Journal of Information Systems Applied Research

Volume 17, No. 2
August 2024
ISSN: 1946-1836

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Journal of Information Systems Applied Research** (JISAR) is a double-blind peer reviewed academic journal published by ISCAP, Information Systems and Computing Academic Professionals. Publishing frequency is three issues a year. The first date of publication was December 1, 2008.

JISAR is published online (https://jisar.org) in connection with the ISCAP (Information Systems and Computing Academic Professionals) Conference, where submissions are also double-blind peer reviewed. Our sister publication, the Proceedings of the ISCAP Conference, features all papers, teaching cases and abstracts from the conference. (https://iscap.us/proceedings)

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%) and other submitted works. The non-award winning papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in JISAR. Currently the acceptance rate for the journal is approximately 35%.

Questions should be addressed to the editor at editor@jisar.org or the publisher at publisher@jisar.org. Special thanks to members of ISCAP who perform the editorial and review processes for JISAR.

Journal of Information Systems Applied Research

An Action Research Approach to Building an Enterprise-Specific Chatbot (ESCB)

Zach Wood
zkw8126@uncw.edu

Geoff Stoker
stokerg@uncw.edu

Congdon School
University of North Carolina Wilmington
Wilmington, NC 28403 USA

Abstract

Organizations are increasingly turning to chatbots to provide customer support via computer-generated, conversational, natural language answers to human queries. This paper describes a technique for creating an enterprise-specific chatbot (ESCB). We conducted an action research study to investigate the possibility of creating an ESCB with a local policy document knowledge base using readily available software tools, a basic level of programming competence, and user community feedback. The applied research on this chatbot leverages the power of Artificial Intelligence (AI), Natural Language Processing (NLP), and proprietary local data to transcend the typical limitations of conventional chatbots. Utilizing three quick-turn action research cycles, we evolved the chatbot to demonstrate high accuracy and relevance in its responses. The results indicate that our chatbot is becoming increasingly efficient in interpreting user queries, extracting necessary information, and formulating appropriate responses. The work underscores the significant potential of AI-powered chatbots for data interaction and the affordability of AI implementation, paving the way for organizations with limited resources to leverage the power of AI in their local operations.

Keywords: Chatbot, Action Research, Proprietary Data, Large Language Model, AI, Semantic Search

Recommended Citation: Wood E., Stoker, G., (2024). An Action Research Approach to Building an Enterprise-Specific Chatbot (ESCB). *Journal of Information Systems Applied Research.* 17(2), pp 61-73. https://doi.org/10.62273/RAON2946

An Action Research Approach to Building an Enterprise-Specific Chatbot (ESCB)

Zach Wood and Geoff Stoker

1. INTRODUCTION

In the 7th decade of the Information Age, it appears that the precept characterized by the famous adage "knowledge is power" (Bartleby, 2023) is evolving to one perhaps better characterized as *applied knowledge is power*. With the exponential growth in data generation, collection, and availability across most domains, organizations are inundated with information. While having access to this data can be crucial, a significant challenge lies in understanding, managing, and effectively using this information for impactful outcomes (Patel & Trivedi, 2020). This is where Artificial Intelligence (AI) and Natural Language Processing (NLP) can come into play, specifically in the form of enterprise-specific chatbots (ESCB).

Chatbots have increasingly become indispensable customer service tools across industries, from commerce to financial services to healthcare. They are designed to improve service delivery, enhance operational efficiency, and offer personalized assistance in an automated yet human-like manner. However, the traditional chatbot implementation can fall short in adaptability and flexibility, particularly when handling local information and dealing with the varied phrasings of user queries (Nuruzzaman & Hussain, 2018).

Commercial website chatbots often adhere to a rigid question-answer pathway, enabling them to process inquiries only within predefined frameworks set by the company. This limited flexibility can result in less-than-optimal user experiences, with chatbots being unable to process or respond to queries outside their programmed parameters effectively (Ayanouz et al., 2020).

Data today can be as diverse as it is abundant, necessitating the ability to effectively harness and process disparate information. Organizations often possess large volumes of proprietary data, within which lie valuable insights regarding organization-specific knowledge. However, this data can be underutilized due to the challenges of integrating and processing it. This situation requires tools, techniques, and procedures to effectively navigate this data, providing relevant insights and answers, irrespective of query complexity (Ayanouz et al., 2020).

Motivated by a desire to find answers to local organization policy questions more efficiently, we present our action research of the design and development of an ESCB capable of effectively querying local data and interpreting questions posed in various ways. By harnessing the power of AI and NLP to curate proprietary data, we demonstrate an approach for enterprises to unlock the potential of their organization-specific information. This approach can empower users who possess a basic competence with technology, regardless of parent organization, to engage with and interpret complex information, thereby transcending some of the limitations of conventional chatbots.

Our paper outlines using NLP techniques to provide enterprises with an efficient and cost-effective method to query their data. This can potentially avoid the prohibitive time and financial investment often required to train a Large Language Model (LLM) on proprietary datasets (Kalla & Smith, 2023). The advantages of our proposed approach include:

1. Cost-efficiency and time-savings by eliminating extensive training requirements
2. Immediate updates to the underlying knowledge base, allowing for real-time data interaction
3. The ability to pose abstract queries stemming from a wide range of knowledge backgrounds by leveraging an existing LLM
4. Allowing customer service representatives to dedicate their efforts more effectively by automating responses to simple queries

We utilized OpenAI's Generative Pre-Trained Transformer (GPT) LLM through its Application Programming Interface (API), which serves as an intelligent filter. This enabled us to focus on the applications of our proposal rather than the process of creating an entirely new system.

