A Reference Model for Dialog Management in Conversational Agents in High-Engagement Use Cases

Nima Samsami, Stephan Kurpjuweit

Hochschule Worms – University of Applied Sciences
samsami@hs-worms.de
kurpjuweit@hs-worms.de

Abstract. The objective of our research is to systematically derive a refined reference model for the dialog management component of a conversational agent. Firstly, we characterize high-engagement conversational agents and derive solution strategies to address this class of agents. Secondly, we propose a set of conceptual components that refines the dialog management component and addresses the solution strategies. Thirdly, we survey implementation approaches for the individual components of the reference model.

Keywords: Conversational Agent; Dialog Management; Natural Language Understanding; Natural Language Generation

1 Introduction

With the general availability of smart speakers since 2017, conversational agents have gained increased popularity among consumers [1]. Based on our experience, the use cases of conversational agents in the consumer domain can be characterized as either "task-oriented use cases" or "high-engagement use cases".
The objective of our research is to systematically derive a reference model for the dialog management component of a conversational agent from the requirements of high-engagement use cases. Our research is based on the following approaches: We characterize high-engagement conversational agents (section 2) and derive solution strategies to address this class of agents (section 3). Then we propose a set of conceptual components that refines the dialog management component and addresses the solution strategies and survey implementation approaches for the individual components of the reference model (section 5).

2 Quality characteristics of high-engagement conversational agents

Based on our experience, the use cases of conversational agents in the consumer domain can be characterized as either "task-oriented" or "high-engagement": For task-oriented use cases the objective is to answer the users' information needs or to complete a task in as few conversational turns as possible. Example domains include banking, customer service or directory services. As the user wants to 'get a job done', satisfying the following two quality characteristics is essential:

(1) Relevance and (2) Focus: By nature, the bandwidth of conversational agents (i.e. the amount of information that can be communicated to the user per time) is small compared to other - esp. screen-based - digital channels like web or mobile apps. Thus,

conversational agents must deliver relevant and focused responses and reduce the number of conversational turns users have to take (‘get to the point‘).

For high-engagement use cases the objective is to keep the user engaged in the conversation for as long as possible. Example domains include media, news, entertainment or conversational commerce. In the context of our research, the level of user engagement is characterized by (a) how often the user starts a session with the agent per time [2], (b) how much time the user spends per session [3], and (c) for how long the user is active overall (customer lifetime).

While focus and relevance is central to all conversational agents, it may not be enough to ensure a high level of user engagement. Depending on the concrete nature of the agent, other quality characteristics should be taken into consideration, including:
(3) Variety: Agent should provide a natural, varied language and avoid repetitive phrases (’don’t bore me).
(4) Topicality: Agents should provide pieces of information that are current and new to a user, so that users frequently feel the need to engage with the agent (’satisfy my curiosity’).
(5) Discoverability: Agents should suggest follow-up actions that may be of interest to the user (’show me what else you can do for me’).
(6) Adaptability: Agents should be personalized and adapt to the user’s needs over time (’become my companion’).

3 Solution strategies for high-engagement conversational agents

To address the quality characteristics of the high-engagement conversational agents outlined above, concrete solution strategies have to be implemented. The following list describes generic solution strategies which we expect to be beneficial for most high-engagement conversational agents:
(1) Text variation generation: To avoid repetitive phrases, text variations should be generated (ideally automatically).
(2) Personalized content: Personalized and current content should be selected and delivered to the user.
(3) Education: Short messages that explain additional features and follow-up actions should be delivered to the user.
(4) Graceful error handling / disambiguation: Instead of entering error flows, the agent should try to understand the user’s intent, e.g. via disambiguation.
(5) Context-awareness: The agent should adapt to the usage context. For example, the agent may decide to deliver a longer response of the user is driving in a car.
(6) Modular responses: To deliver varied responses, personalized content, educational messages, etc. the response should be composed of text fragments in a flexible way.

