Bringing the State-of-the-Art to Customers: A Neural Agent Assistant Framework for Customer Service Support

Stephen Obadinma[b] , **Faiza Khan Khattak**[a], **Shirley Wang**[a,c], **Tania Sidhom**[a,b], **Elaine Lau**[d,g], **Sean Robertson**[a,c], **Jingcheng Niu**[a,c], **Winnie Au**[a], **Alif Munim**[a], **Karthik Raja K. Bhaskar**[e], **Bencheng Wei**[e], **Iris Ren**[e], **Waqar Muhammad**[e], **Erin Li**[e], **Bukola Ishola**[e], **Michael Wang**[f], **Griffin Tanner**[f], **Yu-Jia Shiah**[g], **Sean X. Zhang**[g], **Kwesi P. Apponsah**[g], **Kanishk Patel**[h], **Jaswinder Narain**[g], **Deval Pandya**[a], **Xiaodan Zhu**[a,b], **Frank Rudzicz**[a,c], **Elham Dolatabadi**[a,c]

[a]Vector Institute for Artificial Intelligence, [b]Queen's University , [c]University of Toronto,
[d]McGill University, [e]CIBC, [f]KPMG, [g]PwC Canada, [h]University of Alberta, [g]Mila

Abstract

Building Agent Assistants that can help improve customer service support requires inputs from industry users and their customers, as well as knowledge about state-of-the-art Natural Language Processing (NLP) technology. We combine expertise from academia and industry to bridge the gap and build task/domain-specific Neural Agent Assistants (NAA) with three high-level components for: (1) Intent Identification, (2) Context Retrieval, and (3) Response Generation. In this paper, we outline the pipeline of the NAA's core system and also present three case studies in which three industry partners successfully adapt the framework to find solutions to their unique challenges. Our findings suggest that a collaborative process is instrumental in spurring the development of emerging NLP models for Conversational AI tasks in industry. The full reference implementation code and results are available at `https://github.com/VectorInstitute/NAA`

1 Introduction

Rising demand for AI-powered conversational agents (Fu et al., 2022; Sundar and Heck, 2022), especially for customer support service, is estimated to grow at a compound annual growth rate (CAGR) of 23.4%, earning a predicted revenue of around $29.9B USD by 2028 (Market, 2018). As such, conversational AI research has increased substantially, especially to enhance customer service support (Nicolescu and Tudorache, 2022). Despite a proliferation of agent assistants from Microsoft, IBM, Oracle, and Google, there remain many unanswered questions and challenges (Fu et al., 2022) that need to be addressed before the widespread proliferation of AI in practice. We argue that bridging the gap between natural language processing (NLP) research in academia and industry is an overlooked issue in conversational AI.

Apart from the handful of aforementioned conglomerates, it is difficult for most other companies, which are not in the process of conducting cutting-edge NLP research, to benefit from the recent progress of NLP. Conventional conversational AI architectures are delicate and complex, and require a large degree of specialised knowledge to bring a full system into fruition. However, with the introduction of the large-scale pretrained transformer-based language models such as BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019), it is possible for smaller teams to take advantage of this "monopoly" by being able to fine-tune these powerful models on their comparatively small amount of data and achieve high performance, therefore harnessing the architectural achievements of those powerful language models to devise their own conversational AI systems that exploit their more fine-grained expertise and knowledge of their customers' needs.

The interdisciplinary nature of AI-enabled customer service support makes for an inherently difficulty task (Nicolescu and Tudorache, 2022). To arrive at an answer to a user query, a system must bore through several layers of complexity: first, the intent behind the question must be categorized and quantized into a form which can be manipulated by the system; second, determining from thousands or millions of stored records the information relevant to that intent; and finally to manipulate said information into a form which the user may understand.

To handle such complexity, techniques from across NLP must be employed, raising the barrier-of-entry for companies with significant customer support needs. Larger, well-established companies can offer solutions as a service, but said solutions are often closed-source, trained on generic data with few in-domain terms, and may not be easily integrated into a company's workflow.

This paper represents the cumulative efforts of

Proceedings of EMNLP 2022 Industry Track, pages 450–460
December 9–11, 2020. ©2022 Association for Computational Linguistics

both researchers and industry practitioners over a yearlong project (whose details may be found in Appendix A) to develop state-of-the-art customer service support systems. In its publication, we hope to help lower the barrier-of-entry for future industry practitioners by inspiring similar collaborations. Our contributions are twofold and in line with the strengths of the contributors:

1. We release an open-source Neural Agent Assistant (NAA) based on state-of-the-art neural architectures (Fu et al., 2022). The implementations are well-documented, illustrating how the systems can be extended for specific use cases.
2. To wit, we explore three case studies in which three industry partners successfully adapt the agent to find solutions to their unique challenges.

