batch was pulled from a project that scraped COVID-19 data from 40 trusted websites, created question and answer pairs based on this data, and employed health care experts to determine the relevance of said pairs to unanswered questions [29]. For our purposes, pairs with a rating below 90 were filtered out and duplicates were removed. The second batch also contains COVID-19 questions and answers pulled from trusted sources, such as the World Health Organization and the Center for Disease Control, with direct website links for their sources [5,10]. In our case, outdated answers were updated where necessary by referencing the listed sources.

These refined datasets were then combined and shuffled to create a single FAQ-style data set of 1507 question and answer pairs. A second, Stanford Question Answering Dataset (SQuAD) [32] version was then created using deepset's online annotation tool [16]. SQuAD is a “reading comprehension” data set where crowdsourced users created over 100,000 questions mapped to short answers directly contained within chunks of text taken from a set of Wikipedia articles. In our SQuAD version, existing questions were mapped to answer segments from the prior FAQ-style document's full answers. These full answers were thus treated as the contexts. Numerical indices of the starting and ending points of these answer chunks were later used for metric calculations in an extractive-QA follow-up experiment detailed in Section 4.4. An example of SQuAD used in our experiment is illustrated in Figure 2.

The resulting knowledge base consisted of two primary datasets, one containing FAQ-style longer responses, and the other containing shorter answers in

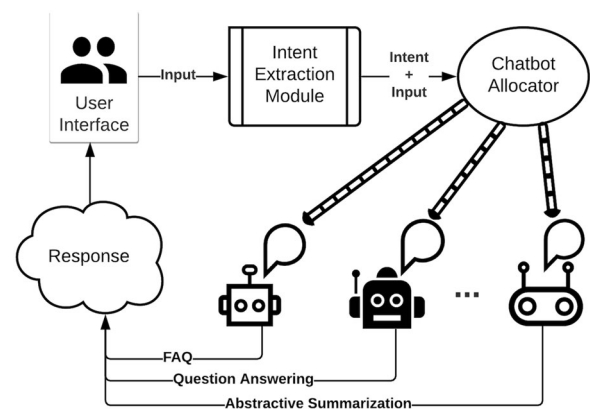


FIGURE 1 An overview of JuggleChat architecture

The spread can also be through **respiratory droplets** released when you sneeze, cough, or talk. Respiratory droplets may also be passed from person-to-person through commonly used surfaces such as door knobs, elevator buttons and bathroom faucets. COVID-19 is able to spread easily in the community, which is called "community spread". Socially distancing is a practice that aims to prevent those who are sick from coming into contact with those who are healthy. This is to reduce the opportunity for disease transmission. It means **avoiding crowds, mass gatherings, and maintaining at least a 6-foot distance from others**. This means no handshakes, hugs, or high-fives. It is also important to practice frequent hand-washing with soap and water or an alcohol-based (at least 60%) hand sanitizer.

How does Covid-19 spread?

**respiratory droplets**

How do I practice social distancing?

**avoiding crowds, mass gatherings, and maintaining at least a 6-foot distance from others**

**FIGURE 2** Pairs of questions and answers from a sample paragraph of the SQuAD data set. Answers are segments of text within the paragraph

SQuAD format. Subsets were created by evenly splitting the primary datasets in half and using one split for a partial FAQ-style set, and the other split for a partial SQuAD set, resulting in four final datasets. These partial and complete datasets were then used to train the underlying models.

### 3.3 | Models

The short-answer chatbot leverages a RoBERTa transformers model [22] called roberta-base-squad2 [12], which is specialized for SQuAD. This model was fine-tuned for the COVID-19 knowledge base by training it on the SQuAD-formatted version of the data set using hyperparameters specified by the authors of the second data source described previously [24]. One model was trained for JuggleChat using the partial SQuAD subset, and another was trained using the full SQuAD data set for an experimental group that would only use the extractive-QA or short-answer chatbot.

The FAQ-style version was used for the retrieval layer of the FAQ chatbot. The FAQ bot uses an embedding model called Sentence BERT (SBERT) [9,33], which is specialized for sentence embeddings. SBERT calculates vector similarity between user queries and those in its stored data set. Again, the partial FAQ subset was used for JuggleChat's underlying FAQ bot, while the full set was used for an FAQ-only experimental group.

The abstractive summarization module uses the Text-to-Text-Transfer-Transformer (T5) model [31]. No fine-tuning was necessary for this model as it relies on

transfer learning powerful enough to not warrant it in this case.