4 Conversational agent reference architecture

Figure 1 shows a well-adapted reference architecture for conversational agents, which decomposes the dialog management component into four sub-components. This architecture serves as a basis for the refined reference model in section 5:
(1) Natural Language Understanding (NLU): Identifies and parses a user’s text input to obtain semantic tags that can be understood by computers, such as entites and intents [4].
(2) Conversational State (CS): Maintains the current conversation state based on the

conversation history. The conversation state is the cumulative meaning of the conversation history, which is generally expressed as slot-value pairs.
(3) Conversational Flow (CF): Outputs the next system action based on the current conversation state.
(4) Natural Language Generation (NLG): Converts system actions to natural language output [5].

We select this modular architecture over an end-to-end architecture (see [6]). End-to-end architectures are based on successes of deep learning approaches in recent years. The system consists of a large neural network that handles all tasks such as NLU, NLG, CF, etc. This model is still being explored and is as yet rarely applied in the industry [6]. Although the trend is toward end-to-end systems, these approaches are still limited and cannot clearly outperform the traditional methods [7]. In practice, it may not be feasible to implement a specific agent solely based on an end-to-end architecture due to a lack of training data.

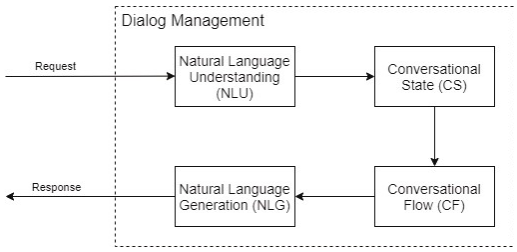

Fig. 1. Modular structure of dialog system

5 Reference model for high-engagement agents

Based on the strategies outlined in section 3, we derive a reference model for high-engagement agents. The reference model is a based of the reference architecture described in section 4. The reference model consists of a set of conceptual software components (see figure 2). Some components are optional and not required for all agents.

For the decomposition of the dialog management component we apply the following criteria: (1) Each component has a manageable set of responsibilities. (2) The implementation approaches for the components can be chosen - to a large degree - independently from each other. (3) The model can be implemented on the basis of existing conversational AI platforms and frameworks like RASA [8] or IBM Watson [9]. Available implementation approaches for the components typically range from traditional (e.g., rule-based) approaches to more sophisticated machine learning (ML)-based approaches. In practice, it may not be feasible to implement a specific agent solely based on ML approaches due to a lack of training data or development budget. Thus, in our opinion the latter criterion is important to allow for a selective component-by-component migration of a traditional implementation towards a ML-based implementation.

For each component we describe its responsibilities and survey possible implementation approaches. Figure 2 lists the conceptual software components. Flows between the components are omitted for clarity.

5.1 Component: Conversational Memory

Responsibilities: One limitation of conversational agents is that they cannot go back and forth in a conversation. This makes natural and dynamic communication between humans and computers difficult. A conversation can be carried across multiple topics. To do this, the agent must store what it has already talked about.
(1) Usage History: The entire usage history of a user stored for analytics purposes.
(2) Session State: The usage history of the current conversational session is stored with the goal to determine the current conversational context, i.e. which pieces of information a user request may refer to.
Approaches: The literature contains descriptions of many models of conversational memory. These models mainly seek to reflect how the human brain implements memory. Elvir et al. also describe an Episodic Memory Architecture to address this problem. [10] Vinkler et al. present an architecture consisting of two memory types. A short-term memory to understand the context and a long-term memory to allow the conversational agent to refer to previous information in the conversation [11].

5.2 Component: Personalization & Context-Awareness

Responsibilities: The personalization and context-awareness component accesses the usage history and calculates context information that may be required to interpret a new user request and determine the response. There are multiple flavours of conversational context which can be addressed by individual sub components:
(1) Personal Preferences: The personal preferences capture which intents and entities the user is especially interested in. These may be set explicitly or derived from the usage history. Personal preferences can be used to prioritize the text fragments selected for a user.
(2) Usage Context: The usage context captures aspects like the time of day, the usage environment (at home, in car, etc.), the device type (smartphone, smart speaker, etc.), and the interface type (chat, voice, multi model, etc.), which all may impact the response delivered to the user.
(3) Emotion Detection: The human being is an emotional being. Each person conditioned by his emotions and type expresses himself differently. To carry out a pleasant communication, it is therefore important to address the emotional intelligence aspect of communication.
Approaches: Hao et al. present a method for using content-consistent conversation to also engage in emotion-consistent communication. Emotional Chatting Machine (ECM) addresses this factor with three new mechanisms that respectively (1) model high-level abstraction of emotion expressions by embedding emotion categories, (2) capture the change of implicit internal emotion states, and (3) use explicit emotion expressions with an external emotion vocabulary. [12]

