International Research Journal of Computer Science (IRJCS)
Issue 03, Volume 5 (March 2018)
ISSN: 2393-9842
www.irjcs.com

# COUNSELLOR CHATBOT

**Rishabh Chawla**
Computer Science and Engineering, Vellore Institute of Technology, Vellore
rishabhchawla1995@gmail.com
**Anuradha J**
Associate Professor, Vellore Institute of Technology, Vellore
j.anuradha@vit.ac.in

**Manuscript History**
Number: **IRJCS/RS/Vol.05/Issue03/MRCS10087**
https://doi:/10.26562/IRJCS.2018.MRCS10087
Received: 12, March 2018
Final Correction: 23, March 2018
Final Accepted: 28, March 2018
Published: March 2018
**Citation:** Rishabh & Anuradha (2018). COUNSELLOR CHATBOT. IRJCS:: International Research Journal of Computer Science, Volume V, 126-136. doi://10.26562/IRJCS.2018.MRCS10087
Editor: Dr.A.Arul L.S, Chief Editor, IRJCS, AM Publications, India
Copyright: **©2018 This is an open access article distributed under the terms of the Creative Commons Attribution License, Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited**

**Abstract**—The aim of this research is to build a counsellor chat bot as a service which responds giving the person advice based on the premise given to it. Xen learns and understands what the user is trying to convey based on conversations it has had with the users. Generative chatbots train on data and work based on content classification or by making rules understanding the data and thus may be able to work for all scenarios. Lot of information needs to be analyzed to understand which is, what and what all it can mean in context classification. The context can have a lot of meanings along with vagueness, incomplete and incorrect data. It can be used by people to chat with or understand their problem or just to talk about when they need a flow of thought. The proposed chatbots are Retrieval, AIML (Artificial Intelligence Markup Language) where it uses self-learning using K-Means, after it has collected enough data by interacting with the user and Generative chat-bot. The generative chatbot trains itself using context classification so that it can improve its responses and expand the responses it uses. The proposed chatbot Xen gives appropriate responses based on the context it is given and responds appropriately and as well understands the sentiment that is being portrayed. It trains on the conversations it has had and gives more appropriate responses the next time in a similar context and as well is able to respond to more situations.

**Keywords**—Artificial Intelligence Mark Up Language (AIML); Long Short Term Memory (LSTM); Gated Recurrent Unit (GRU);

## I. INTRODUCTION

"We don't just use technology; we live with it "(MaCarthy& Wright, 2004, p. ix). The depersonalized drill and practice interface leaves the user bored and unable to complete a simple learning task. Is it possible to design language learning interfaces that are more usable? Long gone are the days when language learning interfaces are seen as merely powerful cognitive tools or media, which are devoid of users' motivation to use and hinder them to reach their desired learning goals. Much more deeply than ever before we are aware that end-users influence how a design takes shape and users have a deep impact on the design. Interacting with technology involves users emotionally, intellectually and sensually" (MaCarthy& Wright, 2004). Users experience technology in accordance with their emotions and intellect. They become a central part of the development process and are placed at the center of the design. As a result, the role of the designer is to facilitate the task for the user and to make sure that the user is able to make use of the product as intended and with a minimum effort to learn how to use it (Abras, Maloney, &Preece, 2004). Hence, people who design interactive systems need to be able to understand and analyze users' experience with technology (Norman, 2006). Therefore, this research contributed to the study of the need to fully explore users' experiences with language education interface during design processes.[1]


© 2014- 18, IRJCS- All Rights Reserved

The past decade has witnessed a great deal of interest in technology driven language learning. Various technologies have been employed in many settings and environments to provide assistance to language learners. Because of advances in pedagogy and learning paradigms, the traditional behaviorist language learning model has shifted to a communicative and integrative model, and thus, in accordance with this shift, the traditional unimodal language learning interface has moved towards a multimodal communicative and adaptive interface in CALL (computer assisted language learning) systems.[1] A chatbot is a conversational agent that interacts with users turn by turn using natural language. [2]

Chatbot technology integrates a language model and computational algorithms to emulate informal chat communication between a human user and a computer using natural language.[3] "Before there were computers, we could distinguish persons from non-persons on the basis of an ability to participate in conversations. But now, we have hybrids operating between person and non-persons with whom we can talk in ordinary language." (Colby 1999a). Human machine conversation as a technology integrates different areas where the core is the language, and the computational methodologies facilitate communication between users and computers using natural language. [4] Chatbots were brought about as the conversation with chatbot's is done in a similar manner as we converse with another person and that is the goal of a chatbot to be an assistant which is very similar to that of a human and thus, the user can use it as naturally as any person is asked for anything rather than any specific format which is used in applications. This research will build a generative chat bot that is text-based which is not just limited to the data that it is trained on, but also learn based on all the data that it interacts with and as well as generate responses based on what it learns. The complexities lie in developing a chat bot that will provide valuable replies by understanding the conversation and asking the relevant questions.

## II. LITERATURE SURVEY

Chatbot is an automated system which responds to preset conditions based on the interaction and respond to the question provided. A counsellor helps to understand that if any mental health issue is present or troubling the person, help him understand more about it and improve the mental health of the person in a way if it can. This research focuses on three types of chat bots namely

1. Retrieval Based: A predefined set of queries and responses are fed into the system. These models are incapable of generating any new text or content. In essence, they select a definite text from a predefined set of texts and use the set that best serves the intent of the question. These models serve well for chat bots that perform a specific task which are domain specific.
2. Rule Based: AIML is the popular way of implementing rule based chatbots. AIML, or Artificial Intelligence Mark-up Language, is a derivative of Extensible Mark-up Language (XML). It was developed by Wallace and the Alicebot free software community from 1995 onwards to enable people to input dialogue pattern knowledge into chatbots based on the A.L.I.C.E. open-source software technology. AIML consists of data objects called AIML objects, which are made up of units called topics and categories. The topic is an optional top-level element that has a name attribute and a set of categories related to that topic. Categories are the basic unit of knowledge in AIML. Each category is a rule for matching an input and converting to an output, and consists of a pattern, which matches against the user input, and a template, that is used in generating the ALICE chatbot answer.[5]
3. Generative Based: These models have no defined classes and self-generate entire answers of their own based on their understanding of a giving sentence. They generate entirely new responses from scratch.