Questions were divided according to the data set split referenced earlier, with half mapped to a "qa" intent used for routing to the extractive-QA chatbot leveraging SQuAD, and the other half mapped to "faq," directing to the FAQ bot. Additional entries were added for summary requests with a corresponding "summary" intent. Entries for URL requests were added in a similar manner. This final intent data set was then used to train a ConveRT [17] intent extraction model. In this model, input and response texts are segmented into subwords, where frequently occurring words remain as-is, and less frequent ones are broken up into manageable chunks where possible. An example of this would be to break up the word "preprocessing" into subwords "pre," "process," and "ing." These subwords are then transformed into vectors containing positional encodings to capture context and subword embeddings. The subword embeddings consist of IDs that are computed using a hash function. Cosine similarity scores are then calculated for the final input and response vectors. Input texts in the model training consisted of the COVID-19 questions in the data set, with the responses being intents. The absence of duplicate questions across intents ensured sufficient differentiation in training.

### 3.4 | Components and data flow

The initial architecture for this study was assembled using open-source components and libraries. The user

interface for input processing and response delivery was constructed with React and TypeScript, with its backend consisting of an Express server communicating with a PostgreSQL database. A Rasa instance is used for the intent extraction module, leveraging the underlying ConveRT model detailed in Section 3.3. Along with “qa” and “faq,” intents were added for URL and summary requests, along with fallback intents able to handle questions regarding mood, whether the system is human, and greeting or departure statements. The extracted intent is then passed to a chatbot allocator class, which determines which underlying chatbot or component to route the input to.

Tasks themselves are managed by Celery with an underlying Redis datastore. These tasks encapsulate calls to an FAQ chatbot, a QA bot, and an abstractive summarization module. The FAQ and QA agents were developed using the Haystack framework [11]. Absum [4] is used for the abstractive summarization module which, in turn, leverages the “t5-small” variety of T5 open-sourced by HuggingFace [36]. Flower is used to expose the intent task as an API. Responses from chatbots are collected by corresponding Celery tasks that then route them back to the user through the API response.

More detailed visualization of the architecture is presented in Figure 3.

Of note is the malleable nature of the framework with respect to underlying components. Not only are all

chatbots and the intent extraction module swappable, but additional components can be seamlessly introduced. For example, a conversational chatbot component could be added that provides humanistic “chit-chat.” The intent extraction model would simply need to be trained with additional questions from the conversational data set mapped to a “conversation” intent, and a simple conditional would need to be added to the allocator.

## 4 | EXPERIMENT

### 4.1 | Purpose

There were two primary objectives of the experiment. The first was to determine whether JuggleChat is an effective tool for assisting students in learning. Presenting participants with a lesson followed by a quiz was an effort to answer this question and is detailed in the next section. The second objective involved discovering how participants felt about the chatbot interactions and how they perceived their accuracy and usefulness. This feature of the research model was influenced by the approach taken by Chavez and Gerosa in their Wizard of Oz experiment, specifically regarding the analysis of reported impressions [6]. In the context of this experiment, the evaluations would be used to quantify user preference for the style of content delivery.