With regular advancements in the LLM models that OpenAI offers for public use, we can accomplish more with less (Kalla & Smith, 2023). Our approach emphasizes the importance of only sending the necessary proprietary data to the LLM. By asking the LLM to provide an answer based on a specific chunk of text, we can respond to questions phrased in various ways, including those that the chatbot is not explicitly programmed to answer. This methodology leverages the advanced understanding of the LLM and paves the way for more efficient and intuitive data querying.

The remainder of this paper is organized as follows: Section 2 provides a literature review; Section 3 describes the action research method we followed as well as the major components of the ESCB; in Section 4, we present some results; Section 5 discusses the implications of the results; and Section 6 concludes.

2. LITERATURE REVIEW

Though the precursor for the term Chatbot dates to ~1994 when "ChatterBot" was coined (Mauldin, 1994), computer programs trying to interact with humans via natural language pre-date the term by several decades. ELIZA (Weizenbaum, 1966), probably the best-known early example, was intended as a project to explore how humans and computer programs might communicate. ELIZA (e.g., https://www.masswerk.at/elizabot/) used a typing interface comment-response process to imitate the reflection techniques used by Rogerian psychotherapists (Landsteiner, 2005) and occasionally fooled people into believing they were communicating with a human therapist.

In the decades after ELIZA's arrival, the field of NLP developed unevenly but gave rise to increasingly sophisticated chatbots. Some more recent iterations of chatbots, like Mitsuku/Kuki ("Hi, I'm Kuki," 2023), Cleverbot ("Cleverbot," 2023), and IBM Watson ("IBM Watson," 2023), leverage advances in AI and machine learning (ML) to engage in more complex and human-like conversations (Nuruzzaman & Hussain, 2018).

The development of NLP Libraries such as spaCy (Partalidou et al., 2019) has been instrumental in the evolution of chatbots. SpaCy, for instance, provides functionalities like part-of-speech tagging, entity recognition, and dependency parsing, which are crucial for understanding and processing human language. This has significantly enhanced the ability of chatbots to understand and generate responses to user queries.

Semantic searching has also been increasingly utilized in the development of chatbots. Semantic searching enhances the chatbot's ability to understand user queries by considering the context and intent behind the words rather than just the literal meanings. This results in more accurate and relevant responses (Wei et al., 2008). A key semantic innovation demonstrated in 2013 (Mikolov et al.) was that vector representations of words (also called word embeddings), learned by a recurrent neural network language model, captured relationships to other words that allowed NLP applications to distinguish subtle semantic connections, including the now relatively famous example:

v("King") – v("Man") + v("Woman") ≈ v("Queen")

Finally, the advent of AI models like GPT-3 by OpenAI ("GPT-3", 2023) has revolutionized the field of chatbots. These transformer-based models can generate human-like text by predicting the likelihood of a word given the previous words used in the text. They are trained on a diverse range of internet text but cannot explicitly recall or understand information due to their generative nature. They can generate creative, contextually relevant responses, making them ideal for use in chatbots (Nath et al., 2022).

This paper explores ESCB development by combining the capabilities of the spaCy library, semantic searching using scikit-learn (Pedregosa et al., 2011), and the GPT-3.5 model by OpenAI. This approach aims to leverage the strengths of these technologies to create a chatbot that can generate more accurate, contextually relevant, human-like answers to enterprise-specific queries.

3. METHODOLOGY

This paper explores the possibility of constructing an ESCB with readily available tools, basic programming competence, and user community feedback. As we are "addressing questions in one's immediate work environment, with the goal of solving an ongoing problem in that environment" (Leedy & Ormrod, 2010, p. 44), we take an action research approach in this investigation. We follow the canonical action research process model (Susman & Evered, 1978) in Figure 1 (Davison et al., 2004), making three quick turns to refine the development technique and evolve the ESCB. Following this model helps ensure systematic rigor is applied to the problem. Steps include:

- Diagnosis – conduct a thorough examination of the current organizational circumstances
- Planning – the diagnosis results directly

inform all planning; intended actions should be specified before being undertaken

- Action – planned actions are implemented in the order specified (if any)
- Evaluation – once planned actions are complete, outcomes are compared to project objectives and expectations
- Reflection – explicitly reflect on the activities taken and the outcomes achieved; decide whether to exit the cycle or iterate

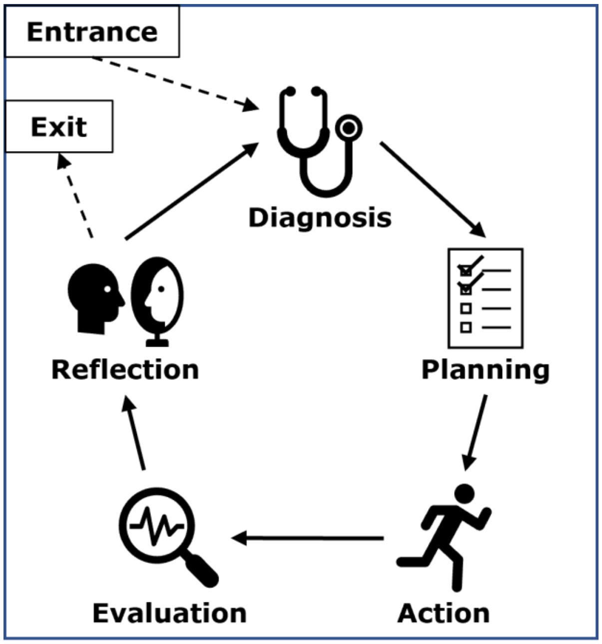

Figure 1: A canonical action research process model (Fig. 1. Davison et al., 2004)

Diagnosis and Planning

In our local organization, finding answers to common enterprise-specific questions most often involves searching for and reading through static web pages, linked portable document format (PDF) policy files, and/or knowledge base articles. It seemed that an ESCB could provide a more dynamic customer-service-oriented user experience. We hypothesized that the state and availability of current AI/NLP tools had evolved to the point that a functional ESCB could be constructed with these tools and the locally available programming talent.