In their presentation, we stress that a perfect, universal customer service system is unattainable as their real-world applications are not identical. As such, any dialogue about real-world NAA must feature a discussion of both the technologies shared with, and the differences between tasks.

2 Neural Agent Assistant framework for Customer Service Support

As the basis of our collaborative project, we established a common Neural Agent Assistant (NAA) framework to serve as a guide for industry partners to adapt to their companies' workflows. Those technologies were made available in a clear, easy-to-use engineering pipeline, alleviating one of the greater challenges for industry in obtaining a foothold in building competent neural agents.

The framework features three high-level components shown in Figure 1: (1) Intent Identification, (2) Context Retrieval, and (3) Response Generation. Though designed as parts in a pipeline in generating a natural language response to customer questions (Figure 1 (b)), the output from any given component can nonetheless be transferred back to the human agent when needed (Figure 1 (a)).

2.1 Intent Identification

Intent identification is a critical component of NAA as it is often the first step in a customer service pipeline upon which all subsequent components depend on. Its goal is to classify the intent of a given query. This may be binary, determining whether the query is relevant or irrelevant to the knowledge base (KB) (see Section 3). More often, the task involves classifying the query into a fine-grained, domain-specific intent class. For this purpose, we designed a query encoder to detect and understand customers' input and classify it into one of N classes. The encoder consists of a pre-trained, Transformer-based language model (Vaswani et al., 2017), followed by a pooling layer, then a final linear layer. A query is fed as an input to the encoder, with the latter outputting a categorical probability distribution over the intents.

We chose BERT (Devlin et al., 2018) as the encoder, which we fine-tune and evaluate on two task-specific corpora: Banking77 (Casanueva et al., 2020) and CLINC150 (Larson et al., 2019). Following the methodology proposed by (Zhang et al., 2021), we first fine-tune the BERT model for the intent classification task using the Banking77 dataset, which is composed of 13k queries labelled into 77 intents. We then evaluate it on the CLINC150 dataset (banking domain) in the few-shot setting using all 15 classes. We achieved 92% F1-score for fine-tuning and 85% and 90% F1-score for one-shot and five-shot learning, respectively.

For NAA's core system demonstration, the intent label is not used to guide the context retrieval and response generation, but it can be easily leveraged for knowledge extraction and abstractive summarization as shown in Section 3.

It is important to note that intent identification is only one of the main natural language understanding (NLU) components that a complete customer support agent would possess. Named-entity recognition (NER) is often an equally important task, where the goal is to identify any named entities, including relevant people, organizations, or products the user is inquiring about, as well as details about time and location, like for scheduling a restaurant reservation. There can be a great need for an agent to identify domain-specific entities in user dialog for the purposes of slot filling, hence there is a similar need for NER frameworks for easily adapting to domain-specific industry needs. For the sake of simplicity, and because an NER component was not requested from our industry partners, it is not included within the scope of the project. We leave developing this NER framework to future work.