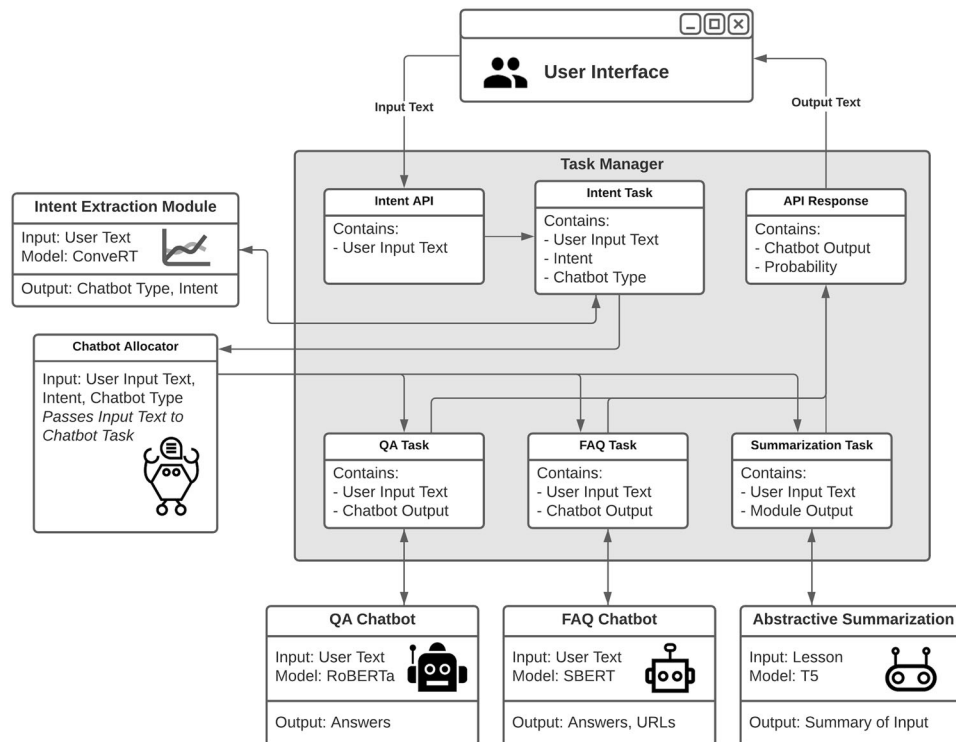


FIGURE 3 JuggleChat components and data flow



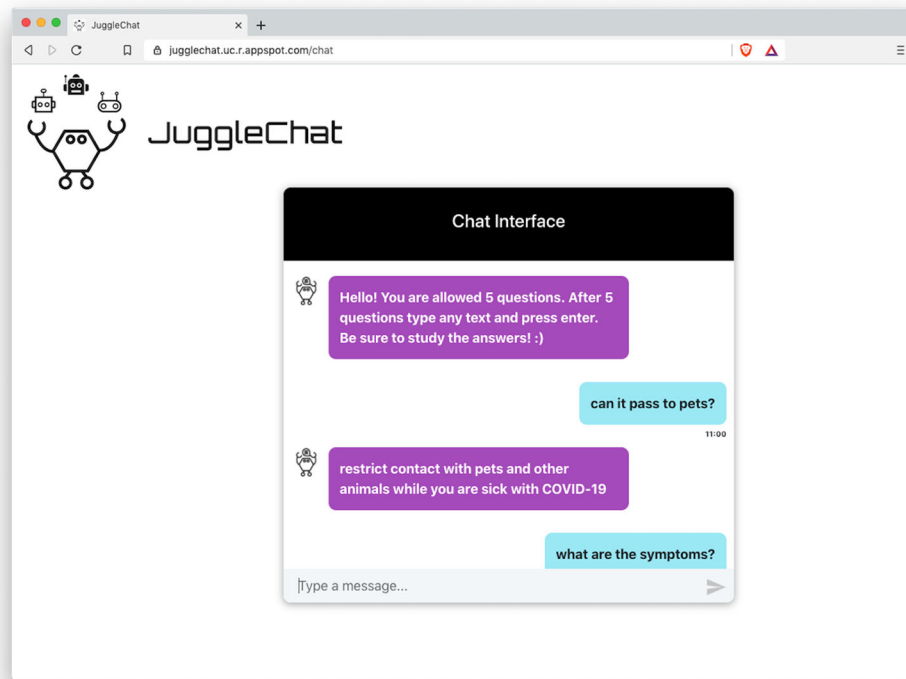


FIGURE 4 JuggleChat's chat interface in action

## 4.2 | Overview

Four groups of 100 participants each were used for evaluation of the framework, all blind to the underlying technology of the chatbot. One group used only the SQuAD-based, short answer bot, one used only the FAQ bot with longer responses, and a third used JuggleChat, which leveraged both chatbots along with a summarization module and the capability to provide URLs. A final group had no chatbot at all. The chatbot groups were allocated to obtain evaluations of the multiagent system along with the underlying chatbots on their own. The purpose of having a nonchatbot group was to assess whether access to the chatbots would affect testing performance.

Prolific Academic was chosen for the tool to run the experiment as it has been demonstrated to produce high-quality results [25,26]. Participants were first given a set of instructions indicating that they would be presented with a lesson containing information related to COVID-19. If they were in one of the three chatbot groups they were instructed that they would be asked to provide five questions to a chatbot, evaluate the chatbot interactions, then take a quiz at the end. JuggleChat participants were notified that one of their questions to the chatbot could be a request for a summary of the lesson. The primary reason for including the quiz outside of testing participant performance was to elicit high-quality

questions to ultimately obtain more realistic evaluations of the chatbots. The quiz questions were as follows:

1. What is the average incubation period?
2. How is COVID-19 primarily transmitted?
3. What underlying medical condition makes people at risk the least?
4. What is IPC?
5. What is the difference between COVID-19 and SARS-CoV-2?

To further influence participants to produce high-quality responses, they were told that they would start out with a certain payment amount, with the opportunity for a bonus payment if they produced authentic evaluations and questions. Instructions also warned them that they would lose their bonus opportunity if they browsed away from the final quiz or failed to complete it within 60 s. The quiz warnings were put in place in an attempt to prevent cheating.

In an effort to increase the variability of questions asked, participants were notified that chatbot responses along with quiz questions could contain information beyond that which was presented in the initial lesson. The participants that were not presented with a chatbot were simply given instructions that they would be quizzed, with a bonus possibility if they were able to complete it within the time limit and did not browse away.