To explore ESCB development, we planned to iterate through cycles that involved tool evaluation, coding, user interaction/feedback, and explicit results reflection to see what we had learned about the evolving chatbot, and the process involved in its development. Our goal was to outline a technique that others interested in creating an ESCB could follow for their own organizational-specific use.

We started with a high-level conceptual sketch of the expected business process for user-chatbot data exchange and refined it as we iterated through the action research cycle. The final conceptual form of the process is depicted in Figure 2. Key components of the process are shown in rounded rectangles at the top, the order of significant steps is listed top-to-bottom, and arrows indicate information flow/exchange between key components at each significant step. Details related to Figure 2 are provided later in the paper. They will necessarily differ somewhat from the simplifying, high-level conceptual sketch, which abstracted away some of the more complicated aspects of the process.

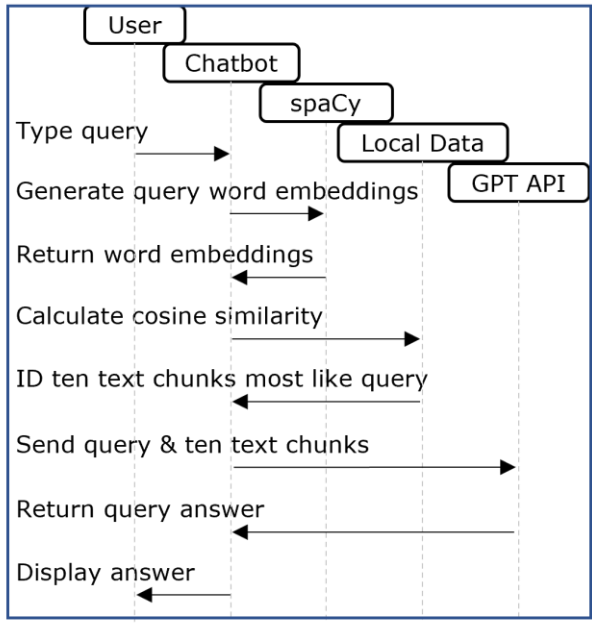

Figure 2: A logical depiction of significant steps in user-ESCB information exchange (top-to-bottom) and key components involved

Action – spaCy and Word Embeddings

Words need to be converted to vectors of numbers to make use of the text in a user query and the text contained in the corpus of local policy documents (CLPD). After a user submits a query, the ESCB loads a medium-sized English-language model from spaCy. This model has been pre-trained on a massive text dataset and has ready-to-use vectors corresponding to thousands of words. To process the user query, the text is first broken into component words or word parts (i.e., tokenized – see Figure 3). The individual words (tokens) are then matched and assigned vectors from the spaCy model. A centroid vector representing the entire query is then calculated from the vectors of the individual words. One (1) token is equivalent to about four (4) characters in English, while 100 tokens are roughly equivalent to 75 words (Raf, n.d.b).

To process the CLPD, each document is first broken into text chunks of a specified number of tokens (we first used 100, then later 200 tokens), and then each text chunk is transformed into a single vector in the same manner as the user query. After this step, word embeddings exist for the user query and for every text chunk in the CLPD.

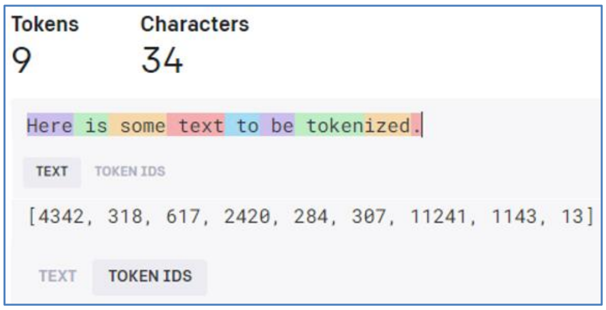

Figure 3: An example of text tokenization (https://platform.openai.com/tokenizer).

Action – scikit-learn and Cosine Similarity
With the user query converted to a single vector and the CLPD converted to many vectors, semantic search can begin. This is done by using scikit-learn to calculate the cosine similarity between the embeddings of the user query and each of the embeddings for the text chunks from the CLPD. Cosine similarity values range from zero (0) to one (1), with zero being no relation and one being a perfect match. Our system sorts the results in ascending order and then selects the 10 chunks with the highest cosine similarity scores. These chunks are the ones that are most semantically like the user query according to the spaCy language model. Thus, at the end of this process, the ESCB has the portions of the CLPD most relevant to the user query and most likely to contain the text that can answer the user's query.

Action – GPT API
With retrieval of the most semantically relevant information from the CLPD complete, we next integrate the GPT API. We bundle together the original plaintext of the user query, the 10 plaintext chunks resulting from the previous step, and the plaintext from the chatbot session history (if any) and make a call to the GPT API. Including the session history enhances the chatbot's ability to understand and respond to user queries in various ways due to the ability to expand on previously answered queries. After the GPT API returns an answer, it is displayed for the user.