5.3 Component: Conversational Flow

Responsibilities: Check if all pieces of information to answer the user request are available (intent, slot values, context information) with sufficiently high confident values and decide whether to (1) answer the user request (standard path), (2) ask a disambiguation / clarification question (esp. if indicated by the entity disambiguation detection component), (3) enter an error handling flow or (4) hand over the conversational flow to a human (optional). Especially check if the input makes sense in the context of the current

conversational state (e.g., if the agent is waiting for a response to a specific question) (5) Conversational State: The conversational state captures "what the conversation has been about" so far, so that the user can refer to entities mentioned in previous conversational turns and ask follow-up questions. The conversational state also determines if a specific type of input is expected in the upcoming conversational turn.

(1) Intent Ambiguity: aims to clarify the semantics of an Intent in context by finding the most appropriate meaning from a predefined Intent.
(2) Entity Ambiguity: Beyond word sense disambiguation, a word can mean something different in different contexts. E.g. Mars, Galaxy and Bounty are all delicious. It is difficult for an algorithm to figure out if it is talking about an astronomical structure or chocolate tokens.
(3) Conversational State: Maintains the current Conversational state based on the conversation history. The conversation state is the cumulative meaning of the conversation history, which is generally expressed as slot-value pairs.

Approaches: Decisions can be made rule-based. Decision criteria are the request data, the conversational state, and the confidence levels. The rules can be specified as part of the language model. Jan-Gerrit Harms et al. define Dialog Management as a component of Conversational Agents that processes the dialog context and determines the agent's next action [13]. Yinpei Dai et al. kategorisieren Dialog Management in three Generations. a) The first-generation dialog systems were mainly rule-based. b) Second-generation dialog systems driven by statistical data (hereinafter referred to as the statistical dialog systems) emerged with the rise of big data technology. At that time, reinforcement learning was widely studied and applied in dialog systems. A representative example is the statistical dialog system based on the Partially Observable Markov Decision Process (POMDP) proposed by Professor Steve Young of Cambridge University in 2005 c) third-generation dialog systems built around deep learning have emerged. These systems still adopt the framework of the statistical dialog systems, but apply a neural network model in each module [6]. In general, third-generation dialog systems are better than second-generation dialog systems, but a large amount of tagged data is required for effective training. Therefore, improving the cross-domain migration and scalability of the model has become an important area of research [6]. To solve the problems of domain dependency in end-to-end systems, Lu Chen et al. propose to use a multi-agent system, where the tasks are passed from a domain specialized agent to an agent trained on another domain [14] . Jan-Gerrit Harms et al. show a taxonomy of the approaches for managing dialogs and a classification of a selection of tools. [13]
Ambiguities can be determined by analyzing the entity and synonym lists of the language model. The problem can also be addressed using entity linking (EL). EL aims to resolve such ambiguities by establishing an automatic reference between an ambiguous entity mention/span in a context and an entity (persons, locations, organization, etc.) in a knowledge base. [15] Neural networks are used for this purpose as end2end systems [16] or in conjunction with ontologies [17] [18]. Sevgili et al. use graph embeddings as an efficient method [15]. María G Buey et al. present a method that work even if the ontology is not known at training time [19].

5.4 Component: Response Generation

Responsibilities: Decide which types of text fragments to include in the response and in which order, request the individual text fragments from the text generation components, build the response to the user by concatenating the text fragments.

(1) Response Assembly: Decide which types of text fragments to include in the response and in which order, request the individual text fragments from the text generation components, build the response to the user by concatenating the text fragments

(2) Text Fragment Generation: Generate the natural language response of a specific type. The response types are usually specific for the intent at hand. However, there are response types that can be used across intents. Examples include: Disambiguation / clarification questions, error messages and educational messages (which suggest additional features to the user).

