A Multi-Talent Healthcare AI Bot Platform

Martin Horn, Xiang Li, Lin Chen, and Sabin Kafle

Cambia Health Solutions
Seattle, WA 98101
{Martin.Horn, Xiang.Li, Lin.Chen, Sabin.Kafle}@cambiahealth.com

Abstract. AI bots have emerged in various industries hoping to simplify customer communication. However, there are a certain set of challenges for such a product in the healthcare domain, including confidentiality and domain knowledge. We discuss the implementation of a secure, multi-task healthcare chatbot built to provide easy, fast, on-demand conversational access to a variety of health-related resources.

Keywords: virtual assistant · chatbot · healthcare · microservices · natural language understanding

1 Introduction

Conversational agents (AI bots) have been widely popular in both the academic research community [4, 8, 7] and several industries in recent years. While they carry promise [2] for customer engagement, most industrial AI bots have failed to meet expectations, often due to a shallow understanding of user inputs [1, 5, 6], with up to 70% failure rates. Moreover, the domain-specific necessities for AI bots (e.g., confidentiality) add an additional challenge to the task.

Healthcare is a complicated and confusing system, often placing a large workload on customer support services. AI bots present a very useful application of facilitating and optimizing the interactions between the customers and support staff, with bots answering simple questions while deferring complex individualized questions to the support staffs in a timely fashion.

AI bots are developed either through a task-oriented approach based on the domain requirements or an end-to-end data driven approach, which is primarily used to develop social bots. While reinforcement learning has enabled the formulation of AI bot learning as a decision making process, thus providing a unified framework [3, 9] for both task-oriented and social bots, its applicability is limited for many task-oriented bots, especially in sensitive domains such as healthcare. In healthcare it is more useful to have a modular task-centric design for both flexibility and compliance.

We have built a flexible, compliant AI bot platform for the healthcare domain with the hope to overcome some of these obstacles and drive progress on practical, useful conversational agents. We primarily focus on the modular design due to its fine control of both query understanding and answering, which requires components unique to the healthcare domains such as stringent authentication criteria, knowledge of rules, and understanding of healthcare plans. It has been designed to be able to learn more intents and functionalities over time, connect to more services, and wear different personas for different applications and audiences.

2 System Architecture

The AI bot uses a robust microservice architecture (Figure 1), allowing it to be pluggable, i.e. new components and services can easily be added or replaced.

2.1 Modular Design

Each component is implemented as a standalone Docker-containerized gRPC service using JSON Web Token (JWT)-based authentication. This architecture allows services to be used and re-used independently or in conjunction to form an entire AI bot platform. Our strict service authorization ensures each service is secure and compliant in a sensitive environment containing Protected Health Information (PHI). We also use the concept of *personas*, which are bot configurations that turn on different skills or personalities for a specific product or brand. A new persona can easily be created and configured in order to create an entirely new bot on the platform.

2.2 The Pipeline

The platform is orchestrated by the Gateway, which serves as the entry-point to the bot's core functionality. Textual requests are initiated by the interchangeable User Interface (UI). We have a desktop UI for several in-house bots as well as mobile UI for a consumer facing app.

The Gateway passes the query to the Natural Language Understanding (NLU) module in order to detect the user's intent. Each intent corresponds to one app in the App Platform which performs one or more specific skills. This involves extracting useful information from the utterance via NLU, managing the dialog, and consulting a variety of knowledge base services and data stores (simplified as Knowledge Base in Figure 1). The app then sends the appropriate response to the Gateway to pass back to the UI.

In addition to facilitating the dialog pipeline, the Gateway connects to the Authorization, Monitoring, and Logging services. Authorization ensures that the services are only accessed by those with the right privileges. For example, users can only get user-specific health benefit information about themselves or dependents for which they have been granted access rights. The Monitoring service captures performance metrics about each service and sends out alerts if certain thresholds are exceeded (such as response time). The Logging service captures service events, dialogs, and feedback. Selections of securely logged conversations are then annotated in order to re-train and improve the NLU models.