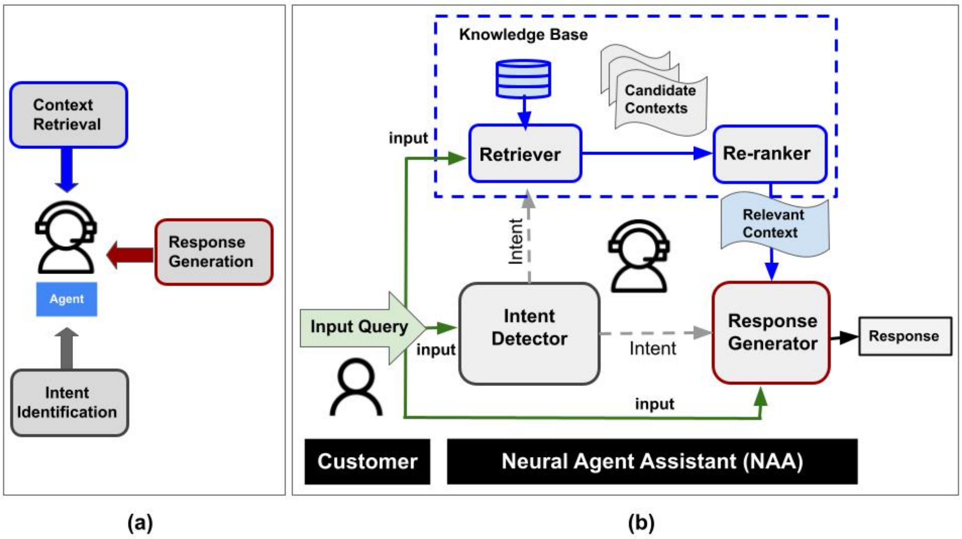

Figure 1: Neural Agent Assistant framework. **(a)** shows how a human agent is able to leverage the three NAA components for support. **(b)** shows the complete NAA pipeline, starting with an input query from a customer which is first fed to the NAA's intent identification module. The intent label is generated and can be used to guide the context retrieval and response generator. The retriever and re-ranker work together to find the most relevant context candidate passages from the KB, which the response generator employs to craft a response to the user query.

Table 1: Primary results for Passage Retriever (PR) and the Re-Ranker (RR), comparing the performance of pretrained and fine-tuned models with a variety of standard passage retrieval evaluation metrics (note higher results are better).

		MRR	MAP
PR	**pretrained**		
	msmarco-distilbert-base-tas-b	0.835	0.801
	fine-tuned		
	multi-qa-mpnet-base-dot-v1	**0.886**	**0.860**
	msmarco-distilbert-base-tas-b	0.864	0.834
RR	**pretrained**		
	msmarco-MiniLM-L-6-v2	0.998	0.997
	fine-tuned		
	msmarco-MiniLM-L-6-v2	**1.000**	**1.000**

2.2 Context Retrieval

This component is designed to retrieve the most semantically similar documents from a predefined knowledge base (corpus of documents) given a user query. Documents or passages extracted from the KB are presumed to contain an appropriate answer to the query, though not in a concise form. Context retrieval is a two-step process starting with passage retrieval and followed by re-ranking. The retriever uses a bi-encoder design (Karpukhin et al., 2020), extracting supporting context from a knowledge-base at the level of paragraphs (hence asymmetric semantic search) based on SentenceBERT (Reimers and Gurevych, 2019). The re-ranker is a BERT-based encoder designed for sequence-pair classification that scores the relevancy of all top-k retrieved candidate contexts for a given input query (Nogueira and Cho, 2019).

Our implementation includes pipelines to run the retriever and re-ranker pre-trained on MSMARCO (Bajaj et al., 2016) as-is, or to fine-tune both models using Multiple Negatives Ranking Loss on a knowledge-grounded, question-answering corpus such as ELI5 (Fan et al., 2019). The MSMARCO dataset was generated by sampling and anonymizing Bing usage logs. The dataset includes over 1 million queries with at least one human-generated answer per query, as well as relevant Wikipedia passages retrieved by Bing for each query. The ELI5 dataset is made up of complex questions and long and explanatory answers from Reddit users about random topics. Each question and answer pair is grounded in relevant Wikipedia passages for supporting information. Table 1 compares and reports the performance of the pretrained and fine-tuned models on the ELI5 dataset, illustrating the benefits of fine-tuning across systems and conditions.

Different alternative approaches have been taken in previous works for context retrieval, including combining dense passage retrieval with lexical rule-based retrieval such as BM25 to improve results and computational efficiency (Gao et al., 2021; Khazaeli et al., 2021), and using extensive pretrain-

Table 2: Primary results for the Response Generation component of NAA. Models were fine-tuned on MSMARCO.

Model	Test-set	F1-score	BLEU-1	Rouge-L
GPT2-meduim	MSMARCO	40.2%	32.0%	36.0%
GPT2-large	MSMARCO	32.0%	20.4%	28.0%
DialoGPT-medium	MSMARCO	28.2%	15.3%	24.6%
GPT2-meduim	ELI5	6.98%	0.1%	5.22%

ing on the encoders (Khattab and Zaharia, 2020). Although these methods have been shown to be effective in some cases, it adds further complexity and makes for a less generalizable approach compared to the BERT-based approach we take.

2.3 Response Generation

The ultimate goal of our NAA is to be able to automatically generate a human-like answer to a customer's query. Therefore, the final component is dedicated to producing in-context natural-language responses to customers' queries via a generative transformer model. The model is fed the ranked passages from the retrieval component and outputs the natural-language response. Though the input is extractive, the output is abstractive.

