# Samvaadhana : A Telugu Dialogue System in Hospital Domain

**Suma Reddy Duggenpudi**
International Institute of Information Technology, Hyderabad
sumareddy.duggenpudi@research.iiit.ac.in

**Subrahamanyam Varma**
International Institute of Information Technology, Hyderabad
siva.subrahamanyam@research.iiit.ac.in

**Radhika Mamidi**
International Institute of Information Technology, Hyderabad
radhika.mamidi@iiit.ac.in

## Abstract

In this paper, a dialogue system for Hospital domain in Telugu, which is a resource-poor Dravidian language, has been built. It handles various hospital and doctor related queries. The main aim of this paper is to present an approach for modelling a dialogue system in a resource-poor language by combining linguistic and domain knowledge. Focusing on the question answering aspect of the dialogue system, we identified *Question Classification* and *Query Processing* as the two most important parts of the dialogue system. Our method combines deep learning techniques for question classification and computational rule-based analysis for query processing. Human evaluation of the system has been performed as there is no automated evaluation tool for dialogue systems in Telugu. Our system achieves a high overall rating along with a significantly accurate context-capturing method as shown in the results.

## 1 Introduction

A dialogue system is a computer system which is used for communication with human beings in natural language. It can be used for communication in either written or spoken form. Dialogue systems is a research problem which is being explored very rigorously over the past few years and there are great advancements as well. But despite that, most of the work is limited to English. This might be mainly due to the lack of resources, domain expertise and tools in other languages. Dialogue systems can be broadly classified into two kinds as *Task Oriented Dialogue Systems* and *Non-task Oriented Dialogue Systems* (Chen et al., 2017). Task oriented or domain-specific dialogue systems are systems which handle queries related to a particular task or a fixed domain. The main purpose of such systems is to provide the users with any information or help about that particular chosen domain. On the other hand, Non-task Oriented or Generic Dialogue Systems are modelled to have natural and extended conversations with human beings and can handle multiple domain queries and can act as our assistants.

In this paper, we make an attempt to model a domain-specific dialogue system which answers various queries related to hospitals and their doctors in Telugu. Telugu is an agglutinative South Indian language which belongs to the family of Dravidian languages. It is spoken mainly in Southern India and is also the third most spoken language in India with approximately 93 Million speakers. It is a morphologically rich and highly inflectional language.

Our approach in modelling a domain-specific dialogue system mainly shows that even if there are limited resources like insufficient data, unavailability of linguistic tools etc., still, by taking some suitable measures and creating simple computational tools will lead to the required results. Our dialogue system mainly has two parts namely Question Classification and Query Processing.

**Question Classification:** In this phase, with the help of a question classifier, the question posed by the user is classified into one of the predefined categories which have been designed using the domain knowledge depending on the aim and intention of the question.

For training the question classifier, the data required was manually created. This is possible when the dialogue system is domain-specific which implies that the questions will only be related to a fixed number of categories. In the hospital domain, questions will majorly be related to the categories like timings and availability of the doctor, specialization of the doctor, location of the hospital and so on. This would result in limited questions classes overall.

**Query Processing:** Once the category of the



*Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo)*, pages 234–242

https://doi.org/10.18653/v1/P17

question is known, we process the question using Named Entity Recognition(NER) and extract all the relevant details which are required to answer the question belonging to the particular category. If the information is sufficient to answer the question, then using it, an SQL query is built for retrieving the data which is required to generate a template-answer. But if the information is not sufficient for answering the question belonging to the category, then the user is asked to give the required information following which an SQL query is generated.

Apart from question classification and query processing, context handling is another important task handled by our dialogue system. This serves as the main differentiating factor between a Question Answering system and a Dialogue system. Further, it facilitates the conversation to seem natural.

## 2 Related Work

Dialogue systems is a field that has rigorous research going on. There are many novel systems that have been developed already in English. There can be different kinds of dialogue systems based on the purpose that it serves. One of the very first dialogue systems is ELIZA (Weizenbaum, 1966), which was a deterministic rule-based system. It was one of the first systems to facilitate conversation between man and computer in natural language. Another such early rule-based dialogue system was PARRY (Colby et al., 1971). It was the first dialogue system to pass the Turing Test.

There are other systems like (Chung, 2004), (Zue et al., 2000) and (Ferguson and Allen, 1998) which are mixed-initiative and domain-specific systems. They operate and deliver information only related to a particular domain. In contrast, there are also generic dialogue system architectures which can adapt to domains. (ALLEN et al., 2000) and (Galescu et al., 2018) propose such architectures.