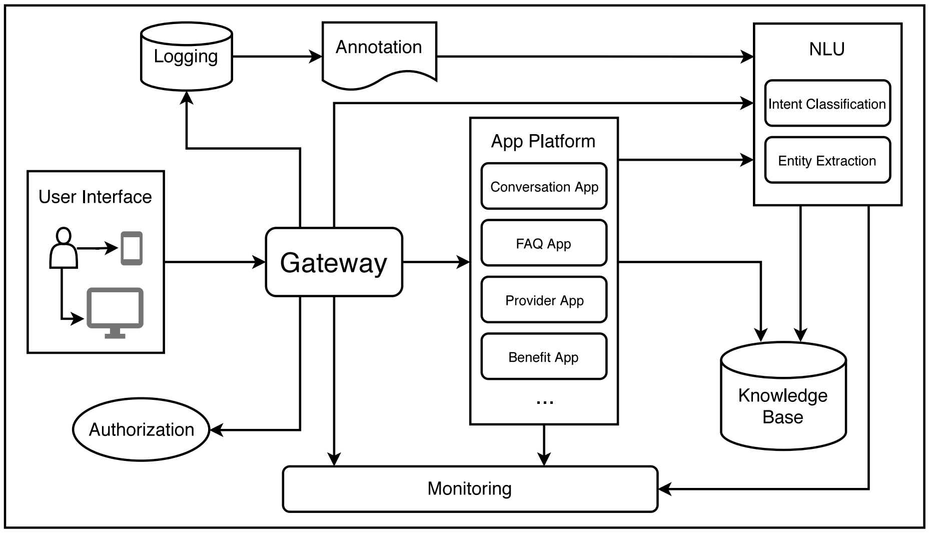

Fig. 1. System Architecture

2.3 App Platform

Our bot platform is multi-talent, meaning it supports multiple different skills and is easily extendable. Each app in the App Platform supplies its own dialog management and business logic so that each skill can be tightly managed and configured for its specific purpose. Once the app has performed its intended task, it generates a custom response depending on the type of service it provides.

Apps like the Conversation App and the FAQ (Frequently Asked Question) App use an information retrieval approach to find fitting responses from a bank of existing question-answer pairs. Others use specific pieces of information from the query to perform a task, such as finding a doctor (Provider App) or looking up insurance benefits (Benefit App). Results from the task are then filtered, sorted, and injected into a response template. Though apps have different dialog logic and response requirements, the addition of new apps is simplified by a standard template and reusable services, such as a dialog state manager and response generator.

2.4 Knowledge Base

The key to creating useful AI bot skills is having a rich knowledge source from which to look up information and draw insights. In a domain such as healthcare, deep and specific business knowledge is crucial. We have codified and operationalized healthcare provider, member, and terminology knowledge bases and connected to a variety of bot-external healthcare services for data and logic around topics like benefits, treatment cost estimation, and prescription drugs.

2.5 Natural Language Understanding

We trained custom Natural Language Understanding (NLU) models so the bot can understand domain-specific requests and handle a variety of custom-built skills. NLU consists primarily of intent classification and entity extraction.

We use a hierarchical intent classification approach in order to reduce the number of classes in each round of classification and improve overall model performance. For example, once a query is classified as Provider intent, it is further classified as either Provider Search (e.g. *Find me a dermatologist in Seattle*) or Network Eligibility Check (e.g. *Is Dr. X in-network?*).

The entity extraction module fills a variety of domain-specific slots in the dialog schema, such as location entities, provider name and specialty entities, and network status for the Provider intents. We treat this as a typical named entity recognition task where tokens are labelled with the IOB (inside-outside-beginning) scheme and then concatenate the relevant adjacent tokens into entities. After entities have been extracted, they are normalized to provide a unified format for downstream processing.

We built and tested a variety of machine learning models for NLU, including a novel joint BiLSTM+CRF intent and entity classification model. For flexibility and performance reasons on new intents and entities with small training data sets, our platform primarily uses a combination of Maximum Entropy (MaxEnt) classifiers for intent classification and Conditional Random Fields (CRF) models for entity extraction.

3 Evaluation

We evaluate our bot platform's performance in a variety of ways. During runtime, we monitor service metrics like job health, response latency, and queries per second. We also collect feedback and calculate custom usage statistics via our logging service.

It is also useful to measure accuracy in order to guarantee the platform's quality. While it is difficult to directly measure an entire bot platform's accuracy, we can isolate and evaluate certain components. Accuracy of information provided by the bot must be tailored to each individual skill. For example, we had domain experts validate in-network provider search performance by comparing results with trusted third-party tools.