Our reference implementation provides pipelines for training GPT-2 models of any size (Radford et al., 2019) for the question answering (QA) task on both of the above-mentioned knowledge-grounded datasets: MSMARCO and ELI5. We used a multi-task loss combining language modeling with a next-sentence prediction objective. Following training, nucleus top-p sampling was used for decoding and text generation. The results for GPT-2 medium are presented in Table 2.

The three components of our pipeline - intent classification, context retrieval, and response generation - perform well on the domains and tasks they were trained on. However, as discussed in Section 1, a general system will fail to account for the specific challenges companies face related to the content and quality of NAA. As was illustrated, the pipeline above provides numerous opportunities for tailoring each function, opportunities leveraged by our industry partners in the next section.

3 Industry-specific Proof Of Concept Implementations

In this section, we demonstrate how the NAA framework outlined in Section 2 was adapted to our industry partners' use-cases and settings, which is continuing to be refined for deployment in real-world scenarios.

Canadian Imperial Bank of Commerce (CIBC) is a financial institution with an Advanced Analytics team focusing on finding the correct curated response to banking queries. KPMG is a professional services firm which is looking to NAA to service internal queries over large bodies of legal and regulatory documents. PricewaterhouseCoopers (PwC) is also a professional services firm looking to better advise its banking clients on building NAA technologies. In the following subsections we clarify the companies' motivations in participating in the project, how they adapted our NAA framework, and their plans for deployment.

3.1 Company I: CIBC - NAA Tools for Banking Customer Service Support

The primary motivation for implementing and deploying NAA by CIBC is to support the bank's continued focus on leveraging digital technologies to make clients' banking experiences even better.

The pipeline modified by CIBC (see Figure 2 in Appendix A) consisted of four components. The first component was a binary classifier (i.e., a BERT encoder) which was trained to classify whether an input query has a banking intent. The inputs identified as being banking related were further processed by the assistant. Module 2 was the same as NAA's intent identification component, though it was improved through data augmentation: the component was fine-tuned on back translation (Sennrich et al., 2016) (English → German → English) and insertion data. With data augmentation, F1-scores improved from 91.8% to 92.7%. Module 3 was a knowledge-driven QA system including both the Context Retrieval and Response Generation components of the NAA. If the banking intent from the previous layer had a confidence score greater than parity, then the top 5 most relevant financial contexts from a curated financial KB were retrieved and ranked. The KB included long-format questions and answers, as well as FAQs downloaded from the company's publicly available website. The context, therefore, was question/answer pairs that, after being retrieved, provided a similar question along with an answer to the customer query. In addition, the GPT-2 Medium model pretrained on MSMARCO was also used to gener-

Table 3: KPMG: Legal data NAA pipeline results. Passage size (PS) refers to number of sentences used for the passages (2, 3, and 4 for *short*, *medium*, and *long* respectively). Corpus refers to whether the dataset was preprocessed. Retriever refers to corpus embedding bi-encoder (all retrieval components are pretrained on MSMARCO). Response generator (RG) shows the pretraining dataset used for GPT-2-medium.

PS	Corpus	Retreiver	RG	Manual Score	BLEU-1	F1-score
Med.	Raw	DISTILBERT	ELI5	44	0.193	0.218
Med.	Clean	DISTILBERT	ELI5	40	0.176	0.198
Short	Clean	DISTILBERT	ELI5	30	0.181	0.206
Long	Clean	DISTILBERT	ELI5	43	0.212	0.232
Med.	Clean	BERT	ELI5	49	0.303	0.336
Med.	Raw	BERT	MSMARCO	**74**	0.450	**0.506**
Med.	Clean	BERT	MSMARCO	N/A	**0.459**	0.443
Short	Clean	BERT	MSMARCO	N/A	0.422	0.426

ate human-like responses for the questions based on the retrieved long answers. Low-confidence queries were passed into the final component of the pipeline, which is based on KeyBERT, an architecture utilizing BERT, and deals with out-of-domain intent identification. Using the KeyBERT, keywords from the query were extracted and kept as new out-of-domain intents. Moreover, these new out-of-domain intents were recorded and tracked to enhance banking intent classification in the future by creating more relevant intent categories.

Following a successful implementation of the pipeline, a web application was developed to be used directly by clients. React and REST API service were used as the basis for the web application, which were integrated with the Amazon Web Services (AWS) cloud platform. Extensive developer testing has been conducted, and additional deployment include carrying out agent volunteer testing, which involves using several agents at the bank testing the application for optimal functionality and usability. Lastly, in order to make the system able to dynamically improve in response to customer feedback, the developers plan to implement a feedback loop using customer data. The feedback loop is a mechanism that helps determine how well the model works in production, and provides the necessary feedback to determine whether any changes are needed as a response to customers' user experience with the application and any desired improvements (see Figure 3 in the Appendix).