Another kind of dialogue systems is data-driven dialogue systems. They mine conversations from the already available dialogue-corpus. (Serban et al., 2015a), (Jafarpour and Burges, 2010), (Ritter et al., 2011) and (Leuski and Traum, 2011) are some of the systems which are data-driven. They mainly extract the relevant required response using Information Retrieval techniques.

There is another kind of dialogue systems like (Fujie et al., 2019) which mainly work with the user feedback combined with any other technique. This helps in the evolution and learning of the dialogue system. There are also some notable dialogue system like (Vinyals and Le, 2015), (Ritter et al., 2010), (Serban et al., 2015b) and (Mutiwokuziva et al., 2017) which are based on neural networks and deep learning.

In Telugu, the first dialogue system is (Nandi Reddy and Bandyopadhyay, 2006) and it uses computational rules and frames for answer generation. Another dialogue system in Telugu is (Ch. Sravanthi et al., 2015). The authors use various complex linguistic properties of the question to understand the meaning of the query and then process it accordingly.

## 3 Data for the Dialogue System

### 3.1 About The Database

As this is a domain-specific dialogue system which is about Hospitals and can be used to answer questions related to hospitals and doctors in the area of Gachibowli, the database consists information related to hospitals and is used in the last stage of the architecture to generate template-answer. A database consisting the details of four major hospitals in Gachibowli namely Continental hospital, Sunshine hospital, Himagiri hospital and Care hospital was created. The database created mainly contains the following information:

- Name of the doctor
- Hospital in which the doctor is working
- Qualification of the doctor
- Experience of the doctor
- Specialization of the doctor (multiple fields, also includes the department in which they are working)
- Recommendation Rating of the doctor
- Consultation fees of the doctor
- Days of availability of the doctor
- Timings of availability of the doctor

On the basis of the available information in the database, the following question categories were defined for question classification task based on its aim:

- Information about the hospitals in the localities
  - Number of hospitals
  - List of all the hospitals
  - Address of the hospital

- Timings of availability of the doctor
- Specialization of the doctor
- Qualification of the doctor
- Experience of the doctor
- Consultancy fees of the doctor
- Checking the availability of a doctor
  - At a particular time of the day
  - On a particular day of the week
- Information about which hospital a doctor works in

### 3.2 Dataset for Question Classification

There is a lack of dialogue conversational data in Telugu. But, any deep learning technique requires some considerable amount of data for training. And due to this, 388 natural language questions were created initially. Since the categories of the questions asked are finite, the questions posed are also limited. But the 388 questions are not sufficient for training a question classifier of 11 classes. Therefore, we performed Data Augmentation which led to a considerable amount of question data that could be used for training the classifier. This idea has been inspired by (Fadaee et al., 2017) and has been modified according to our requirement.

Data augmentation is done by making slight changes in the already present data to create more data. Even when there is a slight change in the sentence, the system always considers it as a different sentence and that is how the dataset grows. The attribute values like doctor name, hospital name, time and day, were replaced with new values and the tenses were changed to generate new questions which finally become a part of the dataset. There are a total of 28837 questions in this dataset after performing data augmentation. Data Augmentation is done for making the system robust.

For training and testing phases of the classifier, the initial manually created data (388 questions) was split in a ratio of 80% (310 questions) for training and 20% (78 questions) for testing. Then the training and testing data were augmented as described above. It is important to note that we first split the manually written data and then we perform data augmentation separately. This is for proper training and evaluation of the question classifier.

## 4 Question Classification

In this phase, the question posed by the user is classified into one of the already predefined categories depending on the aim of the question. We first get a vector representation of the question with the help of word embeddings[1]. Let the number of words in the question be *N*. Let the $i^{th}$ word in the question $q$ be $q_i$. Now each of these words is embedded into a vector with the help of an embedding matrix *W*. Let the vector representation of the $i^{th}$ word be

$$x_i = W.q_i$$

All the vector representations $x_i$, i $\in [1, N]$ are concatenated as $[[x_1], [x_2], ..., [x_N]]$ and final representation of 2-dimensional matrix for the question $X$ is obtained.

### 4.1 Experiments

Multiple experiments using various deep learning models and machine learning approaches were performed for the question classification task. The results are shown in table 1.