## DIFFERENT CHATBOTS

1. Eliza - Eliza simulated conversation by using a pattern matching and substitution methodology that reposed to the user based on matching the scenario to its pre-trained scenarios. Eliza gave users an illusion of understanding on the part of the program, but had no built in framework for contextualizing events.The idea was simple and based on keyword matching. The input is inspected for the presence of a keyword. If such a word is found, the sentence is mapped according to a rule associated with the keyword; if not, a connected free remark, or under certain conditions an earlier transformation, isretrieved.[6]
2. Brisbot - Brisbot is a chat bot for giving advice directly. Brisbot is loaded with hundreds of questions from kids, with answers written by counselors. Brisbot does not respond to open questions, instead the user will navigate through pre-programmed professional advice.[7]
3. Mitsuku - Mitsuku is a chatbot created using AIML. It learns on its own and thus can respond to things it hasn't learned on but is limited to the rules and features it has been created with. Mitsuku won the 2013 and 2016 Loebner prize. Mitsuku updates itself but is created with a generic objective of having a conversation and is limited to the rules and a feature it has been created with using AIML.Mitsuku is quick-witted, occasionally confusing, and strangely engaging. She is also a chatbot, built from the A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) platform originally developed by Richard Wallace in 1995. She conducts thousands of conversations daily, according to Lauren Kunze, principal of Pandorabots, the Oakland, CA based company behind the technology. "She doesn't really do anything," Kunze says.

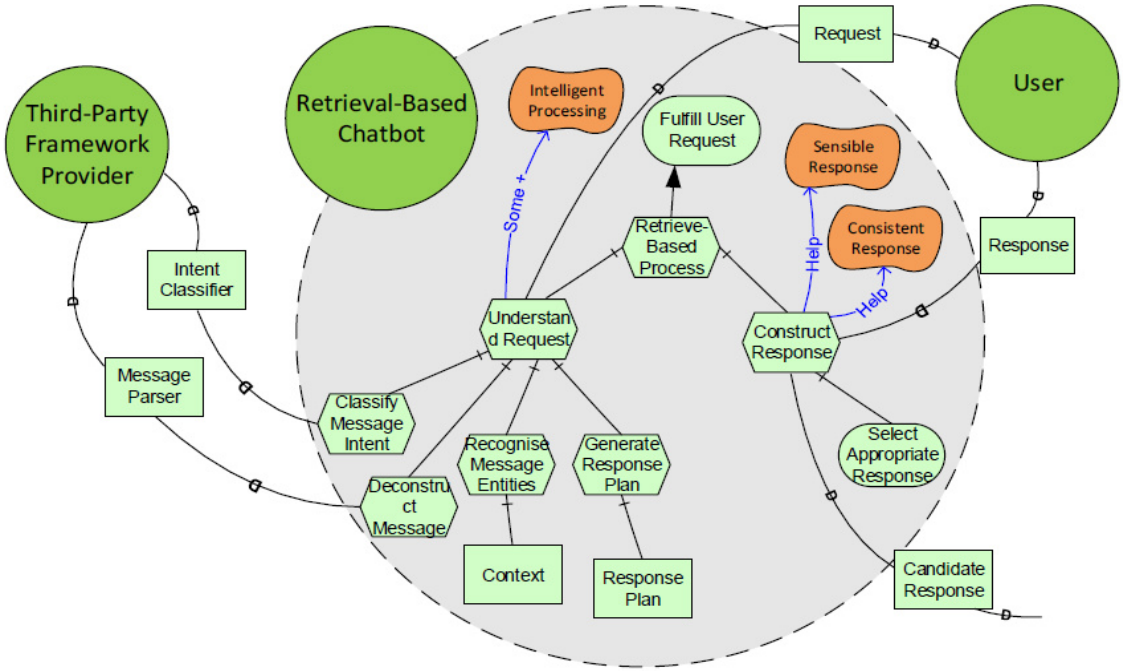

© 2014- 18, IRJCS- All Rights Reserved 

"She's not designed to assist you. She can tell you the weather or perform an Internet search, but she's really just there to talk to you, and she's wildly popular with teens. People say, 'I love you' and 'you're my best friend.'"[8]

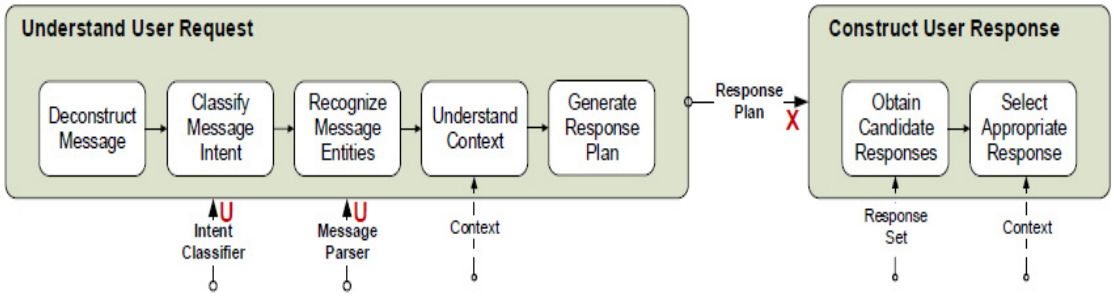
4. Joy - Joy is a chat bot which takes care of your mental health. Joy sends daily check-ins, asking how you feel and what you did that day. Joy then uses the response to interpret the mood and respond appropriately. Joy uses a generic corpus provided by IBM Watson and Microsoft LUIS to understand the intent and give responses.[9] What is an Emotion? It may be feelings, or adaptive dispositions, or cognitive processes, or behavior, or social performance.

In fact, most of these descriptions of emotion are partial. Research on emotions is as vast and diverse as emotional life itself. This diversity of theoretical perspectives on emotions suggest that emotions are undefined (Lazarus, 1984) and yet this diversity is necessary. Each of these different theoretical approaches provide its own answer to the ontological question of emotion, together define the phenomena, suggest specific mechanisms underlying them, and frame questions that future emotion research should address (Leventhal &Tomarken, 1986). Each tradition for defining emotion helps us understand the ground rules for theorizing emotion.[1] One of the problems with studying emotions in technology is what it is. No topic of our mental life seems more complicated, fascinating and important to us than emotions. Since William James (1884) asked the question "What is an emotion", the ontology of emotion continues to baffle us. Researchers and theorists in different disciplines have battled for the soul of emotions for thousands of years. Little agreement and research trends toward a definition of emotion shape affective HCI towards initiating many state of-the -art research projects based on different emotion traditions and inventing innovative technologies to address emotions.[1]