3.2 Company II: KPMG - A Q&A Tool for Legal Documents Analysis

The motivation for adapting NAA and its implementation by KPMG was reduce the inefficiencies in providing accurate information to user queries in the legal domain drawn from agreements and reports. The company aims to improve their existing text search parsing relying on manual/keyword search using a more robust and easy-to-use novel knowledge-based document query tool based on a NAA.

The pipeline for this implementation included using and externally evaluating the context retrieval and response generation components of NAA framework on a curated legal dataset, CUAD v1 (Hendrycks et al., 2021), which contains over 13*k* sentences based on commercial legal contracts labelled into 41 types of legal clauses. A group of 5 experts including 3 data scientists and 2 legal domain data engineers helped create the legal domain knowledge grounded QA dataset and provided manual evaluation of the performance of the model on the response generation task. The legal QA dataset includes in-house human generated question and answer pairs (n=47) drawn from CUAD v1. In particular, the dataset is a collection of question-answer-evidence triplets, and a KB where the questions and answers were generated by the experts and were accompanied by a supporting context from the KB. The quality and accuracy of the generated responses by the pipeline was evaluated manually by one of the data engineers with legal domain expertise, as well as automatically by using BLEU and F1-scores as can be seen in Table 3. Through manual evaluation, each generated response was scored from 0 to 100 based on their quality and relevance, and the average score of all of them was calculated. The retriever and GPT-2-medium fine-tuned on MSMARCO achieved the highest score of 74 through manual evaluation, as well as a BLEU-1 and F1-score of 0.45 and 0.5, respectively.

A two-phase production deployment plan is implemented for the developed prototype. In phase I, the prototype is planned to be made available to internal employees only as a document processing tool. The service will be based on a cloud-hosted managed service environment, and would allow users to upload a set of documents and enter queries. Based on the queries, related context will be extracted from the documents and abstractive responses, which can be used in the formation of summary reports. Built on phase I, in phase II the tool will be made available to external clients and integrated with their systems through an API for the their managed services platform.

3.3 Company III: PwC Canada - Transferring Emerging Technologies to Financial Customer Queries

The main motivation for adapting and applying NAA by PwC was to gain experience in creating an end-to-end NAA pipeline to process customer queries regarding bank account opening and mortgage applications. The aim was to later use the acquired knowledge to support small to mid-sized financial clients in building NAA chatbots that can reduce inefficiencies when dealing with large volumes of requests (particularly during high-volume seasons like home-buying seasons).

The first component of the prototype included a binary classifier to classify the text into either a "general" or "agent-related" topic to mitigate the answer generation model from confusing the user by responding to irrelevant queries. The binary classifier was trained on a labeled dataset including a total of 15,732 queries (10,331 "agent-related" and 5,401 "general"), and achieves an F1-score of greater than 0.95. The "general" queries were extracted from English-Second-Language practice conversations and common greetings, while "agent-related" queries were pulled from the Banking77 dataset (Casanueva et al., 2020). In case of a general query, a pretrained dialogue generation model, DialoGPT (Zhang et al., 2020), was triggered to make a basic response to the user.

The answer generation component included both the retrieval and response generation tasks. For the retrieval task, both the retriever and re-ranker provided by NAA (pretrained on MSMARCO) were used to extract relevant question-answer pairs from the KB. The KB included 250 long format questions and their respective answers downloaded from FAQs owned by top banks in Canada. Following the retrieval task, the top 3 contexts (including question-answer pairs similar to section 3.1) along with the user's query were provided as inputs to the answer generation model which uses a GPT-2 model further fine-tuned on the 250 question answer sets. The generated answers were compared and evaluated on a set of new human-generated question answer pairs (n=30), achieving a Rouge1 F1-score of 0.088 and a RougeL of 0.084.

The deployment of this solution at the production level is to combine NAA components with Automated Speech Recognition (ASR) tools using a backend service that determines whether a client request is audio or text, after which the request would be processed by the appropriate pipeline. An asynchronous web framework such as NodeJS' Express or Python's FastAPI is planned to be used to create a system scalable to multiple concurrent users, and using an Apache MXNet framework in combination with distributing the workload across multiple GPUs will support inference on the ASR and the response generation models.