#### 4.1.1 Support Vector Machine

SVM(Cortes and Vapnik, 1995) is one of the most popular machine learning classifier. The question representation $X$ is used as the input for SVM. The final question representations are the main features on which SVM is trained.

#### 4.1.2 Naive Bayes

Naive Bayes classifier (Xu, 2018) is used for question classification with the following features:

1. Bag of words of Unigrams
2. Bag of words of Bigrams
3. TFIDF Values of Bigrams
4. TFIDF Values of both Unigrams and Bigrams combined

#### 4.1.3 Logistic Regression

Logistic Regression (Genkin et al., 2007) is used for predicting the question class with the following features:

1. Bag of words of Unigrams
2. Bag of words of Bigrams
3. TFIDF Values of Bigrams
4. TFIDF Values of both Unigrams and Bigrams combined

[1] https://drive.google.com/open?id=1fEt7aIzYWGQKto3Nt51M5CdjtzxMqdCz

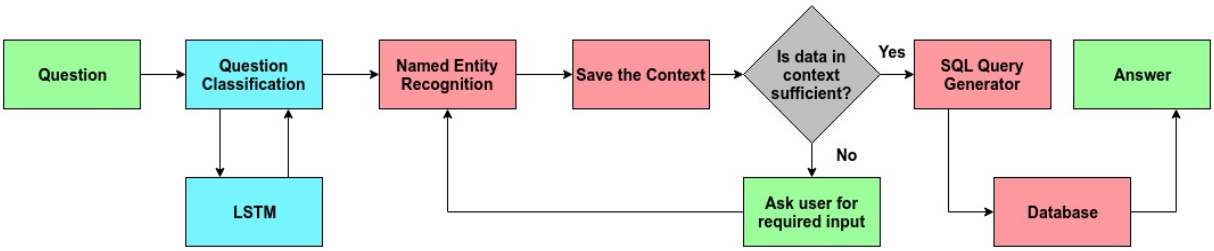


Figure 1: System Flow Diagram

#### 4.1.4 Random Forest

Random Forest(Svetnik et al., 2003) is used for question classification with the following features:

1. Bag of words of Unigrams
2. Bag of words of Bigrams
3. TFIDF Values of Bigrams
4. TFIDF Values of both Unigrams and Bigrams combined

#### 4.1.5 Convolutional Neural Network

Word embedding based model is used for CNN (Kim, 2014), the $X$ matrix is given as input to the CNN model followed by the fully connected layer and finally a softmax layer. The filter of size 4 is used for the convolutions.

#### 4.1.6 Bidirectional LSTM

A single-layer Bidirectional LSTM (Schuster and Paliwal, 1997) is used for question classification. The representation of the question $X$ is given as an input to the Bidirectional LSTM layer. The output of this is then fed into a Dense Layer and then finally softmax is performed. The hidden dimension of Bi-LSTM is 64. The dropout rate is set 0.4 for avoiding overfitting.

#### 4.1.7 Long Short Term Memory

A single-layer LSTM (Hochreiter and Schmidhuber, 1997) model has been implemented for classification. The input to LSTM layer is the concatenated representation of the question $X$ and the output is passed to a dense layer and finally, softmax is performed to predict question category. The hidden dimension of LSTM is 32. A total of 10 epochs were performed and the drop out rate is set to 0.2.

## 5 Named Entity Recognition

After the classification of the question, the question category designated by the classifier along with the question posed by the user is analyzed to extract the important information from the question which is required for SQL query generation. This information is predominantly named entities and so we use Named Entity Recognition (NER). The named entities which were defined are the following:

1. Name of the Doctor
2. Name of the Hospital
3. Time
4. Date or Day
5. Name of the Locality
6. Hospital domain related Technical Terms

For answering the majority of the questions, the required information is mostly the named entities that belong to the above list. There is no readily available computational tool for Named Entity Recognition in Telugu for our domain. Therefore, as proposed in (Srikanth and Narayana Murthy, 2019), a hybrid model is designed which combines both heuristics and rules, based on the nature of the language and patterns in occurrences for identifying named entities. Here heuristics implies some simple probable cases like, in the context of the question, there is a high probability of finding the Name of the Doctor, as the next two words right after the *doctor* or *Dr.* tag and it is also likely to find the name of the hospital right before the *hospital* tag. Tags like *a.m.* and *p.m.* can be used as a clue to find the time intended in the question right before these tags. It is important to note that there can also be some ambiguities, but since this data is domain-specific, there are less chances of facing such ambiguities.