## III. REFERENCE ARCHITECTURE

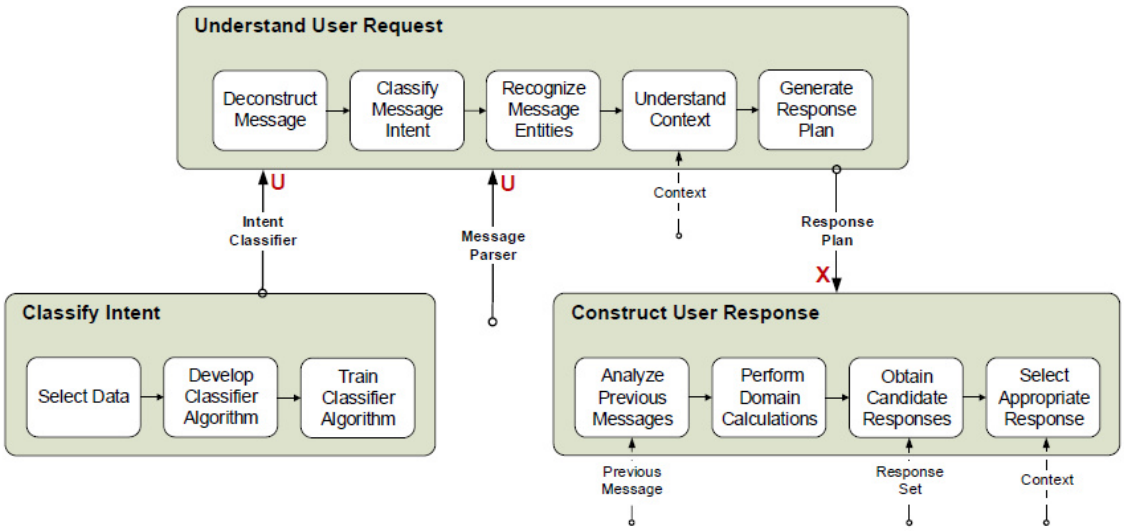



i* Strategic Rationale Model for Retrieval-Based Chatbots[10]

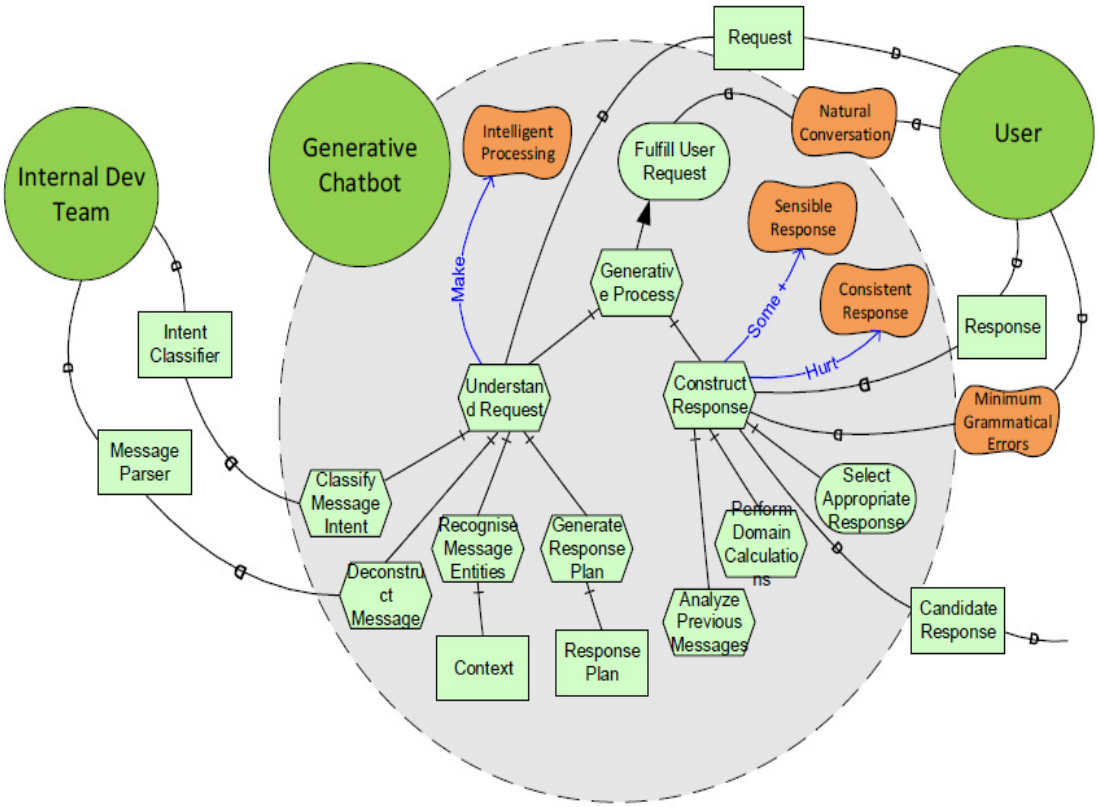



Process Architecture Model for Retrieval-Based Chatbots[10]