4 Summary and Discussion

We summarize our experiences in this section.

Domain/task-specific system works better: As stated earlier, the task/domain agnostic system is not well-suited for different use-cases as our industry partners found, and as confirmed by our experiences. Real-world conversational datasets are noisy and contain domain-specific concepts, and as a result, domain agnostic systems that are trained on clean open-source datasets are likely to have poor performance when evaluated on realistic inputs. The best practice would be to have a system designed/adapted based on domain and task.

Finding a good performance evaluation method: Due to the inherent difficulty in evaluating the quality of retrieved contexts and generated answers, finding a suitable model assessment strategy is imperative (Khapra and Sai, 2021), which includes the processes of model selection, bench-marking, and deciding between manual and automated assessment. Automatic evaluation metrics in particular, although flawed, allow for the ability to benchmark many different model variations, leading to a strong model being picked if there is a general agreement between different metrics. The companies found that using a combination of automatic evaluation metrics like BLEU score, F1-score, Rouge-1, along with expert manual evaluation, helped better inform model selection.

Efficiency and optimization: The efficiency of the algorithm/model being used is important, as is weighing trade-offs and discovering optimizations. For example it was noted that BERT-base models tend to perform better than DistilBERT-base for the bi-encoder, but at the cost of longer inference time. Hence, it is important to consider multiple different approaches when implementing a component and deciding what is the most important aspect of performance. Another challenge that was identified by one company was designing an efficient algorithm for parsing customer inputs into meaningful entities, which was solved by using a custom built parser tailored specifically for in-domain data.

Effectively Combining Academia and Industry Expertise: This collaborative project combined the expertise from academia and industry, which resulted in successful implementation of systems to be deployed at the industry participants' companies. We tried to keep the whole process smooth by having regular meetings and adding documentation. This helped us to collaborate in an efficient and effective manner while keeping engagement throughout the whole process.

5 Conclusion

In this paper, we present a collaborative project for building three customized NAA systems for different tasks and domains in industry. Through our framework and collaborative process, we have tried to bridge the gap between industry and cutting edge NAA technologies by providing much needed open-source tools and expertise in foundational language models and conversational AI. We have observed this framework being successfully adapted into real-world use-cases to enhance customer service support, and it has helped spur NAA development in industry. We hope that our current effort will serve as a valuable use-case to the community, and it is our plan to continue similar collaborative efforts in future in other areas of AI.

6 Ethical Considerations

A limitation of the natural language response generation module of our framework is that there are few guarantees that the model does not produce erroneous or harmful responses in response to user queries. As the base of this module is a language model such as GPT-2 that is pretrained on a massive corpus of data that has not been carefully ensured to be debiased and free from harmful or prejudiced content, it can be susceptible to adversarial inputs and thus has the potential to produce bigoted responses even to benign user queries (Kirk et al., 2021). This presents ethical concerns when having the NAA deal directly with customers, and therefore, to avoid such concerns, we recommend that the NAA in its current form only be used in a human-in-the-loop process, where a human agent works closely with the NAA for assistance. The human agent can ensure that any potential customer or client does not receive any unwanted responses from the NAA.

Acknowledgements

We would like to acknowledge and give thanks to Sedef Akinli Kocak and Gerald Penn from the Vector Institute for Artificial Intelligence for their help in facilitating this project. We also acknowledge the contributions of Andrew Brown from CIBC on CIBC's NAA implementation.

References

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

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Cite arxiv:1810.04805Comment: 13 pages.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Tingchen Fu, Shen Gao, Xueliang Zhao, Ji-rong Wen, and Rui Yan. 2022. Learning towards conversational AI: A survey. *AI Open*, 3:14–28.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. *NeurIPS*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Mitesh M Khapra and Ananya B Sai. 2021. A tutorial on evaluation metrics used in natural language generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 15–19.

O. Khattab and Matei A. Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Soha Khazaeli, Janardhana Punuru, Chad Morris, Sanjay Sharma, Bert Staub, Michael Cole, Sunny Chiu-Webster, and Dhruv Sakalley. 2021. A Free Format Legal Question Answering System. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 107–113, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hannah Rose Kirk, Yennie Jun, Filippo Volpin, Haider Iqbal, Elias Benussi, Frederic Dreyer, Aleksandar Shtedritski, and Yuki Asano. 2021. Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models. *Advances in neural information processing systems*, 34:2611–2624.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Gold Nanoparticles Market. 2018. Global Industry Analysis, Size, Share, Growth, Trends, and Forecast 2017–2026. *Transparency Market Research*.