Apart from such heuristics, some rules were also designed on the basis of nature of the language for identifying the named entities. These are more focused on the language. For example, mostly when the case marker *'ki'* occurs, it is preceded by *time* in the question. In another instance, whenever *'lo'* occurs, it is a case marker which is associated with *location*. We also have some definitive rules like *'gAaru'* is always followed by

| User Query | System Response |
|---|---|
| కాంటినెంటల్ హాస్పిటల్లో పని చేస్తున్న డాక్టర్ అనూషమేక గారు ఎప్పటి నుంచి ఎప్పటి వరకు ఉంటారు?<br>What are availability timings of Dr. Anusha Meka who works in Continental Hospital? | కాంటినెంటల్ లో పని చేస్తున్న డాక్టర్ అనూష మేక 9:00 AM-5:00 PM వరకు అందుబాటులో ఉంటారు.<br>Dr. Anusha Meka, who works at Continental Hospital, is available from 9:00 AM-5:00 PM. |
| తనకు ఎంత ఎక్స్పీరియన్స్ ఉంది?<br>What is her experience? | కాంటినెంటల్ లో పని చేస్తున్న డాక్టర్ అనూష మేకకి 3 సంవత్సరాల ఎక్స్పీరియన్స్ ఉంది.<br>Dr. Anusha Meka, who works at Continental Hospital, has an experience of 3 years. |
| క్వాలిఫికేషన్ ఏంటి?<br>What about qualification? | కాంటినెంటల్ లో పని చేస్తున్న డాక్టర్ అనూష మేక క్వాలిఫికేషన్ BDS.<br>Dr. Anusha Meka, who works at Continental Hospital, has a qualification of BDS. |
| ఫీజు ఎంత?<br>What is the fee? | కాంటినెంటల్ లో పని చేస్తున్న డాక్టర్ అనూష మేక 500 రూపాయలు ఫీజు తీసుకుంటారు.<br>Dr. Anusha Meka, who works at Continental Hospital, charges 500 rupees as fees. |
| సాయంత్రం ఐదు గంటలప్పుడు అందుబాటులో ఉంటారా?<br>Will she be available at 5 in the evening? | కాంటినెంటల్ లో పని చేస్తున్న డాక్టర్ అనూష మేక ఐదు గంటలప్పుడు అందుబాటులో ఉంటారు.<br>Dr. Anusha Meka, who works at Continental Hospital, will be at 5. |

Figure 2: Capturing Linguistic Information from the Context of Chat

| User Query | System Response |
|---|---|
| డాక్టర్ ఎంత ఫీజు తీసుకుంటారు?<br>How much fees does doctor charge? | దయ చేసి హాస్పిటల్ పేరు తెలియచేయండి.<br>Please provide with hospital's name. |
| కాంటినెంటల్ హాస్పిటల్ గురించి అడుగుతున్న.<br>I am asking about Continental Hospital | దయ చేసి డాక్టర్ పేరు తెలియచేయండి.<br>Please provide with doctor's name. |
| డాక్టర్ పేరు అనూష మేక.<br>Doctor's name is Anusha Meka. | కాంటినెంటల్ లో పని చేస్తున్న డాక్టర్ అనూష మేక 500 రూపాయలు ఫీజు తీసుకుంటారు.<br>Dr. Anusha Meka, who works at Continental Hospital, charges 500 rupees as fees. |

Figure 3: System's Response to Missing Data

the doctor's name. It is not necessary that these rules and are also accurate always, but when the previous knowledge of the question category is combined with these rules, it is most likely that the required named entities are found.

## 6 Check for Answer Retrieval

After performing NER on the input question, the next aim is to generate an SQL query for the given question with the respective attribute values and then to retrieve the answer. But, if there is insufficient data in the context, then the SQL query cannot be generated. It is important to check this first. There are 11 categories of questions that can be handled by the system. For each of those categories, there are a set of attributes which are mandatory for answering the question. If they are not present in the context, then the system reverts back to the user asking for the required information. When the user responds with the information, the context is updated. If this context is sufficient for answering the question, then an SQL query is generated, else the same process is repeated until all the required information is available. The same is conveyed with real-time examples of our system in Figure 3.

