Mobile Chatbot for Information Search

Manjula Devi C[1], Santhosh N R C[2], Suraj Kumar R C[3], Thageer Ashraf S[4]

[1]Assistant Professor, Dept. of Information Technology, K.L.N. College of Engineering, Tamil Nadu, India
[2,3,4]Student, Dept. of Information Technology, K.L.N. College of Engineering, Tamil Nadu, India

---***---

Abstract - *A chatterbot or chatbot aims to make a conversation between both human and machine. It can also be built as a question answering system. A chatbot for student score searching is implemented in this system. To process the user requests and to provide efficient information the chatbot is developed with different services and skills. Watson Assistant service is the main service which acts as core of the chatbot. Watson Assistant service has the skeleton of conversations with intents and entities. Input is categorized with the intent and delivers output based on available entities. The entire knowledge of chatbot is stored in a SQL database which is known as knowledge base. The input from user can be fetched by Speech to Text service. Likewise, output is displayed in text as well as with voice by Text to Speech service. To develop intuitive and self-explanatory user interface for the user android studio is used with java language.*

***Key Words*: Chatbot, Watson Assistant, Text to Speech, Speech to Text, SQL, Knowledge base.**

1. INTRODUCTION

A chatbot is a piece of software that conducts a conversation via voice or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner. Chatbots are typically used in dialog systems for various practical purposes including customer service or information acquisition. Some chatbots use sophisticated natural language processing systems, but many simpler ones scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database. These are the most common usages of chatbots.

One of the platforms in which chatbots can be developed is IBM Watson. It provides multiple services and skills to build a chatbot for the users need. Watson is a question-answering computer system capable of answering questions posed in natural language, developed in IBM's DeepQA project. Some of the important services that is used in this system are Watson assistant service, Speech to Text service, Text to Speech service. These services are briefly explained in the following module descriptions.

Skills are conversation skeletons which are developed as a JSON file and integrated with Watson assistant. The JSON file includes all the possible conversational vocabularies based on the service the chatbot provides. The words are picked based on intents and entities. Intents are decided by the input given from user, whether it's a question, positive feedback, negative criticism. This is followed by the entity which has the possible replies for the user. Intent and Entity is also discussed fully in the module description.

Knowledge base is a database in which the values for user's questions is stored. This knowledge base is accessed by SQL query language. Knowledge base can be stored locally in the application storage or it can also be stored as a cloud. The cloud option can be implemented by IBM cloud services. Either option will hold the values that are related to the application safely for the use of chatbot by the user.

Mobile application is the part where the user and the chatbot is interacting with each other. It contains intuitive interface that is easy to be used by all the users without any confusion about chatbot interaction. Which means the chatbot application must be self-explanatory without any user manual or others guidance. For this purpose, android studio comes in place. IBM Watson services are integrated within java program which is built and run by android studio.

2. RELATED WORK

The authors H.Honda and M.Hagiwara[1] propose methods to learn symbolic processing with deep learning and to build question answering systems by means of learned models. Symbolic processing, performed by the Prolog processing systems which execute unification, resolution, and list operations, is learned by a combination of deep learning models, Neural Machine Translation (NMT) and Word2Vec training. To our knowledge, the implementation of a Prolog-like processing system using deep learning is a new experiment that has not been conducted in the past. The results of their experiments revealed that the proposed methods are superior to the conventional methods because symbolic processing (1) has rich representations, (2) can interpret inputs even if they include unknown symbols, and (3) can be learned with a small amount of training data. In particular (2), handling of unknown data, which is a major task in artificial intelligence research, is solved using Word2Vec. Furthermore, question answering systems can be built from knowledge bases written in Prolog with learned symbolic processing, which, with conventional methods, is extremely difficult to accomplish. Their proposed systems can not only answer questions through powerful inferences by utilizing facts that harbor unknown data but also have the potential to build knowledge bases from a large amount of data, including unknown data, on the Web. The proposed systems are a completely new trial, there is no state-of-the-art methods in the sense of ``newest''. Therefore, to evaluate their efficiency, they are compared with the most traditional and robust system i.e., the Prolog system. This is new research that encompasses the subjects of conventional artificial intelligence and neural network, and their systems have higher potential to build applications such as FAQ chatbots, decision support systems and energy efficient estimation using a large amount of information on the Web. Mining hidden information through these applications will provide great value.