However, we were able to perform traditional evaluation on the core NLU components of the system: intent classification and entity extraction. We used a typical train/test split with a hold-out test set of around 1,000 sentences for both tasks.

3.1 Intent Classification

For intent classification, we examined (P)recision, (R)ecall, and F1 score for each intent, as well as micro average for each metric and overall accuracy, found

Table 1. Intent Classification Scores

Intent	P	R	F1	#
auth	0.93	1.00	0.96	25
copay	0.89	0.96	0.92	25
schedule	0.89	0.96	0.92	25
coinsure	0.85	0.88	0.86	25
...	...	...	...	...
cost	0.92	0.62	0.74	71
glossary	0.74	0.71	0.72	41
faq	0.44	0.83	0.57	35
general	0.65	0.50	0.57	26
Avg / Total	0.83	0.81	0.81	747
Overall Accuracy	0.81			

Table 2. Entity Extraction Scores

D	Entity Type	P	R	F1
P	city	0.96	0.93	0.95
	member_id	1.00	1.00	1.00
	network_status	1.00	0.97	0.99
	facility	0.80	1.00	0.89
	practitioner	0.95	0.92	0.93
	specialty	0.96	0.90	0.92
	state	0.98	0.98	0.98
	zip_code	1.00	1.00	1.00
B	benefit_category	0.95	0.85	0.90
	member_id	1.00	0.98	0.99
G	concept	0.94	0.89	0.92

in Table 1. The hierarchical nature of the intents is flattened for purposes of evaluation, and only top and bottom four intents (ordered by F1) are displayed due to space constraints.

Results are shown for our MaxEnt models. It is clear that performance varies considerably from intent to intent. For the very low F1 of 0.57 for `faq` and `general`, this is likely due to high variability in the intents.

3.2 Entity Extraction

Separate sequence labeling models were constructed for different (D)omains of data: (P)rovider, (B)enefit, and (G)lossary. Entity extraction was evaluated on a strict-match per-entity basis. (P)recision, (R)ecall, and F1 score for each entity type are found in Table 2.

Results are shown for our CRF models. The relatively low performance of `facility` may be attributed to a lack of support in train and test data and an overlap in distribution to the `practitioner` entity type. `member_id` and `zip_code` may have received perfect F1 scores due to their low variability (they consist exclusively of numerical digits).

4 Conclusion

We introduced an AI bot architecture consisting of upgradable modules which form the foundation for building a stable AI bot with value to customers. Our system consists of a flexible language understanding module, an app platform facilitating dialogs and performing tasks, robust knowledge base services, and a gateway connecting each microservice together. The modularity of the system aids its flexibility, extendability, and compliance in the sensitive healthcare domain. In future work we would like to integrate a more expansive knowledge graph to further empower the AI bot in handling complex, personalized, domain-specific queries.

References

1. Brandtzaeg, P.B., Følstad, A.: Chatbots: Changing user needs and motivations. interactions **25**(5), 38–43 (2018)
2. Dale, R.: The return of the chatbots. Natural Language Engineering **22**(5), 811–817 (2016)
3. Fang, H., Cheng, H., Clark, E., Holtzman, A., Sap, M., Ostendorf, M., Choi, Y., Smith, N.A.: Sounding board–university of washington's alexa prize submission. Alexa Prize Proceedings (2017)
4. Gao, J., Galley, M., Li, L.: Neural approaches to conversational ai. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1371–1374. ACM (2018)
5. Jain, M., Kumar, P., Kota, R., Patel, S.N.: Evaluating and informing the design of chatbots. In: Proceedings of the 2018 on Designing Interactive Systems Conference 2018. pp. 895–906. ACM (2018)
6. Piccolo, L., Roberts, S., Iosif, A., Alani, H.: Designing chatbots for crises: A case study contrasting potential and reality (2018)
7. Serban, I.V., Lowe, R., Henderson, P., Charlin, L., Pineau, J.: A survey of available corpora for building data-driven dialogue systems. arXiv preprint arXiv:1512.05742 (2015)
8. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A.C., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. In: AAAI. vol. 16, pp. 3776–3784 (2016)
9. Zhou, L., Gao, J., Li, D., Shum, H.Y.: The design and implementation of xiaoice, an empathetic social chatbot. arXiv preprint arXiv:1812.08989 (2018)