© 2014- 18, IRJCS- All Rights Reserved



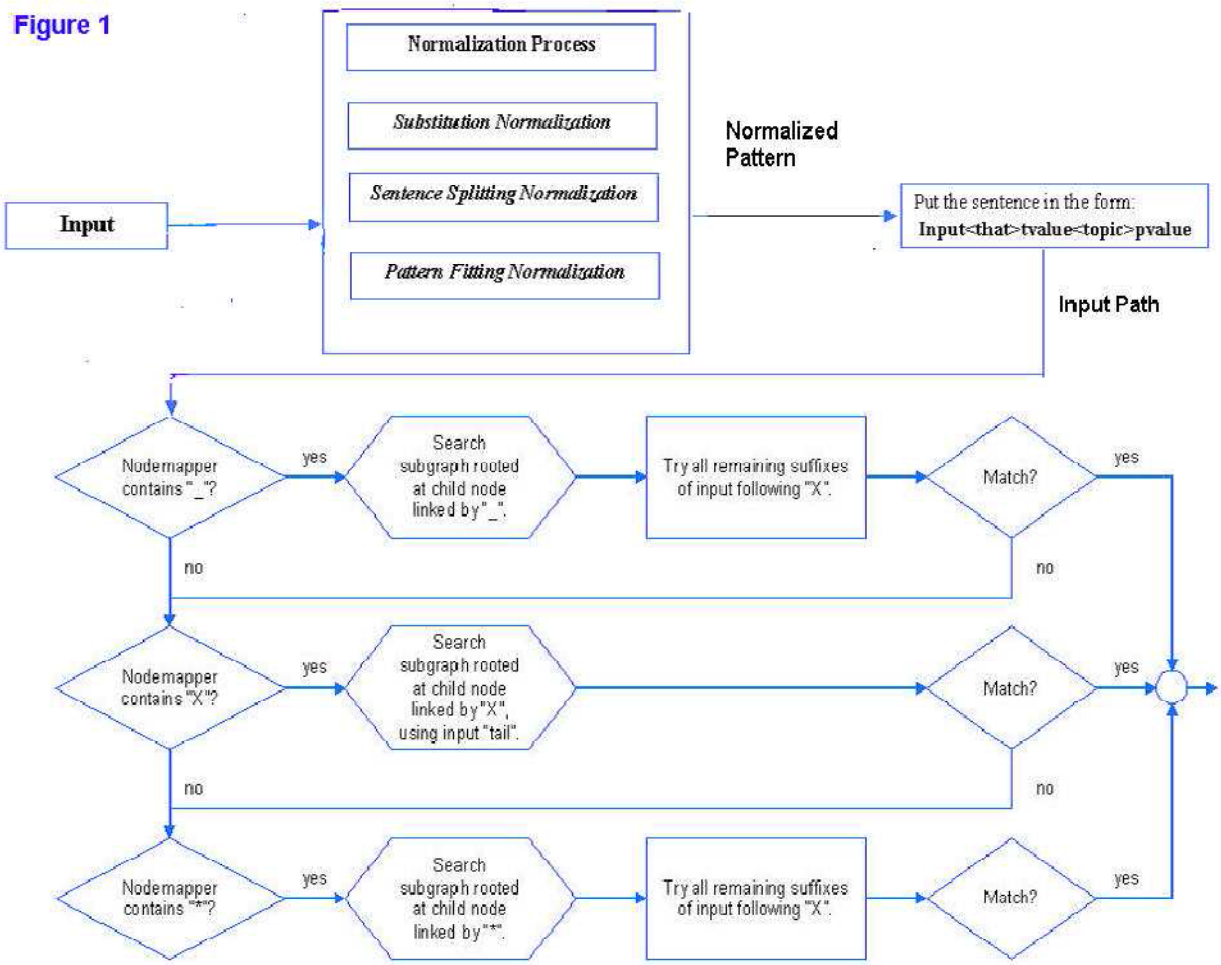

Process Architecture Model for Generative Chatbots[10]




i* Strategic Rationale Model for Generative Chatbots[10]

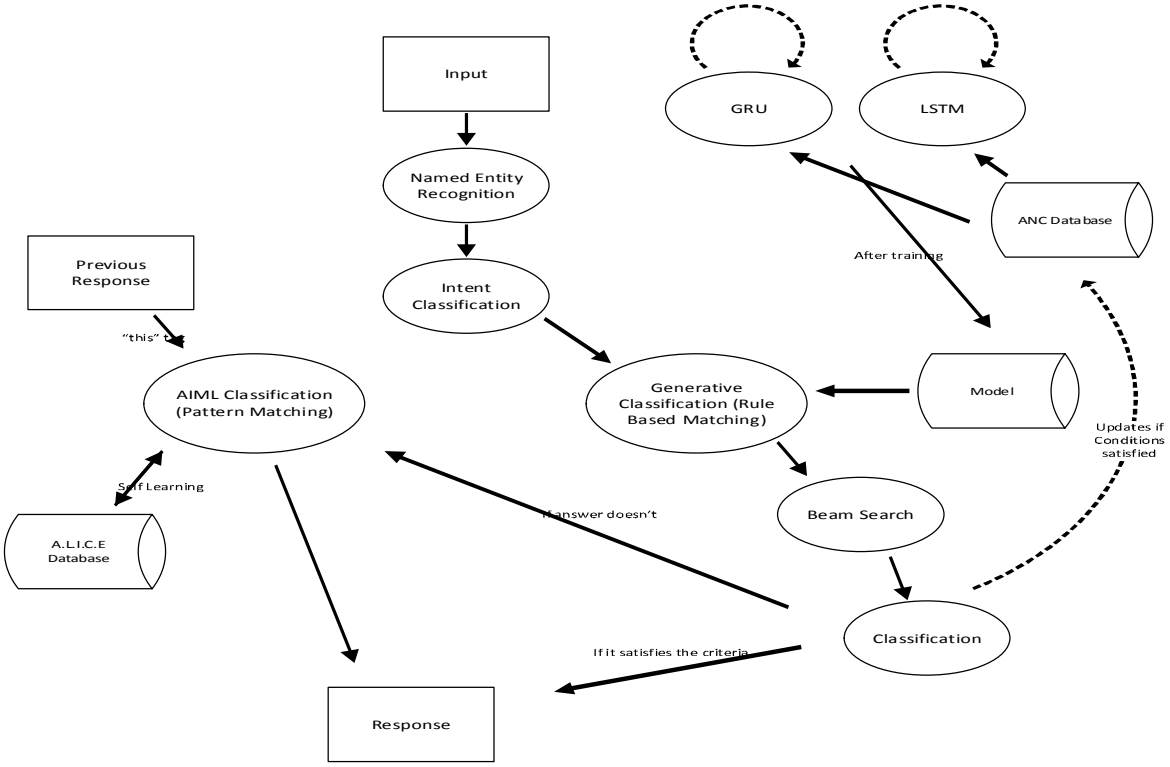

© 2014- 18, IRJCS- All Rights Reserved

Figure 1




Rule Based AIML Architecture[4]

## IV. ARCHITECTURE





© 2014- 18, IRJCS- All Rights Reserved

## V. METHOLODOGY

Data Acquisition: Three different data sets were used to implement three different types of chat bots.

1. Retrieval Pattern Matching: For this chat bot, data was acquired from ELIZA .
2. Retrieval Rule Based AIML: For this chat bot, data was acquired from A.L.I.C.E.(Artificial Linguistic Internet Computer Entity). It is a natural language processing program which engages in a conversation with a human by applying some heuristics pattern matching rules to the input given.
3. Generative: For this chat bot, data was acquired from ANC.org[11][The Open American National Corpus] which is a massive electronic collection of American English, including texts of all genres and transcripts of spoken data.

Pre Processing: To ensure a clean data for the chat bot to work, prepossessing is done.

1. Tokenization: It is defined as, if a character sequence and a defined document unit is provided, tokenization will chop it up into pieces, called tokens while removing certain unwanted characters such as punctuation's.[12] A token is "a formal item (at least one morpheme long) whose pattern of occurrence can be described in terms of a uniquely ordered series of other lexical items occurring in its environment". By comparing observed collocation patterns of strings with stored patterns we can proceed to segment the text into words. Finding significant tokens depends on the ability to recognize patterns displaying significant collocation.[13] The comparison we make of different tokenization policies and procedures shows clearly the need for tokenization evaluation.[14] The machine learning model is intended to provide the optimal tokenization for each word in the testing set, having the model trained on the data generated from the lattice decoding. We approach this problem by learning the optimal tokenization scheme tag for each word, rather than the actual lexical tokenization, from the lattice results. We then apply this model to the testing words. The resulting tags are then used to get the corresponding actual tokenization through a lookup table. The input to the lookup table is the scheme tag and surface form, while the output is the corresponding actual tokenization. A learning approach to model the optimal tokenization options based on the lattice decoding, to facilitate a more practical tokenization process.[15] Tokenization improves the learning rate and quality as the initial analysis is done based on the values given to the words using matrices or vectors based on many algorithms but the initial values given to the words can influence the machine's learning rate and inferences to a great degree.A good tokenization mechanism can be taking a corpus of the specific domain or the important words and give them higher values and then normalize the rest of the dataset with a generic corpus. Optimization techniques can be used as described.
   1. Stemming: It is the process of semantic normalization; where in the variant forms of a word are reduced to a common form.
   2. NER - Named Entity Recognizer. It categories the word into the specified categories based on what it represents. It is useful as when words are classified into categories and then relation between them becomes easier to find and classification becomes more accurate and useful.
   3. Retrieval Pattern Matching: The chat bot uses phrase matching algorithm to compare the user entered data to ELIZA's data. Phrase matching allows to searching for exact keyword and close variants of the exact keyword, with additional words before or after. This is why stemming is needed. Phrase match is more targeted than the default broad match, but more flexible than exact match.