(3) Text Variation Generation: This module ensures that the texts vary based on the situation and the course of the conversation to enable a dynamic conversation. It avoids that always the same answers follows to the same questions.

(4) Education: Helps the user to learn how to use the agent from agent itself and improve his experience with the agent.

(5) Personal Recommendations: Through entertainment history and usage of the agent, the agent learns more about the user and can include this information in the answer. E.g. in the form of interesting facts.

Approaches: Decision which types of text fragments to include can be made rule-based. Decision criteria are the intent and the context information (esp. the conversational state). The component may query the text generation components upfront to figure out, if a new text fragment of a specific type is available.
Text generation can be done rule-based by filling in data from a structured data source into text templates for individual sentences and concatenating the sentences. Traditional language generation methods are based on pipelines, such as the well-known standard Architecture six Component Pipeline, which was originally proposed by Reiter [20] and has been further developed by others. This includes the following stand-alone components: (1) Content Determination [21] (2) Document Structuring [22] (3) Lexicalization [23] (4) Referring expression generation [24] (5) Sentence aggregation [25] (6) Linguistic realization [20] for this module exists different flavors: Hand-coded grammar-based systems, Templates and Statistical Approaches [5] New approaches are based on deep learning. Santhanam et al. divide these into four categories [5] (1) Language Models [26] (2) Encoder-Decoder Archiecture [27] (3) Memory Networks [28] (4) Transformer Models [29]. Lowe et al. present a system for high engaging dialog generation [30].

6 Conclusion

In this contribution we propose a reference model for the dialog management component of a conversational agents which addresses high-engagement use cases.
The reference model may serve as a basis for multiple tasks, especially: system design (as a starting point to design both individual agents and agent creation platforms), system evaluation (as a structure to evaluate and compare agent creation platforms), and research (as a framework to structure future research projects and to put individual research contributions in context).

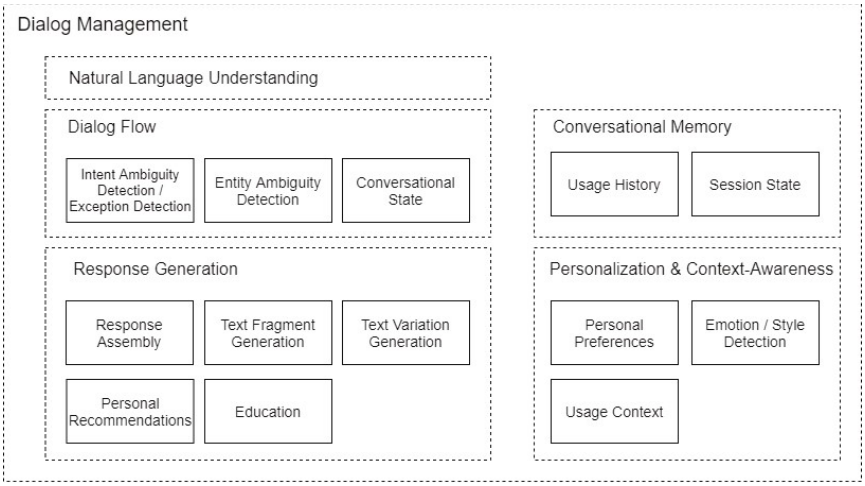

Fig. 2. Dialog Management Refernce Model (Conceptual Sub-components)

