CharmBana: Progressive Responses with Real-Time Internet Search for Knowledge-Powered Conversations

Revanth Gangi Reddy, Sharath Chandra Etagi Suresh, Hao Bai, Wentao Yao, Mankeerat Singh Sidhu, Karan Aggarwal, Prathamesh Sonawane, Chengxiang Zhai
University of Illinois at Urbana-Champaign
{revanth3,sce3,haob2,wentaoy4,mssidhu2,karana5,pks10,czhai}@illinois.edu

ABSTRACT

Chatbots are often hindered by the latency associated with integrating real-time web search results, compromising user experience. To overcome this, we present CharmBana, an innovative social chatbot that introduces the use of *progressive response generation* to effortlessly blend search results into the bot's responses, while ensuring low response latency. The use of progressive responses is especially beneficial for voice-based chatbots, where the preliminary response buys time for a detailed follow-up, ensuring a smooth user interaction. As a result, our method not only cuts down user waiting times by 50% but also generates more relevant, precise, and engaging search inquiries. When tested in the Alexa Prize Socialbot Grand Challenge 5 , our chatbot employing progressive responses consistently received higher user ratings.

CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; • **Information systems** → **Query suggestion**.

KEYWORDS

Open-Domain Dialog, Search-Based Assistants, Query Generation

ACM Reference Format:
Revanth Gangi Reddy, Sharath Chandra Etagi Suresh, Hao Bai, Wentao Yao, Mankeerat Singh Sidhu,, Karan Aggarwal, Prathamesh Sonawane, Chengxiang Zhai. 2024. CharmBana: Progressive Responses with Real-Time Internet Search for Knowledge-Powered Conversations. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Chatbots need to access external knowledge [3, 9] to be able to engage users in captivating and informative discussions on a variety of topics that aren't predetermined. Allowing a chatbot to access static external information raises concerns over the potential use of outdated or inaccurate data. This challenge can be tackled by employing a web search engine and crafting precise search queries to gather pertinent information from the web in *real-time*. However, a significant obstacle preventing current chatbots [14] from seamlessly integrating with these search engines is the delayed response time when attempting to incorporate real-time web search results. This latency arises from a sequence of actions: determining the need to search, crafting a search query, fetching the appropriate results, and then weaving them into the final response. Each of these steps has its own inherent delay, which can result in prolonged response times and potential user dissatisfaction.

In this work, we propose to tackle the issue of high response latency by using a *progressive response generation strategy*, which can effectively reduce the user's waiting time. With this strategy, a quick general response would be initially given to the user, followed by a comprehensive, search-driven follow-up response as an in-depth reply. The progressive response strategy is meant to simulate how a human makes the tradeoff between giving an immediate answer and taking the time to provide a well-thought and informed response. For voice-based chatbots, this approach is particularly suitable. As the text-to-speech component of the chatbot articulates the preliminary reply, it simultaneously allocates additional time to formulate a comprehensive follow-up, ensuring that the intermission feels natural and fluid to the user. Additionally, we utilize the initial response as an underlying guide to shape the generation of search queries. This guarantees that the generated queries aim to fetch information that aligns with the initial response, ensuring the follow-up response is consistent.

Firstly, we demonstrate that our proposed progressive response generation strategy lowers the user wait time by 50%. Empirically, we show (in §3.1) that our approach overcomes limitations of existing query generation techniques [14] that rely solely on explicit dialog information, and produces search queries that are markedly more relevant, specific, and interesting. Our chatbot can be accessed through the Alexa developer console, as depicted in Figure 4. The bot provides voice-based responses, allowing users to interact with it in real-time. Based on evaluations from participating in the Alexa Prize Socialbot Grand Challenge 5 (in §3.2), we observe that real-world Alexa users give higher ratings to our bot's conversations when they feature progressive responses than when they do not.