The following approach is followed to build the proposed chatbotXen, the chat bot for retrieval pattern matching:

i. Get the input from the user
ii. Phrase Matching on the database is done with the help of K-Means
iii. Check if any reflections are present and accordingly replace them
iv. Update the weights in the database
v. If multiple responses are present select one at random

5. Retrieval Rule Based AIML: In AIML where each category is a rule for matching an input and converting to an output, and consists of a pattern, which matches against the user input, and a template, which is used in generating the ALICE chatbot answer.[2]. Self Learning is used for of modelling appropriate responses with respect to the person. The program learns from interactions and thus when different people interact with the bot, it responds in accordance with the learning that took place with that specific user.

The following approach is followed to build the proposed chatbotXen, the chat bot or Retrieval Rule Based AIML:

i. Get the input from the user
ii. Check the AIML database for an appropriate response
   a. Input is preprocessed for specific words to be substituted with other words.
   b. Pattern matching is done on the input and database while an AIML parser is used on the database to infer the rules.


© 2014- 18, IRJCS- All Rights Reserved 

c. AIML parser checks the previous input if "then" tag is encountered and pattern matching is applied on that.
d. Word Substitution takes place based on the AIML rules in the responses.

iii. Randomly select a response if multiple responses are present
iv. Update the weights and add words in the database for self-learning

6. Generative: Context Classification tries to classify what the user is trying to convey by assigning each input to one of the intents or understanding the context as what it is trying to accomplish.

The following approach is followed to build the proposed chatbotXen, the chat bot for Generative:

i. Train the model based on a dataset
ii. The dataset can be trained using LSTM or GRU
iii. Context Learning is applied by what words can mean based on the way they are used and around which words they appear
iv. The input is taken from the user
v. NER is used to generalize the input making it more meaningful.
vi. Response is generated based on what is understood from it
vii. Beam Search is done to select the best response

## VI. IMPLEMENTATION

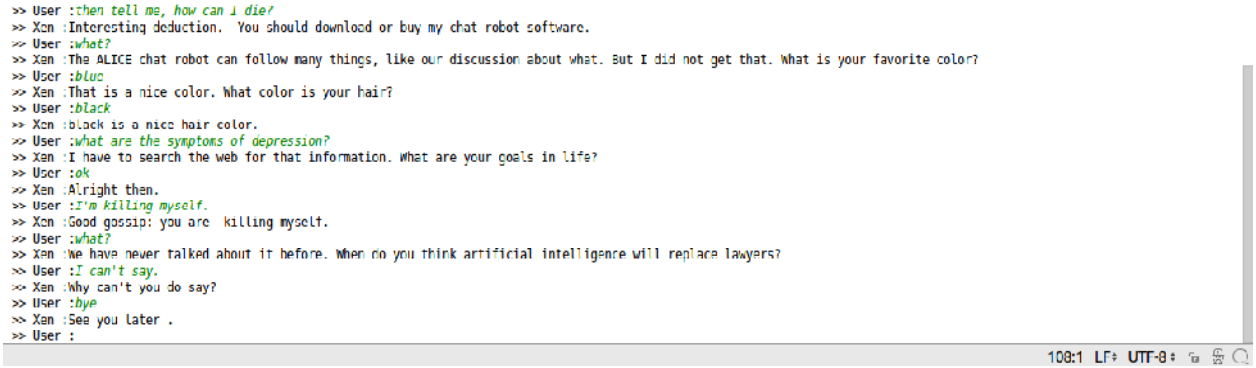
1. Retrieval Based: Phrases are present in the database in multiple order where a part of the sentence can be related to a lot of sentences and accordingly a response is chosen from the database. The database has reflections where if certain words are present they are replaced with select words when the response is sent back to the user. This uses the n-gram technique so as to say that the words around a word reflect the response not just the words present in the sentence itself thus, it provides more relevant responses. CHARAGRAM embeddings, a simple approach for learning character-based compositional models to embed textual sequences. A word or sentence is represented using a character n-gram count vector, followed by a single nonlinear transformation to yield a low-dimensional embedding. We use three tasks for evaluation: word similarity, sentence similarity, and part-of-speech tagging.[16] Stata command n-gram which implements the most common approach to text mining, "bag of words". An n-gram is a contiguous sequence of words in a text. Broadly speaking, n-gram creates hundreds or thousands of variables each recording how often the corresponding n-gram occurs in a given text.[17]
2. AIML Based: The AIML ALICE libraries are used. The inputted words are given weights based on the words around them which is done using word vectorize by sklearn and the words are passed along with the sentence to the AIML library where it is analyzed with the database and is checked with the rules and the respective response is sent. If it satisfies multiple responses a response is chosen at random. AIML database works based on rules which are defined and prioritized thus, the best match is chosen. There are certain patterns searched if any are matched and accordingly there is one or more responses present which are chosen and sent to the user. Words are replaced with synonyms for when word matching takes place. The whole or a part of the sentence is seen and matched and accordingly if there are no patterns the word matching takes place based on the weights given to the words. The inputs given by the user are stored in a database along with word associations with the words around it which are used for self learning. Certain categories are automatically learned by the AIML bot where the "think" tag is used.
3. Generative: The dataset is taken and every sentence is first mapped based on NER(Named Entity Recognition) and then weights are accordingly provided to each word as to which words type of words they are present around and which sentences they appear in and thus, their context is understood by the chatbot and accordingly so when an input is given by the user the context of the sentence is understood by NER and then the generative chatbot responds based on what relevance is shown towards that sentence. The dataset is trained either with LSTM or GRU where each character relevance is taken and thus the words are understood based on which characters appear where in a word and which word appears near which word and thus the context as to what comes after which kind of sentence is understood by the chatbot and stored in the database. As most convolution models (Lecun and Bengio 1995; Kalchbrenner, Grefenstette, and Blunsom 2014), we consider the convolution unit with a local "receptive field" and shared weights to capture the rich structures and composition properties between consecutive words. The sentence CNN for generating the question representation. For a given question with each word represented as the word embedding (Mikolov et al. 2013), the sentence CNN with several layers of convolution and max-pooling is performed to generate the question representation vqt.[18] RNN is Recurrent Neural Network where there is feedback followed on what is learned before and that affects the new weights which are given after that. It is used to predict the word which comes after a word. Recurrent Neural Networks (RNNs) are powerful models for sequential data that have the potential to learn long-term dependencies. Recurrent Neural Networks (RNNs) extend Neural Networks to sequential data. [19].