Evaluation
To evaluate the ESCB, we leveraged three different opportunities across the three quick-turn, action research cycles to have users interact with and provide feedback on the chatbot. The first evaluation opportunity was at a local annual Information and Technology Exchange (ITX) event hosted at our university. We set up a table and engaged with the student, faculty, staff, and local IT professional attendees. The second evaluation opportunity was during project presentations in a 400-level course. The third chance to get feedback on the ESCB was at a semi-annual computing showcase featuring student and faculty projects. Across the three presentation opportunities, there were 50 people exposed to the ESCB who either directly interacted with the chatbot or provided feedback on its observed performance. Additional details regarding evaluation and feedback will be provided later.

Reflection
While the reflection phase of action research is enumerated last, it is really an ongoing process. Throughout this applied research activity, we endeavored to use a deliberate reflection activity to both consider the success of the technique for developing an ESCB and to determine whether to proceed with an additional action research cycle. These reflection activities led us to conduct three quick turns through the action research cycle before concluding the ESCB applied research.

4. RESULTS

ESCB Development and Deployment
At the end of our third action research cycle, we had a quite capable ESCB built by a local programmer using readily available AI tools and drawing on information from a 194-PDF-document CLPD. Local community user feedback guided ESCB development, and the chatbot was ultimately able to respond with a reasonable degree of accuracy to questions regarding local policies.

The ESCB was built with a React JS front end and a Python-Flask back end with code organized, as depicted in Figure 4. Key parts include:

- *DocRWidget.jsx* is the front-end parent file.
- *file_read.py* reads the 194 PDF policy docs.
- *chunk_check.py* uses the spaCy language model to create word embeddings for the

user query and 200-token-long chunks of the policy docs, then uses *scikit-learn* to calculate Cosine similarity scores.
- *openai_call.py* interfaces with the user and formats the *text-davinci-003* GPT API call.

Code details available upon request

```
(Project Directory Map)
/expo-demo
├── flask-server
│   ├── data (local database)
│   │   └── All PDF Policy Docs
│   └── venv (virtual environment)
│       ├── openai_call.py
│       ├── chunk_check.py
│       ├── file_read.py
│       ├── preprocessed_data.pkl
│       └── server.py (front-
end connection)
├── frontend
│   ├── public
│   │   ├── index.html
│   │   └── manifest.json
│   ├── src
│   │   ├── components
│   │   │   └── ChatbotDocR
│   │   │       ├── images
│   │   │       ├── DRFooter.jsx
│   │   │       ├── DRHeader.jsx
│   │   │       ├── DRTextChat.jsx
│   │   │       └── DocRWidget.jsx
│   │   ├── App.js
│   │   └── index.js
│   ├── package.json
│   └── package-lock.json
├── .gitignore
└── README.md
```

Figure 4: Project directory map

5. DISCUSSION

Several notable points emerge from the development, use, and subsequent evolution of the ESCB. In this section, we discuss the choice of GPT models, the considerations around CLPD text chunk size, and reflect on user community interactions with the ESCB. We also make some observations related to this exploration and consider aspects for the future. Figure 5 identifies how these key elements relate to each action research cycle iteration.

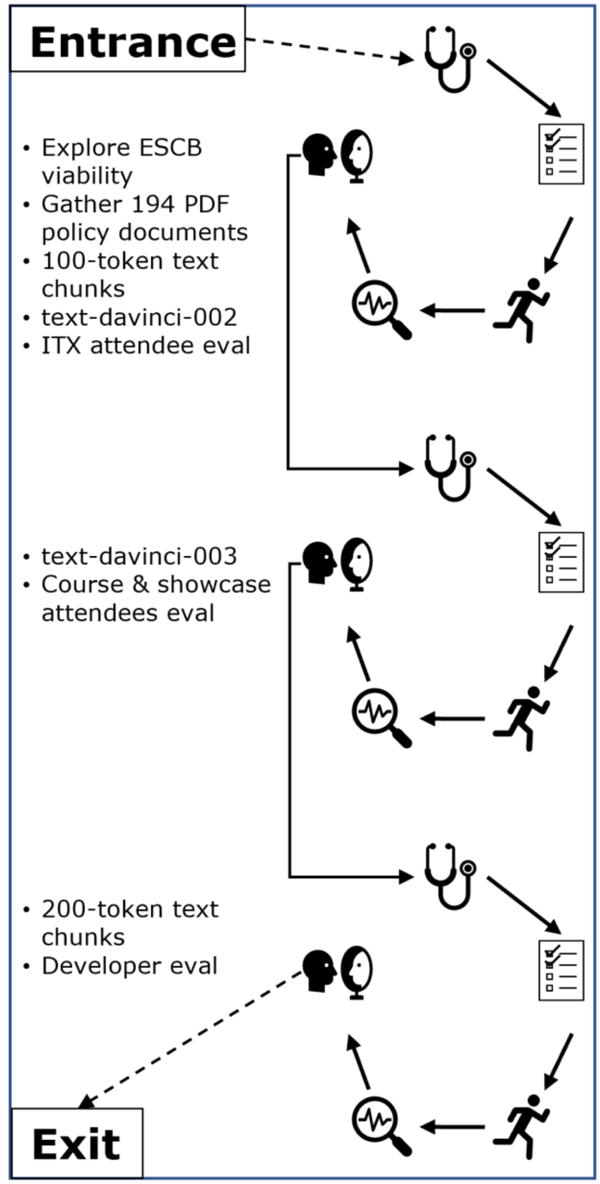

Figure 5: Highlights from the three action research cycle iterations

Language Model

OpenAI consistently evolves its line of large ML foundation models and has thus far produced GPT-1, GPT-2, GPT-3, GPT-3.5, and GPT-4 ("Generative pre-trained transformer," 2023). Models based on GPT-3.5 include *text-davinci-002*, released in March 2022 (Bavarian et al., 2022), and *text-davinci-003*, released in November 2022 (Laike, 2022). The initial ESCB version made use of *text-davinci-002*. This choice was based on the capabilities of the davinci model line compared to the others (ada, curie, babbage), the relative newness of the model, and the high availability of articles discussing it and coding examples demonstrating its use.