2 SYSTEM DESIGN AND IMPLEMENTATION

CharmBana's architecture uses the Cobot toolkit [12], constructed with a modular design for easy alterations. Cobot supports the integration of remote modules, similar to microservice architecture, for handling tasks like dialog topic tracking, external knowledge search, and LLM-based response generation. Our system's architecture is shown in Figure 1, with the below key components:

WSDM '24, March 4–8, 2024, Merida, Mexico.

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

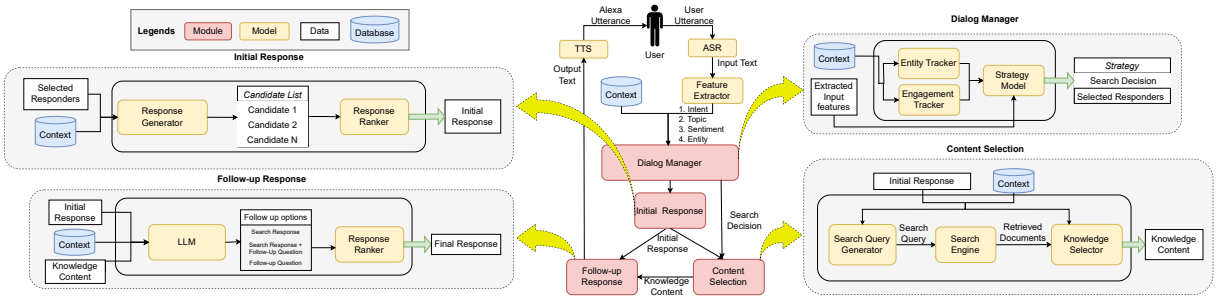

Figure 1: Overall framework for CharmBana using the *progressive responses* methodology with initial and follow-up responses.

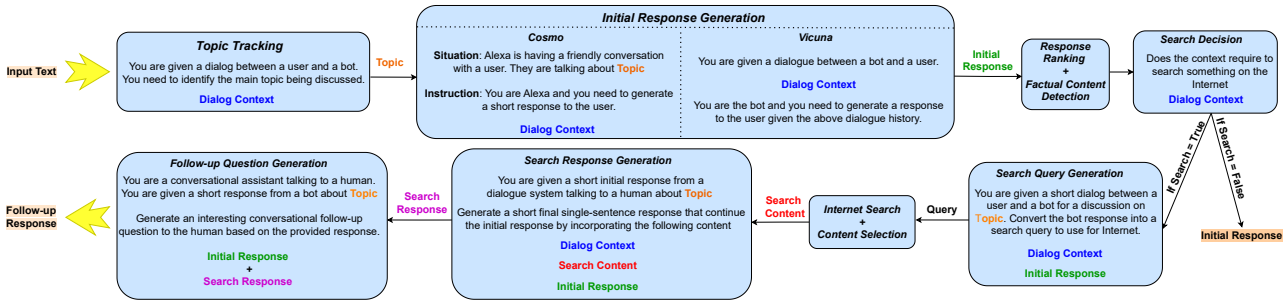

Figure 2: Flowchart showing the flow of information along various components in CharmBana via different prompts.

- **Dialog Manager**: This component orchestrates the dialog strategy, focusing on tracking topics, the selection of appropriate responders and determining whether an internet search is needed (i.e. *search decision*).
- **Initial Response Generation**: This module produces a basic yet coherent initial response, paving the way for a more informed follow-up reply.
- **Content Selection**: Employed for choosing relevant content from the internet to enhance the follow-up response, this module is activated when the search decision is true.
- **Follow-up Response Generation**: This final module integrates the extracted search information to formulate a follow-up response that builds upon the chatbot's initial response.

The diagram in Figure 2 illustrates the information flow through different components of CharmBana. Meanwhile, Figure 3 depicts the average latency for each component. Notably, when utilizing progressive responses, the average waiting time is reduced from 4.3s to 2.2s, marking a 50% improvement in response latency.