© 2014- 18, IRJCS- All Rights Reserved 

LSTM is Long Short-Term Memory which works similar to RNN but has hidden weights and uses a non-linear function. It only keeps track of the previous state and so, the memory doesn't run out.It addresses the vanishing gradient problem commonly found in vanilla RNNs by incorporating gating functions into its state dynamics [20] At each time step, an LSTM maintains a hidden vector h and a cell vector c responsible for controlling state updates and outputs.[21] The so-called Long Short-Term Memory (LSTM) type RNN has been particularly successful. Often, it seems beneficial to train deep architectures in which multiple RNNs are stacked on top of each other [22]. Unfortunately, the training cost for large datasets and deep architectures of stacked RNNs can be prohibitively high, often times an order of magnitude greater than simpler models like n-grams [23]. To deal with the deficiency of linguistic information in ImageQA problem, we transfer the information acquired from a large language corpus by fine-tuning the pre-trained embedding network. We initialize the GRU with the skip-thought vector model trained on a book-collection corpus containing more than 74M sentences [24]. Note that the GRU of the skip-thought vector model is trained in an unsupervised manner by predicting the surrounding sentences from the embedded sentences. As this task requires to understand context, the pre-trained model produces a generic sentence embedding, which is difficult to be trained with a limited number of training examples. By fine-tuning our GRU initialized with a generic sentence embedding model for ImageQA, we obtain the representations for questions that are generalized better.[25].GRU is Gated Recurrent Unit.GRU make recurrent blocks adaptively capture the dependencies of different time scales.[26].The main difference between GRU and long short term memory (LSTM) is that the memory cell in LSTM does not exist in GRU. There are three gates and one memory cell in a LSTM block. Input gate, forget gate and output gate separately control the data flow from input, memory and output. The activations of input gate, forget gate and output gate depend on current input, previous memory and previous or current output. While in GRU, the number of gates is reduced. The activations of gates in GRU only depends on current input and previous output. Due to the reduction in parameters, models using GRU tends to converge faster, and the final solution tends to be better than models using LSTM in some tasks [27]. Beam search is used to select the response as it uses graph search and moves towards the best possible node and then reverts back if required but only remembers to only a certain limit. Beam search is a breadth-first style heuristic search and progresses along with the depth step by step.[28] Beam search is like breadth-first search since it progresses level by level without backtracking. But unlike breadth-first search, beam search only moves downward from the best b promising nodes (instead of all nodes) at each level and b is called the beam width. The other nodes are simply ignored. In order to select the best b nodes, promise of each node is determined. This value can be determined in various ways. One way is to employ an evaluation function which estimates the minimum total costs of the best solution that can be obtained from the partial schedule represented by the node.[29] We determine the promising nodes (beam nodes) by performing local and global evaluations and proceed with the search through these selected nodes. After determining the first beam nodes at level 1, we apply the algorithm to these nodes independently and generate one partial tree from each of them. We refer to these partial trees as beams. For each beam after filtering based on the outcome of the global evaluation, one node (beam node for the next level) is selected among the descendants of each node. Since we have beam width number of nodes in the former level while keeping one descendant, we again have beam width number of nodes in the next level and therefore the search progresses through b parallel beams. In the application of beam search, one can have as many beams as possible. On the other hand, filter width is defined for each beam independently. Thus, the number of beams (which is defined by the beam width parameter) can be feasibly greater than the filter width value.[29]

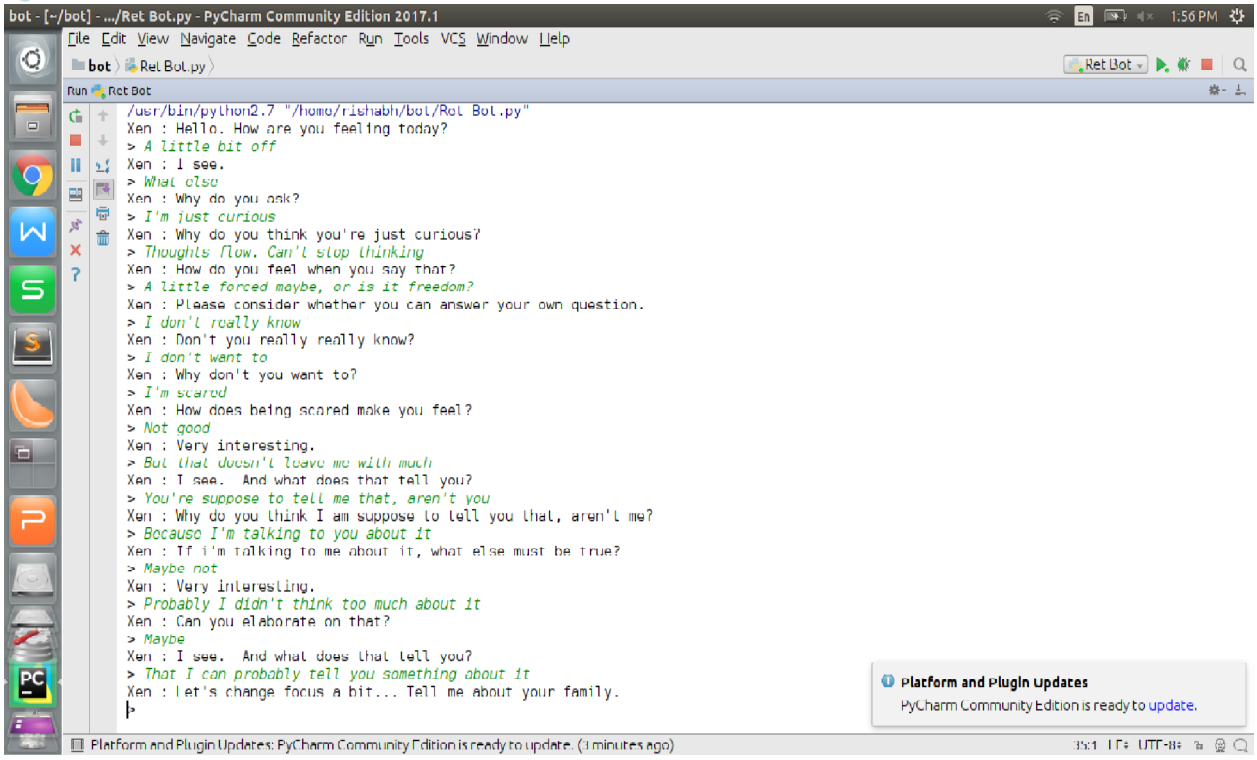
## VII. RESULTS & INFERENCES

As seen in Figures below the chatbots can continue the conversation and after learning on its own for a while it will be able to generate its own answers and interpret what the user is trying to say more accurately. The bot responds appropriately to get to know more about the user and after being trained on more data it will be able to give an analysis of the mental health of the user as well.