Several paragraphs of general information related to COVID-19 were then displayed for the lesson. This content was pulled directly from the CDC website and contains an overview of the virus [5]. For the nonchatbot group, a “Ready for Quiz” button was available that routed them to the quiz when clicked. For chatbot participants, the chatbot interface became available after clicking a “Ready to Chat” button. The system’s chat interface is presented in Figure 4.

Participants were able to ask one question at a time. After a sequence of five questions and chatbot responses, an evaluation section was displayed containing the following:

- A free-form text field requesting a detailed, overall impression of the chat experience by the user in their own words.
- A rating scale of 1–10 on the perceived “accuracy” of the responses.
- A rating scale of 1–10 on the perceived “usefulness” of the responses.

The framework’s evaluation screen is visible in Figure 5.

The text responses containing overall impressions of the chat experience were run through the Huggingface sentiment-analysis pipeline to quantify participants’ attitude toward the chatbot—in particular, whether the

experience was positive or negative. These produced binary sentiment results for participants, with averages for each chatbot group calculated as the quotient of the number of positives and the total participant count per group.

Intents were also recorded for each question given by the JuggleChat participants to know exactly what underlying systems were allocated for responses within the framework.

Finally, users were given a multiple-choice quiz of five questions. As previously indicated, participants were notified that if they browsed away or did not complete the quiz in a 60 s time limit, they would not receive the bonus payment. The count-down timer was visible on the quiz page. Although chatbot participants were told that quiz questions may come from chatbot responses, the quiz itself only included questions with answers directly contained in the lesson presented to all groups. Deception was also used with respect to the bonus, in that bonus payments were given to all nonchatbot participants and to all chatbot users who asked authentic questions and gave quality evaluation feedback.

In summary, experimental variables thus included quiz score, sentiment, perceived accuracy and perceived usefulness. Quiz scores were recorded to not only demonstrate the efficacy of the system as a pedagogical tool but to increase the likelihood of

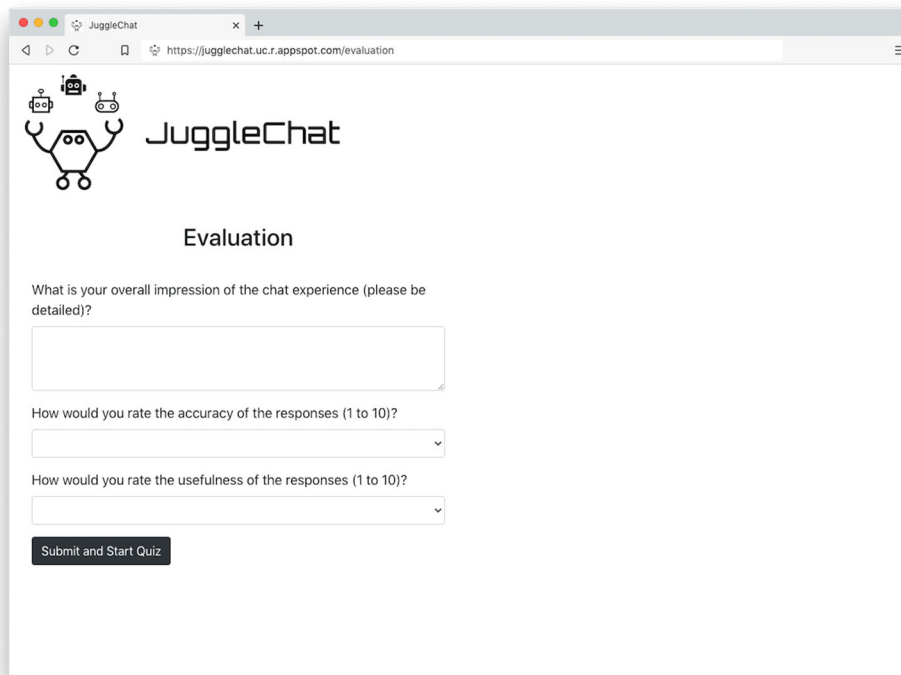


FIGURE 5 JuggleChat’s evaluation interface

receiving high-quality evaluations. Sentiment, perceived accuracy, and perceived usefulness were collected to discover insights into user preferences for the underlying chatbots.

### 4.3 | Results

One-way analysis of variance (ANOVA) tests were completed for all experimental groups against mean quiz scores, sentiment, perceived accuracy, and perceived usefulness, as the primary objective was determining statistical significance among each variable independently. Each ANOVA test started with the standard null hypothesis of equal means across groups and a significance level of  $\alpha = .05$ . For mean quiz scores, the resulting  $p$  value of .5034 indicated that the differences across all test groups were not statistically significant. The  $p$  values of .0009, 1.9685e-09, and 4.0952e-07 for sentiment, perceived accuracy, and perceived usefulness, however, warranted the completion of posthoc tests for each of these metrics.