2.1 Dialog Manager

2.1.1 Topic Tracking. Topic tracking [6] aims to determine the primary subject of the discussion in free-form dialogs. While past methods [7] focused on broad topics like 'movies' or 'sport', we aim to identify specific, detailed topics like 'movie/actor names' or 'teams'. For such fine-grained topic tracking, we use an instruction-tuned model [2] to pinpoint the current topic from the dialog. Despite limited training data, our model leverages strong zero-shot skills to effectively track these detailed topics by using the prompt in Figure 2. Specifically, we use Flan-T5 large (770M) and improve its performance by training on data labeled by ChatGPT from the Wizard of Internet (WoI) [9] training set. The topic tracking output

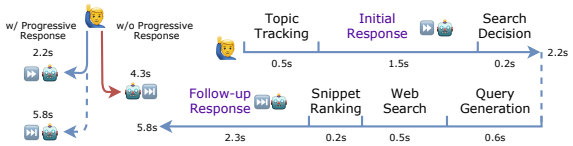

Figure 3: Latency breakdown and comparison of response times with progressive response (ours) vs without.

is incorporated into subsequent module prompts (seen in Figure 2) to ensure dialog coherence.

2.1.2 Engagement Tracking and Topic Switching. We use an engagement tracker [5] to gauge user interest and decide when to change topics. This model combines Bi-LSTM with BERT embeddings and includes an MLP classifier to assess user engagement based on previous dialog turns. To smoothly transition between topics, we create a bridge sentence linking the last bot message to the new topic. Topic switches are determined by rules based on turn count, user intent, and engagement tracker output. We continue on the topic if the user wants specific info, but switch using a special transition reply after six turns or if user engagement decreases.

2.1.3 Search Decision. The search decision module helps the chatbot know when extra information is needed, improving efficiency by reducing unneeded searches, thereby speeding up response times. To buid this model, we adopt the approach from BlenderBot 3 [14], by finetuning an encoder-decoder model with data from sources like Wizard of Internet [9] and Wizard of Wikipedia [3]. Given latency concerns, we use Flan-T5 Base (250M) for the encoder-decoder model, along with the prompt: "Does this conversation require a search query?". A threshold of 0.70 is set for search confidence, resulting in about a 50% search likelihood.

2.2 Initial Response Generation

In our progressive response generation strategy, our goal is to first swiftly generate an initial, coherent reply without relying on internet search. This initial response then guides the formulation of subsequent search query, to obtain relevant information for the follow-up response. We use neural response generators (§2.2.1), such as social commonsense dialogue systems [8] and instruction-tuned LLMs [1], to produce potential initial replies. These are then fed to the Cobot ranker to select one. Additionally, we incorporate a factuality check module (§2.2.2) to eliminate candidate responses containing potentially hallucinated or incorrect information.

2.2.1 Neural Response Generators.

Cosmo-3B: Social commonsense-based dialog systems typically demonstrate a fundamental understanding of handling diverse topics and situations. They either use knowledge graphs like ATOMIC [13] for crafting responses or distill this knowledge into LLMs for direct response generation. We use Cosmo [8], a 3B parameter LLM trained on 1.5M socially-grounded synthetic dialogs generated by prompting InstructGPT [11] using contextualized commonsense knowledge from ATOMIC. Cosmo takes a situation narrative and role instruction as input, and generates a response based on the dialog context. We also incorporate the topic tracking output into the situation narrative, as shown in Figure 2.

Vicuna-13B: We employ Vicuna-13B [1] as an instruction-tuned response generator for open-domain dialog in zero-shot scenarios, guided by the prompt in Figure 2. It is built on the Wizard dataset [15] and expands on ChatGPT 3.5's capabilities, by incorporating roughly 70K conversations from ShareGPT for training.

2.2.2 Factuality Check. LLM-based generators tend to produce hallucinatory content that may contain factual inaccuracies. To tackle this promptly, we employ a 'safer' approach of filtering out responses with excessive new entities. We keep track of entities in the conversation history to recognize new ones in replies. For numbers, we use regex, and for proper nouns, we apply the Cobot toolkit NER tagger. Responses introducing over two new proper nouns or more than one numerical entity are discarded. After analyzing 850 conversations, it was observed that 7.8% of instances (4 out of 850 for Cosmo-3B and 66 out of 850 for Vicuna-13B) were flagged by the factuality-check module for potential inaccuracies. Upon human review of 40 such instances, 32% were false positives.