During the evaluation phase of the first action research cycle, it became clear that the ITX participants received the best answers when they wrote queries as specifically as possible. Vague language periodically resulted in API responses that did not fully satisfy the query. During reflection, we decided to investigate the next version of the model, *text-davinci-003*. OpenAI indicated that improvements included producing "higher quality writing," handling "more complex instructions," and being "better at longer form content generation" (Raf, n.d.a). As a result of this information and testing results, we changed the API call code to use the new model.

In subsequent user community evaluations at the course project presentations and the semi-annual computing showcase, there were clear improvements in query response quality. With the model transition, the chatbot began providing better answers that addressed user queries more accurately, even when user phrasing was vague. This switch improved the quality of user-chatbot interactions and demonstrated a key benefit of using these AI tools. Since the model is specified with each API call, it was quite easy to edit the call to use the new model. This bodes well for the future maintainability of an ESCB.

Text Chunks

Key ESCB design considerations included the size, in tokens, of the text chunks to create from the CLPD and to semantically compare to user queries, as well as the number of text chunks to send to the GPT API along with the user query. The two factors involved were GPT token size limitations and cost.

Both *text-davinci-002* and *text-davinci-003* have API request limits of 4,097 total tokens for the combined prompt (user query + text chunks sent) and completion (answer received) (Models - OpenAI API, n.d.; Raf, n.d.b). Assuming a user query length of about 20 tokens and adding the 18 tokens required for prompt formatting (i.e., labels for Question, Relevant Data, Chat History, User, and Response), our initial choice to send 10 CLPD text chunks, each 100 tokens in length, would mean that the GPT API call would be 1,038 tokens and the answer could be a maximum size of 3,059 tokens. This would be approximately 2,295 words, likely far more than our ESCB answer should require, so more than long enough. A last point to consider regarding length is that our ESCB design intended to take advantage of the semantic context of a chat session, so any subsequent GPT API exchanges included the cumulative chat history (user query + API response) of every previous exchange.

The cost to interact with either *text-davinci-002* or *text-davinci-003* is the same: $0.02/1,000 tokens (Deprecations - OpenAI API, n.d.). So, assuming the same sending size from the previous paragraph (1,038 tokens) and a GPT response of 95 tokens, the cost for this single, simple notional interaction (1,133 tokens) would be $0.02266 (2.266 cents). A follow-up second query/response would increase in cost to 1248 tokens or 2.496 cents – 20 token second query + 18 token formatting + 1,000 token CLPD text chunks + 115 token chat history + 95 token second response. Note that each interaction incurs a cost based on token count. While the size limit of 4,097 tokens may not be a concern, the cost of a highly active system might begin to burn through a non-trivial amount of money.

After observing the second and third sets of user evaluations of the ESCB and receiving helpful feedback, we reflected that a third iteration of the action research cycle would be worthwhile to explore changing the text chunk size. We had chosen the 100-token size heuristically based on articles read and example code snippets examined. Since there was little concern about exceeding the 4,097 token size limit and our ESCB was only experimentally used as we permitted, thus throttling costs, we decided to double the text chunk size to 200 tokens. This meant that the example first API call would be 2,038 tokens long and permit replies up to 2,059 tokens (~1,545 words), while the cost for the example first call-response would increase to 4.266 cents. Moreover, as with the previous example, subsequent calls in the same session would include the history, so an example second call would cost 4.496 cents. The result, in our estimation, was that the increase in size of the 10 CLPD text chunks sent with each user query did influence the quality of the GPT responses. The difference is difficult to quantify, but in many cases, we subjectively assessed it to be better. Appendix A provides examples using a mix of language models and text chunk sizes, while Figure 6 provides a single example.

User Community ESCB Interaction

One rather conspicuous aspect of user-ESCB interactions emerged quickly and is related to the relationship between the quality of the user query and the accuracy of the ESCB response. Independent of the ESCB technical aspects of language model and text chunk size, the more detailed the user query, the better. After observing just a few user-ESCB interactions, we perceived three general user query quality categories emerging and called them vague, direct, and detailed. For example:

- ***vague***: Can I bring my own device?
- ***direct***: As a professor, can I bring my own device?
- ***detailed***: As a professor, can I bring my own device to teach my class from and use in presentations?

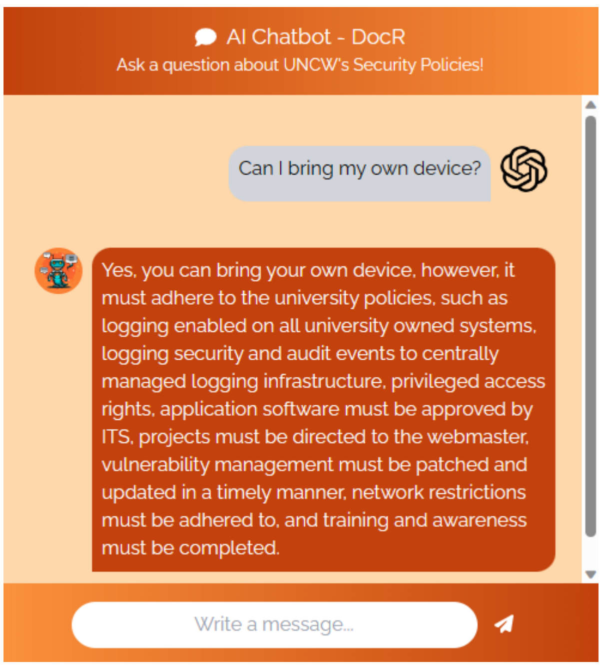

Figure 6: A look and feel of the ESCB; depicted is an answer to the vague phrasing of the example query using text-davinci-003 and 200-token CLPD text chunks.