From the example in Figure 2, it is understood from the first question that the conversation is about Dr. Anusha Meka. Now as a continuation to the first question, the user asks questions like *How much experience does she have?* or *How much is the consultation fee?*. The basic necessity of the dialogue system is to be able to understand how these questions are related to the first question and to have information as the context while answering these questions. To know that these questions are

about Dr. Anusha is the pre-context that is being captured by the system and the further questions are answered accordingly. This is also primarily done by maintaining the context in every level of the dialogue. When a new question comes up without any contextual information, then the system goes back to the context available, looks for the attribute values and fills the missing attributes required for answering the question. When a new question comes up with a different doctor's name from the previous context, then it is assumed that this question is of different context, hence the previous context is flushed and the new context from this question is registered. With this process, context is grabbed and the output also seems more natural and realistic, and this property lets the system and user engage in a normal, natural and complete conversation, which is close to the real-world human-human conversation.

## 7 SQL Query Generation and Answer Retrieval

There are a total of 11 question categories that are handled. Each question frame has a definite and fixed SQL query. After the question is completely processed and once the required information for answering the question is available, it is put into the attribute blanks of SQL query accordingly. Then this query is given to the SQL database where all the information regarding the hospitals is stored. The attribute values which are required to build the template-answer are retrieved from the database and finally, the template-answer is generated and returned back to the user.

## 8 Other Simple Handled Issues

Apart from the detailed framework presented above, there is a need to handle some challenging linguistic issues to enhance the dialogue system and make the conversations more natural.

### 8.1 Anaphora Resolution

In this system, as a part of context handling, it is important for the system to understand various kinds of references. If there is a pronoun in a question, then the system should understand what is the actual reference to that pronoun. In this system, for pronoun handling, simple rule-based anaphora resolution is modelled. For example, if there is a pronoun intended for female, like '*Ame*'(she) then the system looks for a female doctor in the context available.

### 8.2 Resolution of ambiguity in names

It is very likely that there are two or more doctors with the same first name and the user also generally addresses the doctor with the first name. In such a case, it is important for the system to understand which of the doctors is being referred to by the user. For this, the system prompts the user to select the doctor from a list of doctors having that same first name.

### 8.3 Handling Spelling Mistakes

It is possible that users can very easily misspell the name of a doctor or hospital because proper nouns can have many versions of pronunciations and corresponding spellings as well. Therefore, to find out what is exactly being referred to, character level matching is done and the similarity score is calculated with Levenshtein distance(Miller et al., 2009) between the user's spelling and all the names in the database. Based on the similarity score, the one with the highest and which passes the cutoff score is chosen as the correct spelling.

## 9 Results

### 9.1 Question Classification

Several models have been used for the task of Question Classification. The accuracies have been reported in Table 1.

| Model | Accuracy |
|---|---|
| Support Vector Machine | 68.778% |
| Naive Bayes+BOW+Unigrams | 86.046% |
| Naive Bayes+TFIDF+Unigrams | 83.721% |
| Naive Bayes+TFIDF+Bigrams | 83.721% |
| Naive Bayes+TFIDF+Both | 81.395% |
| Logistic Regression+BOW+Unigrams | 90.697% |
| Logistic Regression+BOW+Bigrams | 93.023% |
| Logistic Regression+TFIDF+Bigrams | 88.372% |
| Logistic Regression+TFIDF+Both | 90.697% |
| Random Forest+BOW+Unigrams | 95.348% |
| Random Forest+BOW+Bigrams | 90.697% |
| Random Forest+TFIDF+Bigrams | 95.348% |
| Random Forest+TFIDF+Both | 95.348% |
| Convolutional Neural Network | 73.423% |
| Bidirectional LSTM | 77.136% |
| **Long Short Term Memory** | **99.326%** |

Table 1: Experiment Results of Question Classification

We can notice that LSTM outperforms all the other algorithms. It is also important to have such high accuracy because if the question is classified wrong, then the output generated will also be wrong eventually.

### 9.2 Dialogue System

For evaluating a dialogue system, there is no automated evaluation tool available. Hence the system was manually evaluated by 8 people. The evaluators were native Telugu speakers. A special User Interface was created for easy evaluation of the system. After every answer from the system, the evaluator was expected to mark the response as 'correct', 'not sure' or 'incorrect'. Everyone evaluated the system for about 20-30 dialogues(here dialogue is a conversation between the user and the system until the answer is retrieved). A total of 195 responses were recorded. Table 2 shows the ratings given by the evaluators in various aspects for judging the overall performance of the system. The scaling followed is 0-5, where 0 being Poor and 5 being Excellent. The Table 3 shows the accuracy metrics.

| Metric | Percentage |
|---|---|
| Correct/Total | 88.717 |
| Correct/(Correct+Incorrect) | 90.769 |

Table 3: Human-evaluation accuracy metrics

## 10 Conclusion

This work mainly combines both deep learning techniques as well as rule-based computational techniques. Though this approach is domain-specific, it can be easily extended to any other domain as well. It only requires the creation of some domain-specific data and some domain-specific rules and heuristics. Even if the data is little, using some simple techniques like Data Augmentation and standard classifier gives good results and serves the required purpose. This can really be helpful with resource-poor languages. With such vast applications of the dialogue system, this is definitely one step closer to creating dialogue systems in resource-poor languages.