References

1. Adamopoulou, E., Moussiades, L.: Artificial Intelligence Applications and Innovations, 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II. IFIP Advances in Information and Communication Technology (2020) 373–383
2. Moore, R.J., Arar, R.: Conversational Ux Design: A Practitioner's Guide to the Natural Conversation Framework. Illustrated edition edn. ACM Books (5 2019)
3. Mandryk, R., Hancock, M., Perry, M., Cox, A., Porcheron, M., Fischer, J.E., Reeves, S., Sharples, S.: Voice Interfaces in Everyday Life. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (2018) 1–12
4. Peng, B., Li, X., Gao, J., Liu, J., Wong, K.F., Su, S.Y.: Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning. arXiv (2018)
5. Santhanam, S., Shaikh, S.: A Survey of Natural Language Generation Techniques with a Focus on Dialogue Systems - Past, Present and Future Directions. arXiv (2019)
6. Dai, Y., Yu, H., Jiang, Y., Tang, C., Li, Y., Sun, J.: A Survey on Dialog Management: Recent Advances and Challenges. arXiv (2020)
7. Chernyavskiy, A., Ilvovsky, D., Nakov, P.: Transformers: "The End of History" for NLP? arXiv (2021)
8. Bhattacharyya, S., Ray, S., Dey, M.: Proceedings of the Global AI Congress 2019. Advances in Intelligent Systems and Computing (2020) 303–318 URAI21: Context-Aware Conversational Agent for a Closed Domain Task.
9. IBM: Conversational chatbot reference architecture
10. Elvir, M., Gonzalez, A.J., Walls, C., Wilder, B.: Remembering a Conversation – A Conversational Memory Architecture for Embodied Conversational Agents. Journal of Intelligent Systems **26**(1) (2017) 1–21
11. Vinkler, M.L., Yu, P.: Conversational Chatbots with Memory-based Question and Answer Generation. PhD thesis (11 2020)
12. Zhou, H., Huang, M., Zhang, T., Zhu, X., Liu, B.: Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. arXiv (2017)

13. Harms, J.G., Kucherbaev, P., Bozzon, A., Houben, G.J.: Approaches for Dialog Management in Conversational Agents. IEEE Internet Computing **23**(2) (2018) 13–22
14. Chen, L., Chen, Z., Tan, B., Long, S., Gasic, M., Yu, K.: AgentGraph: Towards Universal Dialogue Management with Structured Deep Reinforcement Learning. arXiv (2019)
15. Sevgili, O., Panchenko, A., Biemann, C.: Improving Neural Entity Disambiguation with Graph Embeddings. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop (2019) 315–322
16. Kolitsas, N., Ganea, O.E., Hofmann, T.: End-to-End Neural Entity Linking. Proceedings of the 22nd Conference on Computational Natural Language Learning (2018) 519–529
17. Gracia, J., Mena, E.: Multiontology semantic disambiguation in unstructured web contexts. In: Proceedings of the 2009 K-CAP Workshop on Collective Knowledge Capturing and Representation. 1–9
18. Distante, D., Faralli, S., Rittinghaus, S., Rosso, P., Samsami, N.: DomainSenticNet: An Ontology and a Methodology Enabling Domain-Aware Sentic Computing. Cognitive Computation (2021) 1—16
19. Buey, M.G., Bobed, C., Gracia, J., Mena, E.: Semantic Relatedness for Keyword Disambiguation: Exploiting Different Embeddings. arXiv (2020)
20. Reiter, E., Dale, R.: Building Natural Language Generation Systems. (2000) 23–40
21. Konstas, I., Lapata, M.: Unsupervised Concept-to-text Generation with Hypergraphs. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Montréal, Canada, Association for Computational Linguistics (6 2012) 752–761
22. Dimitromanolaki, A., Androutsopoulos, I.: Learning to Order Facts for Discourse Planning in Natural Language Generation. arXiv (2003)
23. Gatt, A., Krahmer, E.: Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research **61** (2018) 65–170
24. Engonopoulos, N., Koller, A.: Generating effective referring expressions using charts. Proceedings of the INLG and SIGDIAL 2014 Joint Session 162–171
25. Barzilay, R., Lapata, M.: Aggregation via set partitioning for natural language generation. Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics - (2006) 359–366
26. Ghosh, S., Chollet, M., Laksana, E., Morency, L.P., Scherer, S.: Affect-LM: A Neural Language Model for Customizable Affective Text Generation. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2017) 634–642
27. Ke, P., Guan, J., Huang, M., Zhu, X.: Generating Informative Responses with Controlled Sentence Function. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2018) 1499–1508
28. Wang, P., Wu, Q., Shen, C., Dick, A., Hengel, A.v.d.: FVQA: Fact-Based Visual Question Answering. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(10) (2016) 2413–2427
29. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv (2018)
30. Lowe, R., Noseworthy, M., Serban, I.V., Angelard-Gontier, N., Bengio, Y., Pineau, J.: Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. arXiv (2017)