Tukey's Honestly Significant Difference (HSD) is an effective posthoc testing tool for determining whether the mean differences observed in sets of data are statistically significant and where those differences lie. This allows one to discover possible relationships in the said sets of data. In our case, we wished to determine whether there were any relationships between sentiment, perceived accuracy and perceived usefulness. These posthoc HSD tests were conducted with a standard family-wise error rate of 0.05. The simultaneous confidence intervals for sentiment can be seen in Figure 6.

A statistically significant difference can be observed between FAQ and extractive-QA, indicating that participants preferred the FAQ-style responses over those provided by the extractive-QA chatbot. JuggleChat's confidence interval placement is likely the result of its marginally larger number of FAQ allocations (236 vs. 228 for QA) as opposed to the 13 summary requests, as the subset of users with Absum allocations showed a slightly smaller mean sentiment of 0.23.

A similar pattern of preference was also observed in the confidence intervals for perceived accuracy and usefulness, as visualized in Figures 7 and 8.

Statistically significant differences in perceived accuracy were observed between all groups, with the FAQ bot being rated highest, followed by JuggleChat and then QA. With respect to perceived usefulness, a significant difference was again displayed between FAQ and extractive-QA with JuggleChat lying in the middle once more.

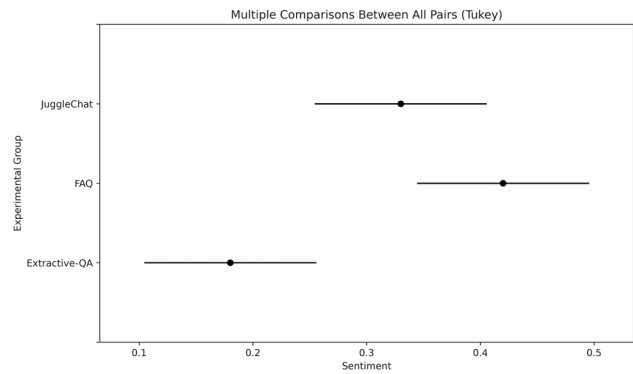


FIGURE 6 Tukey HSD for sentiment

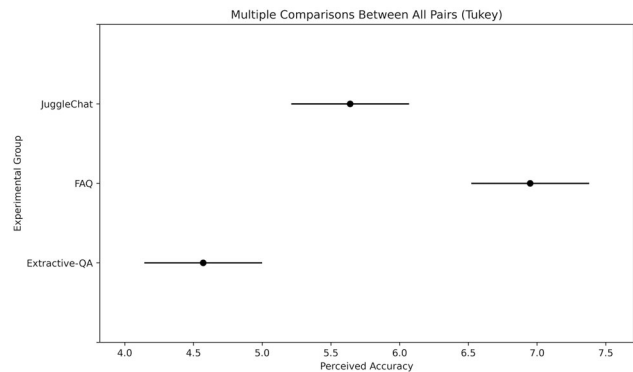


FIGURE 7 Tukey HSD for perceived accuracy

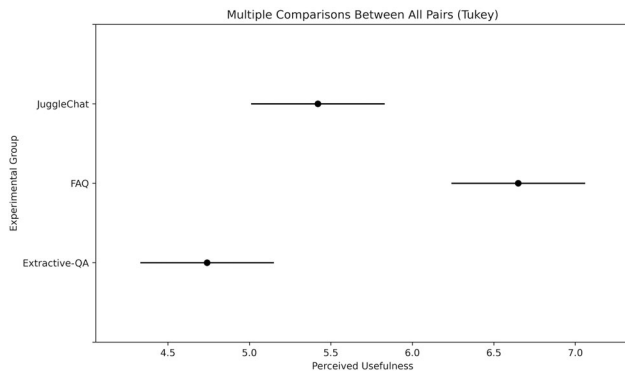
### 4.4 | Extractive-QA comparison

In an attempt to gain further insight into the lower perceived quality of the extractive-QA responses, a performance comparison was completed with our COVID-19 data set against Möller's, using the same script he used in his study. The data we used in our attempt at replication is deepset's SQuAD-based COVID-QA data set, not to be confused with the FAQ-style subset that we integrated into our own.

Exact Match was calculated as the frequency with which the starting and ending indices for predicted answers exactly matched those of the test set labels. F1 was computed as a mean of the maximum  $F$  score of each prediction against each label, with  $F$  score calculated according to the following formula, where precision is the quotient of the number of correctly overlapping starting and ending indices between the predictions and the labels and the number of predictions, and recall being the quotient of that number of correctly overlapping indices and the number of test labels:  $F \text{ score} = (2 \times \text{precision} \times \text{recall}) / (\text{precision} + \text{recall})$ .