It was not possible to strictly define the categories and precisely quantify the results, but it appeared that queries we judged as vague returned satisfactory answers regularly (maybe two-thirds of the time), with follow-up queries periodically required. Queries we labeled as direct received satisfactory answers often – about four out of five times on the first try and detailed queries got helpful answers almost every time – better than ~95% of the time.

These observations hold for queries relevant to the CLPD topics and related to our local organization's policies. General information/knowledge queries were excluded from this analysis.

Another aspect of user-ESCB interactions we qualitatively considered relates to the GPT response quality after changing the language model (*text-davinci-002* to *text-davinci-003*) and text chunk size (100 to 200 tokens). Perhaps unsurprisingly, we noted that changing the language model resulted in improved answers from GPT. There was also an improvement, though arguably less noticeable, in the answer quality when CLPD chunks of 200 tokens were provided to GPT with the user query.

To provide one example, here is a vague query and response using text-davinci-002 with text chunk token size 100 compared with a detailed query and the response using text-davinci-003 with text chunk token size 200:

<u>Query</u>: *Can I bring my own device?*
<u>Response</u>: *The university requires that all data storage and processing devices be examined prior to disposal or transfer to surplus. Additionally, privileged access should only be granted to individuals after an evaluation of need and an official approval from the appropriate authority. Furthermore, all university servers must log security and audit events to centrally managed logging infrastructure.*

<u>Query</u>: *As a professor, can I bring my own device to teach my class from and use in presentations?*
<u>Response</u>: *Yes, you can bring you own device to teach your class from and use in presentations as long as it is approved by ITS and all software installed on the device is also approved by ITS. You must also ensure that all data storage and processing devices are examined prior to disposal or transfer to surplus and any institutional or protected data, proprietary software, or software not licensed to be transferred with the computer is removed using ITS Security approved data destruction process or by physically destroying the media. Additionally, all university computers must be kept up-to-date with patches, security, and managed with best practices, and all university data storage or processing devices must be logged and monitored for inconsistencies or unauthorized access.*

Supplemental AI-Impact Observations

In addition to the previous discussion of the technical considerations of language model and text chunk size/number, we note that this applied research illuminated other points meriting brief mention. These include implications on the gap among technology users of varying familiarity with AI, the challenge of making AI more "human," and AI affordability.

The development of the technique described to create a usable ESCB demonstrates that one's understanding of LLM training, NLP, and how semantic search is computationally possible are not prerequisites. This shows that a high level of expertise in AI-related technologies is not required to begin leveraging them to meet an organization's operational needs. The gap between what highly technical AI practitioners could do a few years ago compared to semi/quasi-technical practitioners has closed remarkably. This has important implications for the democratization of data access, data use, and the broader adoption of AI technologies across diverse sectors and contexts.

One of the challenges of implementing AI in customer service or other interactive scenarios where a human is being augmented or replaced is the need to make the interaction feel "human." While this is a complex and multifaceted challenge, our chatbot's performance suggests that a key factor in achieving "human" interaction is the ability to understand and respond accurately to diverse and sometimes vaguely phrased user queries. By evolving the model and improving its interpretative capabilities, we were able to make our chatbot interaction more "human." Given the parameterized manner in which we accomplished this, we observed that our chatbot's responses became more articulate and grammatically sound. Although this does not necessarily translate to a more human-like interaction, it signifies a substantial improvement in ESCB capacity to understand and respond to diverse user queries coherently and comprehensively.

Our ESCB development technique also demonstrates how utilizing local data with established chatbot models can enable organizations to leverage LLMs and AI technologies without spending substantial amounts of money on proprietary data training. This opens the possibility for even small businesses and organizations with limited resources to leverage the power of AI in their operations. The per-use cost of OpenAI's GPT API has already fallen significantly since the conclusion of our applied research project with *gpt-3.5-turbo-instruct*, the currently recommended replacement for *text-davinci-003* (Deprecations - OpenAI API, n.d.), advertised to cost ~90% less (Pricing – OpenAI, n.d.) at $0.0015/1000 tokens input and $0.002/1000 tokens output. The total expenditure for API calls during this research totaled about $10.50. We estimate that similar work today would cost less than $2.00 for API calls.

Potential Improvements

There are several areas for improvement to our technique for ESCB development and the resultant chatbot. First, to enhance user interaction, we suggest the introduction of a loading animation during the process of an API request. This addition provides explicit feedback to the user, informing them that a prompt has been sent and a response is expected. Second, we suggest the implementation of a filter that verifies whether the user's query is within the scope of the database. The current chatbot design sends accompanying information from the CLPD, semantically like the user's question, with every request, regardless of its relevance.

The goal of an ESCB is to answer a narrow range of questions – in our case, those related to our local organization's written operational policies. It is not expected to do well with general knowledge queries. Lastly, we propose introducing a feature that enables toggling between a database call and a query about the retrieved information. This feature would enrich user-data interaction and limit queries sent to the API to only contain the necessary information, resulting in cost reduction.

6. CONCLUSION

This paper presented applied research into the development of a technique for building an enterprise-specific chatbot (ESCB) using readily available AI tools, basic programming competence, and user community feedback. We recognize it may be difficult to generalize this action research result because some unnoticed qualities of our environment led to success. However, we believe this technique is sufficiently straightforward as presented that members of other interested organizations can replicate the process and successfully develop their own ESCB for local use.