2.3 Content Selection

2.3.1 Search Query Generation. Given the dialog context, conversation topic and a latent directive in the form of an initial response, our goal is to generate a search query to obtain relevant information for continuing the conversation. We use an instruction-tuned model [2] for query generation, by prompting (see Figure 2) it to transform the initial response into a search query. We incorporate the fine-grained topic into the prompt to improve relevance and specificity of the search queries. In particular, we employ Flan-T5 large (770M) for the purpose of query generation. We use ChatGPT to obtain silver labels for search queries using the Wizard of Internet (WoI) training set [9]. To create finetuning data, we chose turns corresponding to internet search from WoI, yielding 20k examples.

Figure 4: A hypothetical dialog situation showing the image from web search for a discussion about Indian cuisine.

2.3.2 Internet Search. To obtain relevant snippets for the given search query, we use the Bing Search API to get the top-5 web results. With latency in mind, we directly rely on the concise snippets provided by Bing as the relevant information. We then employ a re-ranker [4] to identify the most relevant snippet. It is worth noting that relying solely on the top-ranked search result can be misleading, as the Bing API occasionally prioritizes sponsored ads. Furthermore, the top image retrieved by the Bing Image API is displayed on the interaction screen, with an example in Figure 4.

2.4 Follow-up Response Generation

As the voice-based chatbot 'reads out' the initial response to the user, it 'buys' time to formulate a more comprehensive follow-up response. This is a continuation of the initial response, with the integration of relevant results that were obtained using internet search. The follow-up response itself is structured into: (a) a search response formulated using the search result, and (b) a follow-up question designed to promote user engagement. Using the prompts described in Figure 2, the system crafts several follow-up questions based on the combination of initial response and search response. The Cobot ranker then evaluates these to identify the most relevant one. We leverage Vicuna-13B as the instruction-tuned LLM for both search response and follow-up question generation.

3 EXPERIMENTS AND ANALYSIS

3.1 Performance of Query Generation

To evaluate the quality of generated search queries, we focus on the WoI test set with dialog turns that had queries annotated for generating responses. Using an intent detection model [7], we identify and remove turns related to information or opinion requests, and randomly selected 200 examples for human evaluation. We conducted a human study with four experienced NLP students to evaluate the quality of generated search queries. Queries were assessed based on relevance, specificity, usefulness, and potential to maintain user engagement in the dialog. Recent studies, like G-EVAL [10], show that LLMs such as GPT-4 can effectively evaluate natural language generations and align well with human assessments. Therefore, we utilize GPT-4 for automatic evaluation, prompting it to provide an overall score (ranging from 1-10) for search queries.

Table 1 shows the results of human and automatic evaluation of search queries. Mainly, we notice that instruction-tuned models outperform Blender Bot 3 significantly, and using Cosmo's commonsense response as a directive for guiding query generation with Flan T5 shows consistent improvements. Additionally, substantial

Query Generation Approach	Type	Human				Automatic
		Rel.	Spe.	Use.	Int.	GPT-4
Blender Bot 3	Finetuned	3.13	2.29	2.61	2.28	35.1
Flan T5 w/o Cosmo	Zero-shot	3.38	3.21	3.06	3.02	44.3
Flan T5 with Cosmo (Ours)	Zero-shot	3.59	3.51	3.39	3.29	49.9
	Finetuned	**4.16**	**4.05**	**3.98**	**3.91**	**72.2**
ChatGPT	Zero-shot	4.51	4.49	4.48	4.45	80.7

Table 1: Evaluation of different query generation approaches on the WoI dataset. The acronyms for human evaluation are: Relevance, Specificity, Usefulness, Interestingness.