We were able to closely approximate Möller's results for Exact Match and F1 with his data set while





**FIGURE 8** Tukey HSD for perceived usefulness

**TABLE 1** Performance comparison of COVID-19 datasets

Data set	Exact match	F1
deepset Covid-QA	26.15	58.59
JuggleChat Covid-QA	21.55	44.32

demonstrating a significantly lower F1 score for ours, as Table 1 can attest.

The low scores overall could be related to the complexity of the question/answer pairs combined with the absence of multiple annotations per question [24]. With respect to the latter, in the case of our data set the presence of multiple similar questions pointing to the same single answer might have also contributed.

Direct analysis of chat behavior produced by deepset's COVID-QA model indicated performance unsuitable for user evaluation in the context of our experiment as it had no responses to basic COVID-19 questions. This was not entirely unexpected, as its underlying data set consists of question/answer pairs extracted directly from scientific articles related to the virus, as opposed to containing general questions and answers as in the case of ours.

## 5 | CONCLUSIONS AND LIMITATIONS

The experimental results demonstrated participant preference for FAQ-style responses in the given context. Other learning scenarios might have demonstrated a preference for different chatbot styles. This emphasizes the benefit of JuggleChat's eclectic capability, which not only provides various functionality to meet diverse student needs but also allows for the discovery of what those needs might be.

Participants clearly rated the accuracy and usefulness of the extractive-QA model lowest. This points to a need for more clarity regarding data set engineering for

optimal performance in extractive-QA models. Researchers developing multichatbot systems would benefit by keeping this shortcoming salient in their consideration of underlying chatbot components. Perceived accuracy and usefulness with respect to JuggleChat and FAQ in particular, however, suggest that quiz performance may not be the sole demonstrator of pedagogical efficacy. The benefits of JuggleChat itself could more aptly be demonstrated in a context that better emphasizes its eclectic functionality. User behavior and evaluation from beta deployments with client-specific functions and knowledge bases will likely provide more appropriate insights as emphasized in Section 6.

The conceptualization of optimal chatbot performance would strongly benefit from an expansion beyond the constraints of standard machine learning metrics. As evidenced in Section 4.4, one data set may result in higher F1 scores but is unable to provide responses to basic queries related to the topic purported to be its specialization, emphasizing that these metrics should not replace human evaluation for tools that will eventually reach an educational setting.

## 6 | FEASIBILITY AND PRACTICALITY OF JUGGLECHAT

As the spread of COVID-19 made online learning a necessity, the need for educator assistance became immediately evident. The clear advantages of JuggleChat that highlight its practicality in this scenario are its component flexibility to address the needs of students and educators and its deployment scalability. Its feasibility is emphasized by the simplicity of integrating domain knowledge specific to the requirements of the instance. Introduction of knowledge bases, such as math or history, in a multisubject institution, such as high schools, involves a simple, one-time data-entry process. A given deployment can then be leveraged across schools or even school districts that share curriculums. For MOOCs, a single instantiation could contain multiple knowledge bases that span a diverse set of online courses. The evolution of the framework with respect to underlying components can proceed according to the measurement of student preference or component engagement.

JuggleChat's feasibility is also evident when considering the practicality of its maintenance. The process of updating a knowledge base is a simple matter of adding or updating data entries in the training file and then running the training command to update the model. The ease with which underlying components can be interchanged allows for upgrade flexibility to ensure that the instance remains state of the art. Once deployed, the

framework is limited in student capacity only by the constraints of its underlying hardware, realizing its fundamental objectives of freeing up the valuable time of human instructors and ultimately helping students learn.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in the JuggleChat repository at <https://github.com/aaronbriel/jugglechat>

## ORCID

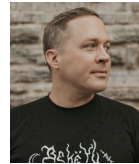
Aaron Briel  <http://orcid.org/0000-0002-8189-7774>