The author J.R.Quinlan[3] suggests the technology for building knowledge-based systems by inductive inference from examples has been demonstrated successfully in several practical applications. This paper summarizes an approach to synthesizing decision trees that has been used in a variety of systems, and it describes one such system, ID3, in detail. Results from recent studies show ways in which the methodology can be modified to deal with information that is noisy and/or incomplete. A reported shortcoming of the basic algorithm is discussed and two means of overcoming it are compared. The paper concludes with illustrations of current research directions.

The authors Stephen Muggleton[4] and Luc de Raedt explain that Inductive Logic Programming (ILP) is a new discipline which investigates the inductive construction of first-order clausal theories from examples and background knowledge. We survey the most important theories and methods of this new field. First, various problem specifications of ILP are formalized in semantic settings for ILP, yielding a "model-theory" for ILP. Second, a generic ILP algorithm is presented. Third, the inference rules and corresponding operators used in ILP are presented, resulting in a "proof-theory" for ILP. Fourth, since inductive inference does not produce statements which are assured to follow from what is given, inductive inferences require an alternative form of justification. This can take the form of either probabilistic support or logical constraints on the hypothesis language. Information compression techniques used within ILP are presented within a unifying Bayesian approach to confirmation and corroboration of hypotheses. Also, different ways to constrain the hypothesis language or specify the declarative bias are presented. Fifth, some advanced topics in ILP are addressed. These include aspects of computational learning theory as applied to ILP, and the issue of predicate invention. Finally, we survey some applications and implementations of ILP. ILP applications fall under two different categories: first, scientific discovery and knowledge acquisition, and second, programming assistants.

The author D. Bahdanau[6] suggests that Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-) alignments found by the model agree well with our intuition.

Author R.Higashinaka[8] says the paper proposes a corpus-based approach for answering why-questions. Conventional systems use hand-crafted patterns to extract and evaluate answer candidates. However, such hand-crafted patterns are likely to have low coverage of causal expressions, and it is also difficult to assign suitable weights to the patterns by hand. In our approach, causal expressions are automatically collected from corpora tagged with semantic relations. From the collected expressions, features are created to train an answer candidate ranker that maximizes the QA performance with regards to the corpus of why-questions and answers. NAZEQA, a Japanese why-QA system based on our approach, clearly outperforms a baseline that uses hand-crafted patterns with a Mean Reciprocal Rank of 0.305, making it presumably the best-performing fully implemented why-QA system.

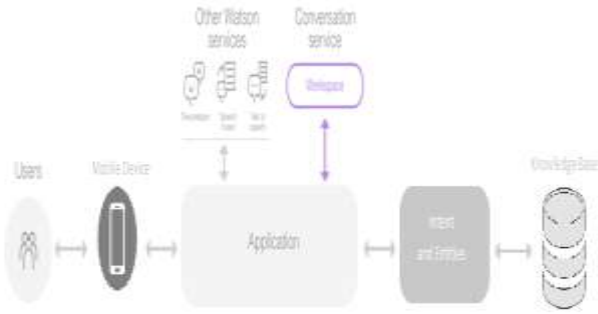

Fig-1: Architecture diagram of proposed system.

3. MODULE DESCRIPTION

3.1 Intent Identification

An intent represents the purpose of a user's input, such as answering a question or processing a bill payment. The developer defines an intent for each type of user request he/she wants the application to support. By recognizing the

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

intent expressed in a user's input, the Watson Assistant service can choose the correct dialog flow for responding to it. The intent is developed by the developer to define a valid question that can be answered by the chatbot assistant. Questions that are not suitable for the service provided by the chatbot can be replied with not answerable replies. Some inputs might contain gibberish which can also be detected and replied with not suitable input messages. A group of example questions are categorized into one intent heading. An intent that is used in the chatbot system is shown below with intent heading and example questions. It must be noted that every intent heading has at least five questions for better performance by the chatbot.