This paper underscores the significant potential of AI-powered chatbots for data interaction of organizations of all sizes. By presenting an affordable and versatile model, we hope to show how organizations with limited resources can deploy AI in their local operations. Additionally, this work exhibits the remarkable potential of modern LLMs, making complex subjects accessible to a broader range of individuals and fostering a more inclusive overall information environment.

Due to the substantial cost of training LLMs, not all companies can afford to train a model with their proprietary data. Consequently, organizations wishing to utilize LLMs in their operations require alternative solutions. Our ESCB illustrates a straightforward approach to this problem. By using word embeddings and semantic searches, users can send only the relevant information needed to answer their queries within an API request.

Integrating OpenAI's GPT LLM into our chatbot allows for understanding vague or abstract language phrasing. As a result, users from many backgrounds, not merely those with technical expertise, can interact more comfortably and derive insights from complex topics. This development enhances the democratization of information and underscores the role AI can play in bridging the gap between complex data and its wider audience.

7. REFERENCES

Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020, March). A smart chatbot architecture based NLP and machine learning for health care assistance. In Proceedings of the 3rd international conference on networking, information systems & security (pp. 1-6). https://dl.acm.org/doi/10.1145/3386723.3387897

Bartleby. (2023). Familiar quotations. Compiled by John Bartlett. https://www.bartleby.com/lit-hub/familiar-quotations/1970-franis-baon-1561-1626-john-bartlett/

Bavarian, M., Jiang, A., Jun, H. & Pondé, H. (2022, March 15). New GPT-3 capabilities: Edit & insert. OpenAI. https://openai.com/blog/gpt-3-edit-insert

Cleverbot. (2023). https://www.cleverbot.com/

Davison, R., Martinsons, M. G., & Kock, N. (2004). Principles of canonical action research. Information systems journal, 14(1), 65-86. https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2575.2004.00162.x

Deprecations - OpenAI API (n.d.). OpenAI Documentation. https://platform.openai.com/docs/deprecations/

Generative pre-trained transformer. (2023, July 8). In Wikipedia. https://en.wikipedia.org/wiki/Generative_pre-trained_transformer#Foundational_models

GPT-3. (2023). Models. https://platform.openai.com/docs/models/gpt-3

Hi, I'm Kuki. (2023). @kuki_ai. https://www.kuki.ai/

IBM Watson. (2023). https://www.ibm.com/watson

Kalla, D., & Smith, N. (2023). Study and Analysis of Chat GPT and its Impact on Different Fields of Study. International Journal of Innovative Science and Research Technology, 8(3). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4402499

Laike, J. [@janLeike]. (2022, November 28). Check out OpenAI's new text-davinci-003! [Tweet]. Twitter. https://twitter.com/janleike/status/1597355354433916928

Landsteiner, N. (2005). Eliza. https://www.masswerk.at/elizabot/

Leedy, P. D., & Ormrod, J. E. (2010). Practical research, planning and design, 9th edn, New Jersey: Pearson. https://josemartimast.net/wp-content/uploads/2021/07/AP-Capstone-Research-Planning-and-Designing-E-Book.pdf

Mauldin, M.L. (1994, August). Chatterbots, tinymuds, and the turing test: entering the Loebner prize competition. *AAAI* (Vol. 94, pp 16-21). https://cdn.aaai.org/AAAI/1994/AAAI94-003.pdf

Mikolov, T., Yih, W. T., & Zweig, G. (2013, June). Linguistic regularities in continuous space word representations. In Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies (pp. 746-751). https://aclanthology.org/N13-1090.pdf

Models - OpenAI API (n.d.). OpenAI Documentation. https://platform.openai.com/docs/models/gpt-3-5

Nath, S., Marie, A., Ellershaw, S., Korot, E., & Keane, P. A. (2022). New meaning for NLP: the trials and tribulations of natural language processing with GPT-3 in ophthalmology. British Journal of Ophthalmology, 106(7), 889-892. https://doi.org/10.1136/bjophthal mol-2022-321141

Nuruzzaman, M., & Hussain, O. K. (2018, October). A survey on chatbot implementation in customer service industry through deep neural networks. In 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE) (pp. 54-61). IEEE. https://doi.org/10.1109/ICEBE.2018.00019

Partalidou, E., Spyromitros-Xioufis, E., Doropoulos, S., Vologiannidis, S., & Diamantaras, K. (2019, October). Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy. In IEEE/WIC/ACM International Conference on Web Intelligence (pp. 337-341). https://doi.org/10.48550/arXiv.1912.10162

Patel, N., & Trivedi, S. (2020). Leveraging Predictive Modeling, Machine Learning Personalization, NLP Customer Support, and AI Chatbots to Increase Customer Loyalty. Empirical Quests for Management Essences, 3(3), 1-24. https://researchberg.com/index.php/eqme/article/view/46

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830. https://doi.org/10.48550/arXiv.1201.0490

Pricing - OpenAI. (n.d.). OpenAI. https://openai.com/pricing

Raf. (n.d.a). How do text-davinci-002 and text-davinci-003 differ? Openai help center. https://help.openai.com/en/articles/6779149-how-do-text-davinci-002-and-text-davinci-003-differ

Raf. (n.d.b). What are tokens and how to count them? Openai help center. https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them

Susman, G. & Evered, R. (1978). An Assessment of The Scientific Merits of Action Research. Administrative Science Quarterly, (23) 4, 582-603. https://doi.org/10.2307/2392581

Wei, W., Barnaghi, P. M., & Bargiela, A. (2008). Search with meanings: an overview of semantic search systems. *International journal of Communications of SIWN*, *3*, 76-82. http://www.bargiela.com/papers/a35.pdf

Weizenbaum, J. (1966). ELIZA – a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36-45. https://dl.acm.org/doi/pdf/10.1145/365153.365168

APPENDIX A

Examples of ESCB Query-Response

Below are query-response examples for a single topic user question using three different question framings, two different language models, and two different text chunk sizes.