## REFERENCES

1. R. Bathija, P. Agarwal, R. Somanna, and G. B. Pallavi. 2020. *Guided interactive learning through chatbot using bi-directional encoder representations from transformers (BERT)*. 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), pp. 82–87. <https://doi.org/10.1109/ICIMIA48430.2020.9074905>
2. T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol. 2017. *Rasa: Open source language understanding and dialogue management*. arXiv:1712.05181 [cs]. <http://arxiv.org/abs/1712.05181>
3. L. Bollweg, M. Kurzke, K. M. Asif Shahriar, and P. Weber. 2018. *When robots talk—Improving the scalability of practical assignments in MOOCs using chatbots*. pp. 1455–1464. <https://www.learntechlib.org/primary/p/184365/>
4. A. Briel. 2020. *Abstractive summarization for data augmentation*. Medium. <https://towardsdatascience.com/abstractive-summarization-for-data-augmentation-1423d8ec079e>
5. CDC. 2020. *Coronavirus disease 2019 (COVID-19)*. Centers for Disease Control and Prevention. <https://www.cdc.gov/coronavirus/2019-ncov/hcp/non-us-settings/overview/index.html> and <https://www.cdc.gov/coronavirus/2019-ncov/hcp/faq.html>
6. A. P. Chaves and M. A. Gerosa. 2018. *Single or multiple conversational agents? An Interactional Coherence Comparison*. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp. 1–13. <https://doi.org/10.1145/3173574.3173765>
7. S. Crown, A. Fuentes, R. Jones, R. Nambiar & D. Crown. 2010. *Ann G. Neering: Interactive chatbot to motivate and engage engineering students*. <https://doi.org/10.18260/1-2-16687>
8. L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou. 2017. *SuperAgent: A customer service chatbot for e-commerce websites*. Proceedings of ACL 2017, System Demonstrations, 97–102. <https://www.aclweb.org/anthology/P17-4017>
9. deepset/sentence\_bert. *Hugging face*. [https://huggingface.co/deepset/sentence\\_bert](https://huggingface.co/deepset/sentence_bert)
10. deepset-ai/COVID-QA. *deepset*. <https://github.com/deepset-ai/COVID-QA>
11. deepset-ai/haystack. *deepset*. <https://github.com/deepset-ai/haystack>
12. deepset-ai/roberta-base-squad2. *Hugging face*. <https://huggingface.co/deepset/roberta-base-squad2>
13. L. K. Fryer, K. Nakao, and A. Thompson, *Chatbot learning partners: Connecting learning experiences, interest and competence*, Comput. Human. Behav. **93** (2019), 279–289. <https://doi.org/10.1016/j.chb.2018.12.023>
14. A.-A. Georgescu. 2018. *Chatbots for education—Trends, benefits and challenges*. Conference Proceedings of eLearning and Software for Education (eLSE) 2, 14: pp. 195–200. <https://doi.org/10.12753/2066-026X-18-097>
15. A. Goel. 2020. *AI-powered learning: Making education accessible, affordable, and achievable*. arXiv:2006.01908 [cs]. <http://arxiv.org/abs/2006.01908>
16. Haystack Annotation Tool. <https://annotate.deepset.ai>
17. M. Henderson, I. Casanueva, N. Mrkšić, P.-H. Su, T.-H. Wen, and I. Vulić. 2020. *ConveRT: Efficient and accurate conversational representations from transformers*. arXiv:1911.03688 [cs]. <http://arxiv.org/abs/1911.03688>
18. H. T. Hien, P.-N. Cuong, L. N. H. Nam, H. L. T. K. Nhung, and L. D. Thang. 2018. *Intelligent assistants in higher-education environments: The FIT-EBot, a chatbot for administrative and learning support*. Proceedings of the Ninth International Symposium on Information and Communication Technology (SoICT 2018), pp. 69–76. <https://doi.org/10.1145/3287921.3287937>
19. H.-H. Hsu and N.-F. Huang. 2018. *Xiao-Shih: The educational intelligent question answering bot on Chinese-based MOOCs*. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1316–1321. <https://doi.org/10.1109/ICMLA.2018.00213>
20. V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W-tau Yih. 2020. *Dense passage retrieval for open-domain question answering*. arXiv:2004.04906 [cs]. <http://arxiv.org/abs/2004.04906>
21. M. P.-C. Lin and D. Chang, *Enhancing post-secondary writers' writing skills with a chatbot: A mixed-method classroom study*, Journal of Educational Technology & Society **23** (2020), no. 1, 78–92. <https://doi.org/10.2307/26915408>
22. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. *RoBERTa: A robustly optimized BERT pretraining approach*. arXiv:1907.11692 [cs]. <http://arxiv.org/abs/1907.11692>
23. O. Makhkamova, K.-H. Lee, K. H. Do, and D. Kim. 2020. *Deep learning-based multi-chatbot broker for Q a improvement of video tutoring assistant*. 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 221–224. <https://doi.org/10.1109/BigComp48618.2020.00-71>
24. T. Möller, A. Reina, R. Jayakumar, and M. Pietsch. 2020. *COVID-QA: A question answering dataset for COVID-19*. <https://openreview.net/forum?id=JENSKEEzsoU>
25. S. Palan and C. Schitter, *Prolific.ac—A subject pool for online experiments*, J. Behav. Exp. Finance **17** (2018), 22–27. <https://doi.org/10.1016/j.jbef.2017.12.004>
26. E. Peer, L. Brandimarte, S. Samat, and A. Acquisti, *Beyond the Turk: Alternative platforms for crowdsourcing behavioral research*, J. Exp. Soc. Psychol. **70** (2017), 153–163. <https://doi.org/10.1016/j.jesp.2017.01.006>
27. J. Pereira, M. Fernández-Raga, S. Osuna-Acedo, M. Roura-Redondo, O. Almazán-López, and A. Buldón-Olalla, *Promoting Learners' Voice Productions Using Chatbots as a Tool for Improving the Learning Process in a MOOC*, Tech. Knowl.