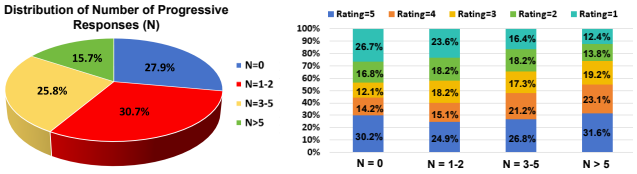

Figure 5: Distribution of number of progressive responses N in a conversation (left) and user ratings based on N (right).

enhancements in query quality are observed upon fine-tuning the zero-shot system with ChatGPT annotations. By computing the Spearman correlation between automatic metrics (GPT-4) and overall score for human evaluations (average of four aspect ratings), we found a strong correlation (0.674) between the two measures.

3.2 Analysis of Progressive Responses

CharmBana incorporates progressive response generation to be able to leverage web search while ensuring a low user wait time. Here, we analyze the direct impact of these progressive responses on our bot's performance. This assessment stems from ratings received during the Alexa Prize Socialbot Grand Challenge 5 in June'23, where Alexa users evaluated our bot's conversations on a scale of 1-5.

Firstly, we show how frequently progressive responses occur in conversations with our bot, by grouping conversations based on the count of progressive responses: 0, 1-2, 3-5, or more than 5. From figure 5 (left), we see that progressive responses are used in at least 72% of the conversations, with 15.7% having more than 5. Next, figure 5 (right) reveals that as the progressive response frequency rises, there is a decline in % of conversations receiving lower ratings (either 1 or 2). Further, conversations with over 5 progressive responses have 55% rated highly (4 or 5), versus 45% for those without. Figure 6 (left) validates this trend, as it can be seen that conversations with more progressive responses generally get higher ratings. Moreover, we analyze the impact of progressive responses by controlling for number of dialog turns (e.g., <5, 6-10, and so on). Figure 6 (right) shows the average ratings for these categories, differentiating between conversations without any progressive responses and those with at least one. Predictably, we see an uptrend in the ratings as the number of dialog turns increases. Importantly, conversations with at least one progressive response consistently outperform those without.

4 CONCLUSION

In this work, we introduce progressive response generation as a general means to seamlessly integrate real-time web search into

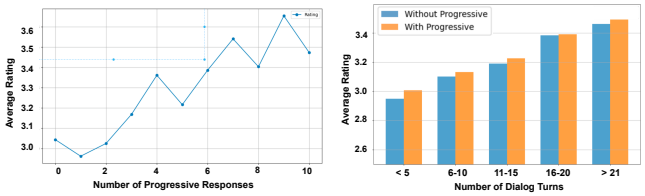

Figure 6: Average rating vs no. of progressive responses (left) and comparison of ratings for conversations with and w/o progressive responses for varying no. of dialog turns (right).

chatbots, while ensuring swift response times. Our approach significantly lowers user wait time and leads to higher ratings by real-world Alexa users. Anticipating future chatbots' need to access real-time web information, we expect them to frequently use search engines. Our proposed progressive response strategy can therefore address the latency issue for various intelligent task agents.

REFERENCES

[1] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[2] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

[3] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. [n. d.]. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *International Conference on Learning Representations*.

[4] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 7780–7788.

[5] Sarik Ghazarian, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. 2020. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7789–7796.

[6] Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. 2018. Topic-based evaluation for conversational bots. *arXiv preprint arXiv:1801.03622* (2018).

[7] Chandra Khatri, Rahul Goel, Behnam Hedayatnia, Angeliki Metallinou, Raefer Gabriel, and Arindam Mandal. 2018. Contextual topic modeling for dialogue systems. In *SLT 2018*. https://www.amazon.science/publications/contextual-topic-modeling-for-dialogue-systems

[8] Hyunwoo Kim, Jack Hessel, Liwei Jiang, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, et al. 2022. SODA: Million-scale Dialogue Distillation with Social Commonsense Contextualization. *arXiv preprint arXiv:2212.10465* (2022).

[9] Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-Augmented Dialogue Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 8460–8478.

[10] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634* (2023).

[11] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[12] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604* (2018).

[13] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3027–3035.

[14] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188* (2022).

[15] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244* (2023).