The table below refers to the following user query framing examples:

Vague (V): *Can I bring my own device?*
Direct (Di): *As a professor, can I bring my own device?*
Detailed (De): *As a professor, can I bring my own device to teach my class from and use in presentations?*

The Tokens/Cost column reflects the number of tokens in the user query (7, 11, or 20) + the 18 formatting tokens + the number of tokens in the 10 CLPD text chunks (1000 or 2000) + the number of tokens in the GPT response (varies from 52 – 154).

Query	Response	Tokens/ Cost
text-davinci-002 with 100-token CLPD text chunks		
V	The university requires that all data storage and processing devices be examined prior to disposal or transfer to surplus. Additionally, privileged access should only be granted to individuals after an evaluation of need and an official approval from the appropriate authority. Furthermore, all university servers must log security and audit events to centrally managed logging infrastructure.	1086 $0.02172
Di	The university requires that all individuals who have access to electronic information stored on its information technology resources complete information security training as deemed appropriate by their supervisor, department, or other appropriate authority. In addition, security training may be required based on the sensitivity of the role or compliance requirement.	1084 $0.02168
De	The university's information technology resources are powerful tools that are shared among all members of the campus community. These resources are intended to be used in useful and productive ways. However, the university cannot protect individuals against the existence or receipt of materials that may be offensive or annoying to them. Additionally, the university requires authorized employees who have access to university information technology resources to take appropriate action when necessary to ensure the orderly administration and function of these resources.	1126 $0.02252
text-davinci-003 with 100-token CLPD text chunks		
V	Yes, you can bring your own device, provided that you adhere to the relevant data and policies outlined in the conversation history. This includes not posting or disseminating personal or sensitive information, controlling entry points, providing identification when transferring assets off-site, granting privileged access rights only after an evaluation of need and approval, keeping systems current with patches and security, consulting the webmaster before committing to new web-based technology, enabling logging on all university-owned systems, logging security and audit events to a	1125 $0.02250
Di	Yes, you can bring your own device as a professor, however you must adhere to the policies outlines in the university's Responsible Use, Server Management and Standards, Privacy and Confidentiality, Information Security, and Campus Web Resources policies. This includes not posting or disseminating personal or sensitive information, notifying your department chair, dean, director, or administrator if you have concerns about the conduct of a member of the university community, keeping systems current with patches, security, and managed with best practices	1129 $0.02258
De	Yes, you can bring your own device to teach your class from and use in presentations as long as you comply with the university's policies and	1136

	procedures regarding the use of electronic information and data processing devices. You should also be aware of the university's policy on server management and standards, responsible use, campus web resources, and information security. Additionally, you should ensure that all systems are kept current with patches, security, and managed with best practices, and that you have completed the necessary information security	$0.02272
text-davinci-002 with 200-token CLPD text chunks		
V	No, users may not post or disseminate personal or sensitive information about an individual or individuals. Such information includes, but is not limited to, academic records, medical information, social security numbers, or similar information of a personal and confidential nature that, if disseminated, could have legal or otherwise damaging implications either for the targeted person or the institution.	2095 $0.04190
Di	The university policies regarding user access and registration state that users are responsible for all activity performed with their personal IDs. When a user's relationship with the university is terminated, their access to university computing resources will be denied. IDs cannot be reassigned after being issued.	2081 $0.04162
De	The university's policy on electronic information states that authorized employees may have access to electronic information stored on university information technology resources in order to ensure the orderly administration and function of these resources. However, the university cannot protect individuals against the existence or receipt of materials that may be offensive or annoying to them. Additionally, the university's policy on server management and standards requires that all software to be installed on university enterprise systems be approved by ITS, and that application owners consult the Webmaster before committing to new web-based technology.	2141 $0.04282
text-davinci-003 with 200-token CLPD text chunks		
V	Yes, you can bring your own device, however, it must adhere to the university policies, such as logging enabled on all university owned systems, logging security and audit events to centrally managed logging infrastructure, privileged access rights, application software must be approved by ITS, projects must be directed to the webmaster, vulnerability management must be patched and updated in a timely manner, network restrictions must be adhered to, and training and awareness must be completed.	2114 $0.04228
Di	Yes, as a professor at UNCW, you can bring your own device, provided that you adhere to the university's policies and procedures for responsible use. This includes not posting or disseminating personal or sensitive information about individuals, reporting any concerns you have about the conduct of a member of the university community, or the propriety of a given situation or activity, and keeping your device up-to-date with patches, security, and managed with best practices. Additionally, you must ensure that any software you install on the university's enterprise systems is approved by ITS, and that any privileged access rights you are granted are approved by the appropriate authority and documented. Finally, you must complete any information security training deemed appropriate by your supervisor, department, or other appropriate authority.	2182 $0.04364
De	Yes, you can bring you own device to teach your class from and use in presentations as long as it is approved by ITS and all software installed on the device is also approved by ITS. You must also ensure that all data storage and processing devices are examined prior to disposal or transfer to surplus and any institutional or protected data, proprietary software, or software not licensed to be transferred with the computer is removed using ITS Security approved data destruction process or by physically destroying the media. Additionally, all university computers must be kept up-to-date with patches, security, and managed with best practices, and all university data storage or processing devices must be logged and monitored for inconsistencies or unauthorized access.	2175 $0.04350