Luminița Nicolescu and Monica Teodora Tudorache. 2022. Human-Computer Interaction in Customer Service: The Experience with AI Chatbots—A Systematic Literature Review. *Electronics*, 11(10):1579.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv e-prints*, page arXiv:1901.04085.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Anirudh Sundar and Larry Heck. 2022. Multimodal Conversational AI: A Survey of Datasets and Approaches. *arXiv preprint arXiv:2205.06907*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing.

Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y. S. Lam. 2021. Effectiveness of Pre-training for Few-shot Intent Classification. *arXiv preprint arXiv:2109.05782*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *ACL, system demonstration*.

A Appendix

Project participants: The participants of this project consist of:

- *Technical and project management teams from Vector Institute for AI* initiating this project with the mission to bring industry and academia closer so that the two groups could learn and benefit from each other.

- *Data scientists from three industry partners* participated to enhance their knowledge and apply new methods in their companies.

- *University faculty from partnering universities* providing additional advising and exposure to the state-of-the-art methods.

Project steps: The following are the points describing the project steps and participation.

- Vector Institute for AI:
  - defined the problem statement and project scope with the help of some input from the industry partners,
  - implemented general domain-agnostic NAA reference implementations consisting of three high-level components: (1) Intent Identification, (2) Context Retrieval, and (3) Response Generation,
  - provided tutorial, open-source datasets, computing services, training, and constant feedback & support.
- Faculty provided advising on cutting edge technologies.
- Industry partners (CIBC, KPMG, PwC):
  - identified the use-cases,
  - modified, trained/re-trained and improvised the framework on the datasets of their choice according to their use-cases and industry settings.
  - developed their own domain/industry specific end-to-end pipeline to deploy in their respective companies.

B Environment Setup

The following resources were provided by the *Vector Institute for AI* to the participants.

- **Git repos:** Git repositories containing full reference implementation code and training details were provided by the *Vector Institute for AI*[1].

- **Cluster and GPU access:** Access to cluster and GPUs was provided by the *Vector Institute for AI*. Model training was conducted on a remote cluster on one of 3 GPUs: NVIDIA T4 (16GB), NVIDIA® Tesla P100 (16GB), and NVIDIA RTX 6000 GPU (24GB)

- **Dataset storage:** All datasets used were made available on cluster. Moreover, each participant had 50GB storage space (which could be increased upon request) to store datasets, model checkpoints, and other files necessary to formulate their solutions.

- **Google Cloud Platform (GCP) access:** The participants were provided access to GCP and TPUs. Also a GCP training session from Google was hosted by the *Vector Institute for AI*. The purpose of this platform was to provide the participants with experience using cloud services to train and deploy their system.

C Reference Implementation Training Details

In this section, we provide some of the preprocessing decisions, training details, and hyperparameters for our reference implementations for replication purposes. For our implementation of the transformer-based language models we use the python libraries, Huggingface Transformers (Wolf et al., 2019) for BERT and GPT-2, and Sentence-Transformers (SBERT) (Reimers and Gurevych, 2019) for the bi-encoder and cross-encoder. We use Pytorch as the backbone of our implementations. An AdamW optimizer is used for the training of all of the models.

- **Intent Identification**: The max sequence length for BERT is set to be 300 tokens. The model is fine-tuned for 40 epochs using a batch size of 16 and a learning rate of 2e-5. We utilize a linear scheduler with a warmup ratio of 20%. We used 5% of the training set for model validation purposes, and the provided

[1] https://github.com/VectorInstitute/NAA

Banking77 test set for evaluation. Early stopping based on validation loss with a patience of 2 was used to select the best performing model. The training settings for the few shot model include using a frozen BERT encoder previously fine-tuned on Banking77 as a feature extractor for few shot classification. A new linear layer on top of the BERT encoder is trained to classify the new 15 classes from CLINC150. The linear layer is trained for 5 epochs with a batch size of 6 and a learning rate of 0.001. The rest of the training settings are similar to before.

- **Context Retrieval:** For the version of the bi-encoder fine-tuned on the ELI5 dataset, the dataset is preprocessed such that a reranker pretrained on MSMARCO re-ranks the original 7 wikipedia passages per query, and picks the most relevant passage to be part of the question-answer training pair. The base *msmarco-distilbert-base-tas-b* bi-encoder is fine-tuned on this dataset for 3 epochs with a batch size of 16. A scheduler with a warmup ratio of 10% is also used. 15% of the training data is set aside for testing. The cross encoder has the same hyperparameters, with the addition of the max sequence length being set to 512 tokens. A learning rate of 2e-5 is used for fine-tuning both modules.