Table-1: Intent heading with questions

Intent heading	Questions
Test Name	CIT
	Centralized Internal Test
	Internal test
	Internal examination
	Examination

3.2 Entity and Dialog Flow

An entity represents a term or object that is relevant to the intents and that provides a specific context for an intent. The developer lists the possible values for each entity and synonyms that users might enter. By recognizing the entities that are mentioned in the user's input, the Watson Assistant service can choose the specific actions to take to fulfil an intent. Entities are common words also known as keywords used by users for the particular service the chatbot provides. These entities have various synonyms for the words to cover as much words as possible to provide maximum relevant output. Based on the intent and entity the reply is selected from the dialog flow. A dialog is a branching conversation flow that defines how the application responds when it recognizes the defined intents and entities. The developer uses the dialog builder to create conversations with users, providing responses based on the intents and entities that the chatbot recognize in the user's input. An example of entity is listed below.

Table-2: Entity and synonyms

Entity	Synonyms
Marks	Score
	Grade
	Rank
	Result

3.3 Application Development

Application development is the part where the user interface for chatbot is developed. To develop an intuitive and self-explanatory user interface android studio is used. All the services and skills from the IBM Watson chatbot is integrated into the java program b API keys and URLs. Each service has a unique API key and URL. These API keys and URLs are added with the java program to build an IBM Watson interface. Mobile applications that are developed from android studio are supported for all android devices. Android studio uses XML to build the user interface. This interface is developed in such a way that user builds a conversation with the chatbot and gathers the required information. A Screenshot of the working chatbot interface is attached in this image.

Fig-2: User Interface

3.4 Knowledge Storage

Knowledge base is the database that is used by the chatbot. A database is an organized collection of data, generally stored and accessed electronically from a mobile. Where databases are more complex, they are often developed using formal design and modelling techniques. The values in the database are fetched for suitable questions and returned as reply for the user's request. The Knowledge base is accessed by SQL (Structured Query Language) Query Language. In this system the knowledge base mainly stores scores from the students.

3.5 STT and TTS API

The two main services used in this chatbot is Speech to Text and Text to speech services. They are implemented along with the Assistant service. Their API keys and URLs are added in the application development module. The Speech to Text is enabled by using the dedicated button which can be enabled by the user to give input by voice. It is captured by the mic in the mobile device and created as input for the chatbot. After fetching the response or reply for the user's request it is displayed as well as given as a voice output by the chatbot. For voice output the Text to Speech API is used.

A clear representation for the working of Text to Speech and Speech to Text service is shown in the diagram.

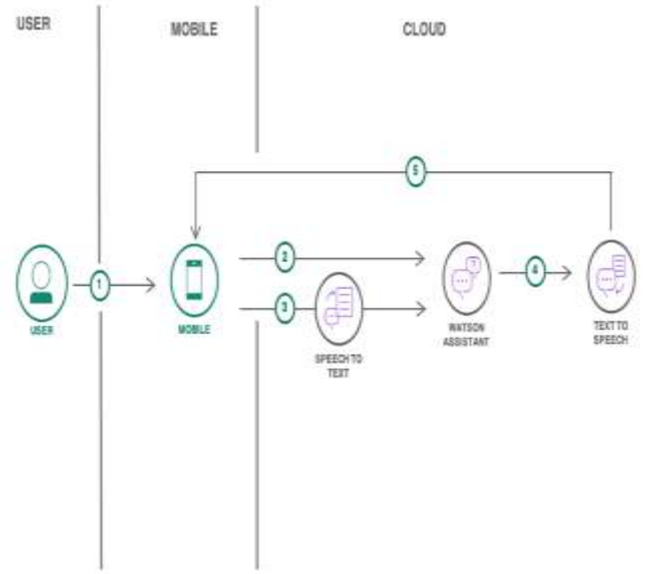

Fig-3: STT and TTS Working

4. FUTURE WORK

Future studies must focus on building admin and user logins in which the admin are mostly professors who can add new students and their marks. The user logins might be given for registered users mostly parents of the students who can raise queries to teachers and also help by suggesting app features for better experience for the users. This chatbot can also be created for IOS devices too, as it has been developed for android devices now. Reputed universities can implement extra features that can make the relationship between professors and parents even stronger.

5. CONCLUSION

The system is developed for Students marks searching chatbot. It consists of 3 phases. The request phase, Analysis phase, response phase. The request phase gets the use input by audio or text. Then comes the analysis phase in which input is checked for relevant intent. It is matched with the dialog flow and the reply is generated. In response phase the generated response is displayed as output through the interface of the chatbot for the user. It is also given as voice output for the user. The output accuracy is 97% based on the small number of chatbot conversations.