- Learn. **24** (2019), no. 4, 545–565. <https://doi.org/10.1007/s10758-019-09414-9>
28. C. S. Pinhanez, H. Candello, M. C. Pichiliani, M. Vasconcelos, M. Guerra, M. G. de Bayser, and P. Cavalin. 2018. *Different but equal: Comparing user collaboration with digital personal assistants vs. teams of expert agents*. arXiv:1808.08157 [cs]. <http://arxiv.org/abs/1808.08157>
  29. A. Poliak, M. Fleming, C. Costello, K. W. Murray, M. Yarmohammadi, S. Pandya, D. Irani, M. Agarwal, U. Sharma, S. Sun, N. Ivanov, L. Shang, K. Srinivasan, S. Lee, X. Han, S. Agarwal, and J. Sedoc. 2020. *Collecting verified COVID-19 question answer Pairs*. Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020. <https://www.aclweb.org/anthology/2020.nlpCOVID19-2.31>
  30. S. Quarteroni & S. Manandhar 2007. A chatbot-based interactive question answering system. Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pp. 83–90. <http://www-users.cs.york.ac.uk/%7Eesuresh/papers/ACIQAS.pdf>
  31. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. 2020. *Exploring the limits of transfer learning with a unified text-to-text transformer*. arXiv:1910.10683 [cs, stat]. <http://arxiv.org/abs/1910.10683>
  32. P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. *SQuAD: 100,000+ questions for machine comprehension of text*. arXiv:1606.05250 [cs]. <http://arxiv.org/abs/1606.05250>
  33. N. Reimers and I. Gurevych. 2019. *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
  34. M. A. Senese, G. Rizzo, M. Dragoni, and M. Morisio. 2020. *MTSI-BERT: A session-aware knowledge-based conversational agent*. Proceedings of The 12th Language Resources and Evaluation Conference, pp. 717–725. <https://www.aclweb.org/anthology/2020.lrec-1.90>
  35. J. Singh, M. H. Joesph, and K. B. Abdul Jabbar, *Rule-based chatbot for student enquiries*, J. Phys.: Conf. Ser. **1228** (2019), 12060. <https://doi.org/10.1088/1742-6596/1228/1/012060>
  36. t5-small. *Hugging face*. <https://huggingface.co/t5-small>
  37. Y. F. Wang and S. Petrina, *Using learning analytics to understand the design of an intelligent language tutor—Chatbot Lucy*, Int. J. Adv. Comput. Sci. Appl. **4** (2013), 11. <https://doi.org/10.14569/IJACSA.2013.041117>
  38. Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan. 2020. *DialoGPT: Large-scale generative pre-training for conversational response generation*. arXiv:1911.00536 [cs]. <http://arxiv.org/abs/1911.00536>
  39. J. F. Zolitschka. 2020. *A novel multi-agent-based chatbot approach to orchestrate conversational assistants*. Business Information Systems (Lecture Notes in Business Information Processing), pp. 103–117. [https://doi.org/10.1007/978-3-030-53337-3\\_8](https://doi.org/10.1007/978-3-030-53337-3_8)

## AUTHOR BIOGRAPHY



Aaron Briel received his BA in Psychology and BS in Computer Science from the University of Minnesota. He has over 20 years of experience in development, automation, cybersecurity, and Machine Learning. Data produced by an application he wrote for a Fortune 100 company was presented in Congressional hearings to demonstrate post-breach compliance. His work in Artificial Intelligence and Machine Learning started at the University of Minnesota, where he developed a Genetic Algorithm capable of generating hip-hop drum sequences. Aaron is currently a Senior Machine Learning Engineer at Pearson and is pursuing his MS in Computer Science at Georgia Institute of Technology. His primary professional and research interests include Natural Language Processing, Deep Learning, and Machine Learning infrastructure.

**How to cite this article:** A. Briel, *Toward an eclectic and malleable multiagent educational assistant*, Comput. Appl. Eng. Educ. (2021), 1–11. <https://doi.org/10.1002/cae.22449>