- **Answer Generation:** Lastly, the GPT-2 answer generation model generates an output sequence that is a maximum of 200 tokens, at a temperature of 0.7, and with top_k and top_p values of 100 and 0 respectively. The model is trained for 5 epochs on MSMARCO using a batch size of 1 (with 8 gradient accumulation steps simulating a batch size of 8), and a learning rate of 5e-5. Furthermore, the maximum number of input tokens is set to 330. Additional settings include setting the language modelling loss coefficient to 10.0, applying gradient clipping with a magnitude of 10.0, and using a multiple choice classification head for the second head of GPT-2 with a loss coefficient of 1.0. The MSMARCO dataset is preprocessed such that we use the well-formed answer if possible, and avoiding using questions for which there are multiple answers without a well formed answer. We use the existing train and validation sets for fine-tuning and evaluating the model.

D Supplementary Figures

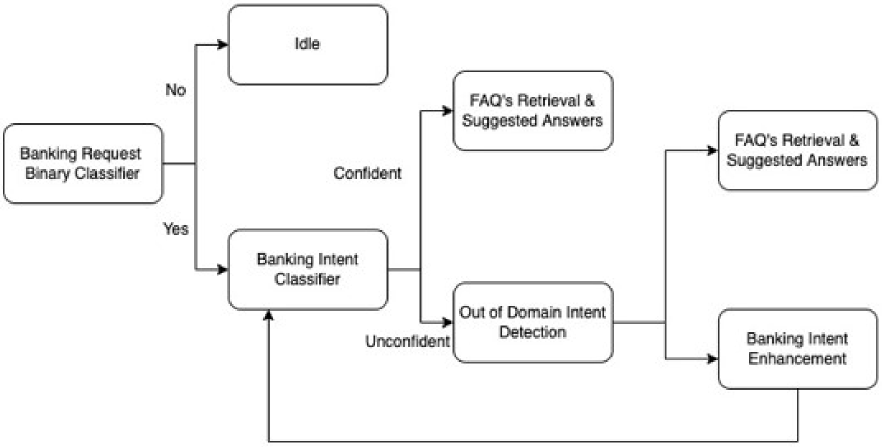
Idle
No
FAQ's Retrieval & Suggested Answers
Banking Request Binary Classifier
Confident
FAQ's Retrieval & Suggested Answers
Yes
Banking Intent Classifier
Out of Domain Intent Detection
Unconfident
Banking Intent Enhancement

Figure 2: CIBC Banking Agent Assistance Implementation Pipeline.

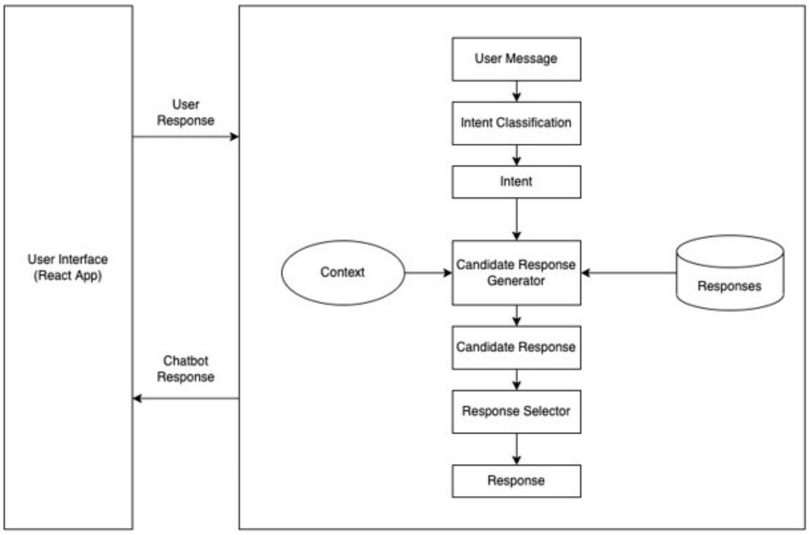

Figure 3: CIBC - Diagram showing the Banking Chatbot Service API. The user (human agent) interacts with the NAA pipeline through the React App, and sends a response. Then the NAA pipeline processes the request and a chatbot sends a response.