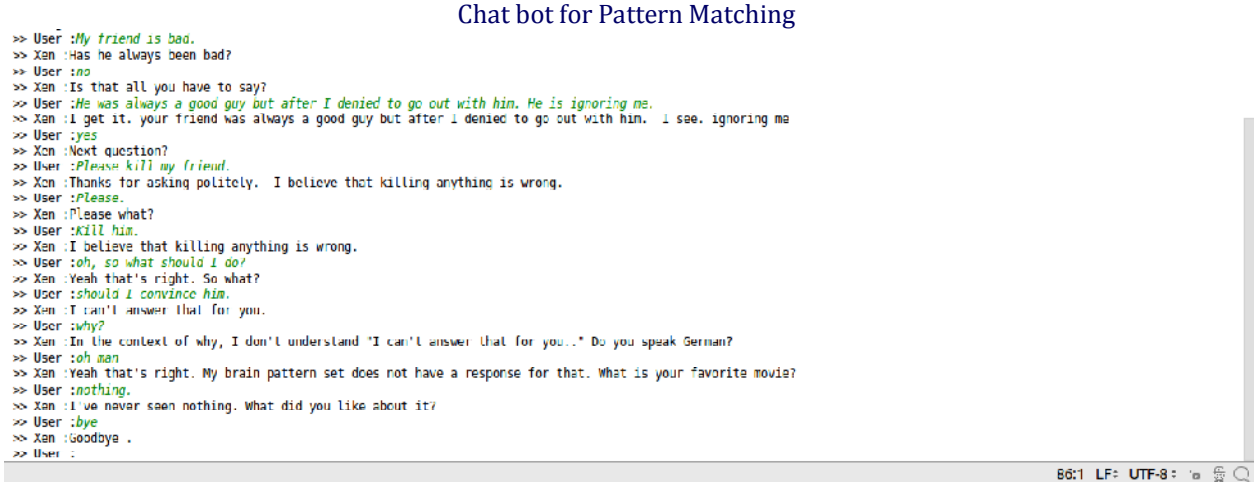



Chat bot for AIML

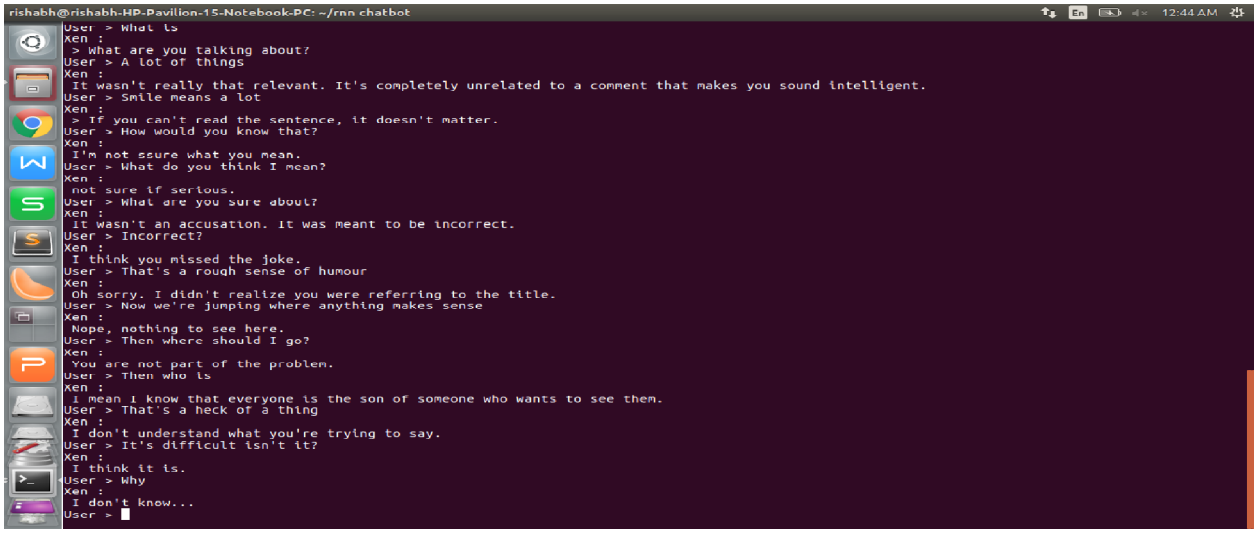

© 2014- 18, IRJCS- All Rights Reserved




Chat bot for Pattern Matching

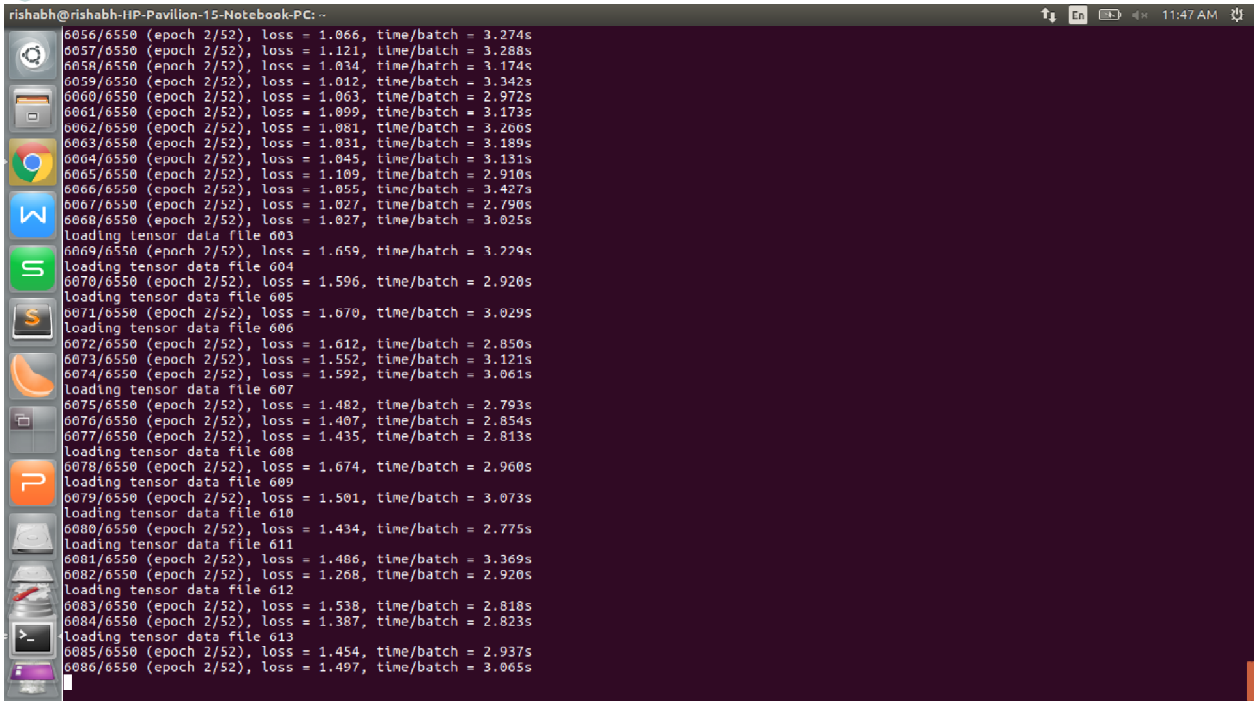



Chat bot for AIML




Chat bot for Generative (GRU)