REFERENCES

[1] H. Honda and M. Hagiwara, "Question Answering Systems with Deep Learning-Based Symbolic Processing," in IEEE Access, vol. 7, pp. 152368-152378, 2019. doi: 10.1109/ACCESS.2019.2948081.

[2] R. K. Lindsay B. G. Buchanan E. A. Feigenbaum J. Lederberg Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project New York NY USA:McGraw-Hill 1980.

[3] J. R. Quinlan "Induction of decision trees" Mach. Learn. vol. 1 pp. 81-106 1986.

[4] S. Muggleton "Inductive logic programming" New Gener. Comput. vol. 8 no. 4 pp. 295-318 Feb. 1991.

[5] G. Brewka Nonmonotonic Reasoning: Logical Foundations of Commonsense Cambridge U.K.:Cambridge Univ. Press 1991.

[6] D. Bahdanau, K. Cho, and Y. Bengio, ``Neural machine translation by jointly learning to align and translate,'' in Proc. ICLR, San Diego, CA, USA, 2015, pp. 1_15.

[7] J. Doyle "The ins and outs of reason maintenance" Proc. 8th Int. Joint Conf. Artif. Intell. (IJCAI) pp. 349-351 1983.

[8] R. Higashinaka and H. Isozaki, ``Corpus-based question answering for why-questions,'' in Proc. IJCNLP, Hyderabad, India, 2008, pp. 418_425.

[9] A. C. Kakas R. A. Kowalski F. Toni "Abductive logic programming" J. Log. Comput. vol. 2 no. 6 pp. 719-770 Dec. 1993.

[10] G. E. Hinton "Preface to the special issue on connectionist symbol processing" Artif. Intell. vol. 46 no. 1 pp. 1-4 Nov. 1990.

[11] D. S. Touretzky "BoltzCONS: Dynamic symbol structures in a connectionist network" Artif. Intell. vol. 46 no. 1 pp. 5-46 Nov. 1990.

[12] M. Schlichtkrull T. N. Kipf P. Bloem R. van den Berg I. Titov M. Welling "Modeling relational data with graph convolutional networks" Proc. Eur. Semantic Web Conf. (ESWC) pp. 593-607 2018.

[13] T. Trouillon J. Welbl S. Riedel E. Gaussier G. Bouchard "Complex embeddings for simple link prediction" Proc. 33nd Int. Conf. Mach. Learn. (ICML) pp. 2071-2080 2016 2016.

[14] T. Rocktaschel S. Riedel "End-to-end differentiable proving" Proc. Annu. Conf. Neural Inf. Process. Syst. pp. 3788-3800 2017.

[15] P. Minervini M. Bosnjak T. Rocktäschel S. Riedel "Towards neural theorem proving at scale" arXiv:1807.08204 2018.

[16] H. Dong J. Mao T. Lin C. Wang L. Li D. Zhou "Neural logic machines" Proc. Int. Conf. Learn. Represent. pp. 1-22 2019.

[17] S. Tellex B. Katz J. Lin G. Marton A. Fernandes "Quantitative evaluation of passage retrieval algorithms for question answering" Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. pp. 41-47 Aug. 2003.

[18] R. Sequiera G. Baruah Z. Tu S. Mohammed J. Rao H. Zhang J. Lin "Exploring the effectiveness of convolutional neural networks for answer selection in end-to-end question answering" arXiv:1707.07804 2017.

[19] N. T. Thomas "An e-business chatbot using AIML and LSA" Proc. Int. Conf. Adv. Comput. Commun. Inform. (ICACCI) pp. 2740-2742 Sep. 2016.

[20] L. Cui F. Wei S. Huang C. Tan C. Duan M. Zhou "SuperAgent: A customer service chatbot for e-commerce websites" Proc. 55th Annu. Meeting Assoc.Comput. Linguistics-Syst. Demonstrations pp. 97-102 Jul. 2017.

[21] H. Bhargava D. Power "Decision support systems and Web technologies: A status report" Proc. Amer. Conf. Inf. Syst. pp. 46 Dec. 2001.

[22] M. S. Kohn J. Sun S. Knoop A. Shabo B. Carmeli D. Sow T. Syed-Mahmood W. Rapp "IBM's health analytics and clinical decision support" Yearbook Med. Inf. vol. 9 no. 1 pp. 154-162 Aug. 2014.