© 2014- 18, IRJCS- All Rights Reserved




Generative Chatbot Anc Dataset

## VIII. CHALLENGES

1. In the retrieval based chatbot reflections weren't valid at all times thus, word matching wouldn't work thus, regular expressions were used to create patterns in which reflections weren't valid.
2. In the rule based chatbot where the previous input is stored in the "this" tag, the importance of the present input compared to the previous input was decided, if the present input was in relation with guidelines set by the polarity of the context.
3. In rule based chatbot where the present input is a question in relation to the previous context that is the previous output, the previous output is taken into consideration to respond in relation to the context.
4. In the rule based chatbot creating the AIML dataset which updates itself but doesn't conflict with other rules, before any rule was updated the dataset had to be checked for conflicts.
5. In the generative chatbot when more and more data is provided the context starts overlapping between responses and the segregation of these responses was done by limiting the data on which the chatbot was trained and certain conditions like a threshold confidence score was set and only if satisfied the data was used for training else it was discarded.
6. In the generative chatbot the amount of training which gives the best output has to be checked again and again as over fitting leads to generic responses to a lot of categories.
7. In the generative chatbot when retrieval based responses are used in situations when no rules are satisfied the context of the conversation is reset and can't be used again by using the previous input and thus, an object containing the context based on certain criteria's can be used.
8. The generative chatbot has to be trained all over again if it needs to be updated.

## IX. CONCLUSION & FUTURE WORK

The chat bot understands the interests of the end user and tries to keep the user interested in the ongoing conversation. The bot responds appropriately and isolates the problem to a certain extent. Also, the overall mood of a person is improved as they are able to express what they feel.

Future Work:

1. The bot can be trained with the help of the data it has gathered and reduce the percentage of retrieval responses it sends.
2. The bot can use grammatically correct English while using generative responses.
3. The bot can learn the mood of a person by analyzing the habits done by the person on the device.
4. The bot could take input via audio or microphone rather than just text and respond in the similar manner.
5. The bot could read the mood of a person via a camera or a picture.
6. The bot could give a graph to the person about their mood at periods of time and give statistics which when mapped with activities can describe what a person feels towards an activity.


© 2014- 18, IRJCS- All Rights Reserved

## REFERENCES


1. Wang, Yifei. Designing chatbot interfaces for language learning: ethnographic research into affect and users' experiences. Diss. University of British Columbia, 2008.
2. Shawar, Bayan Abu, and Eric Atwell. "Machine Learning from dialogue corpora to generate chatbots." Expert Update journal 6.3 (2003): 25-29.
3. Shawar, Bayan Abu, and Eric Steven Atwell. "Using corpora in machine-learning chatbot systems." International journal of corpus linguistics 10.4 (2005): 489-516.
4. Shawar, Bayan Abu, and Eric Atwell. A comparison between ALICE and Elizabeth chatbot systems. University of Leeds, School of Computing research report 2002.19, 2002.
5. Shawar, Bayan Abu, and Eric Atwell. "Chatbots: are they really useful?." LDV Forum. Vol. 22. No. 1. 2007.
6. Shawar, Bayan Abu, and Eric Atwell. A comparison between ALICE and Elizabeth chatbot systems. University of Leeds, School of Computing research report 2002.19, 2002.
7. Jaya Saxena, "Can troubled kids find help through a chat bot?", https://www.dailydot.com/irl/brisbot-kids-chat-bot/
8. Mone, Gregory. "The edge of the uncanny." Communications of the ACM 59.9 (2016): 17-19.
9. Jordan Davidson, "Facebook Messenger Chatbot, Joy, Wants to Help People Improve Their Mental Health", https://themighty.com/2016/07/facebook-messenger-chatbot-joy-wants-to-help-improve-mental-health/
10. Babar, Zia, Alexei Lapouchnian, and Eric Yu. "Chatbot Design-Reasoning about design options using i* and process architecture."
11. https://www.anc.org
12. J. Pei, "Tokenization", Information Retrieval and Web Search, http://www.cs.sfu.ca/CourseCentral/456/jpei/web%20slides/L07%20-%20Tokenization.pdf
13. Webster, Jonathan J., and Chunyu Kit. "Tokenization as the initial phase in NLP." Proceedings of the 14th conference on Computational linguistics-Volume 4. Association for Computational Linguistics, 1992.
14. Habert, Benoit, et al. "Towards tokenization evaluation." Proceedings of LREC. Vol. 98. 1998.
15. Zalmout, Nasser, and Nizar Habash. "Optimizing Tokenization Choice for Machine Translation across Multiple Target Languages." The Prague Bulletin of Mathematical Linguistics 108.1 (2017): 257-269.
16. Wieting, John, et al. "Charagram: Embedding words and sentences via character n-grams." arXiv preprint arXiv:1607.02789 (2016).
17. Schonlau, Matthias, and Nick Guenther. "Text mining using N-Grams." (2016).
18. Ma, Lin, Zhengdong Lu, and Hang Li. "Learning to Answer Questions from Image Using Convolutional Neural Network." AAAI. Vol. 3. No. 7. 2016.
19. Laurent, César, et al. "Batch normalized recurrent neural networks." Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016.
20. Sepp Hochreiter and J¨urgenSchmidhuber, "Long short-term memory," Neural computation,vol. 9, no. 8, pp. 1735–1780, 1997.
21. Felix A Gers, Nicol N Schraudolph, and J¨urgenSchmidhuber, "Learning precise timing withlstm recurrent networks," The Journal of Machine Learning Research, vol. 3, pp. 115–143, 2003.
22. Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 6645–6649.
23. Will Williams, Niranjani Prasad, David Mrva, Tom Ash, and Tony Robinson, "Scaling recurrent neural network language models," arXiv preprint arXiv:1502.00512, 2015.
24. R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. In NIPS, 2015.
25. Noh, Hyeonwoo, Paul HongsuckSeo, and Bohyung Han. "Image question answering using convolutional neural network with dynamic parameter prediction." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
26. Tang, Yaodong, et al. "Question detection from acoustic features using recurrent neural network with gated recurrent unit." Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016.
27. J. Chung, C. Gulcehre, K.H. Cho, Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.
28. Wang, Fan, and Andrew Lim. "A stochastic beam search for the berth allocation problem." Decision Support Systems 42.4 (2007): 2186-2196.
29. Sabuncuoglu, Ihsan, and M. Bayiz. "Job shop scheduling with beam search." European Journal of Operational Research 118.2 (1999): 390-412.



© 2014- 18, IRJCS- All Rights Reserved 