## 11 Future Work

Our future work would mainly be focused on working with Telugu-English Code-Mixed Data as more commonly used in Telugu speaking regions. Another thing that we would focus more on is error handling, that is basically to identify a completely irrelevant question as an irrelevant one and also will try to handle Out-of-Vocabulary(OOV) words. We are also looking forward to design better heuristics for handling spelling mistakes. Also, using the available recommendation ratings in the database, we would try to inculcate the doctor recommendation system also as a part of this Dialogue System. The objective would be to recommend a doctor according to the patient's request or even based on the diseases/symptoms. Apart from that, we would also like to make this a multi-domain dialogue system which would consist of information from multiple domains and switching between the domains in the conversation would also be facilitated.

## References


JAMES ALLEN, DONNA BYRON, MYROSLAVA DZIKOVSKA, GEORGE FERGUSON, LUCIAN GALESCU, and AMANDA STENT. 2000. An architecture for a generic dialogue shell. *Natural Language Engineering*, 6(3-4):213228.

Mullapudi Ch. Sravanthi, Prathyusha Kuncham, and Radhika Mamidi. 2015. A dialogue system for telugu, a resource-poor language.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *CoRR*, abs/1711.01731.

Grace Chung. 2004. Developing a flexible spoken dialog system using simulation. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 63–70, Barcelona, Spain.

Kenneth Mark Colby, Sylvia Weber, and Franklin Dennis Hilf. 1971. Artificial paranoia. *Artif. Intell.*, 2:1–25.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Mach. Learn.*, 20(3):273–297.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. *CoRR*, abs/1705.00440.

George Ferguson and James F. Allen. 1998. Trips: An integrated intelligent problem-solving assistant. In *AAAI/IAAI*.

Shinya Fujie, Riho Miyake, and Tetsunori Kobayashi. 2019. Spoken dialogue system using recognition of user's feedback for rhythmic dialogue.

| Metric | Description | Average Rating |
|---|---|---|
| Relevance/Correctness | How relevant/correct is the answer retrieved? | 4.375 |
| Robustness | How robust is the system? | 4.25 |
| Intention Identification | How well does the system identify the intention of the question posed? | 4.625 |
| Context Capturing | How well is the context captured in a full-fledge dialogue conversation? | 4.25 |
| System Responsiveness | How responsive is the system? | 4.562 |
| Reliability | Is the system helpful in providing all the relevant information? | 4.25 |
| Complex Sentences | How well are the complex sentences handled? | 3.75 |
| Missing Data | How well are questions with missing data handled? (with respect to the system responses) | 4.312 |
| Error Handling | How well are errors handled? (Eg: Absence of data in the database) | 4.0 |
| Overall Rating | How much will you rate the system overall? | 4.343 |

Table 2: Human Evaluation Results


Lucian Galescu, Choh Man Teng, James Allen, and Ian Perera. 2018. Cogent: A generic dialogue system shell based on a collaborative problem solving model. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 400–409, Melbourne, Australia. Association for Computational Linguistics.

Alexander Genkin, David D Lewis, and David Madigan. 2007. Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304.

Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Sina Jafarpour and Chris J.C. Burges. 2010. Filter, rank, and transfer the knowledge: Learning to chat. Technical Report MSR-TR-2010-93.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Anton Leuski and David Traum. 2011. Npceditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine*, 32:42–56.

Frederic P. Miller, Agnes F. Vandome, and John McBrewster. 2009. *Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press.

Milla T Mutiwokuziva, Melody W Chanda, Prudence Kadebu, Addlight Mukwazvure, and Tatenda Trust Gotora. 2017. A neural-network based chat bot. *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, pages 212–217.

Rami Reddy Nandi Reddy and Sivaji Bandyopadhyay. 2006. Dialogue based question answering system in telugu. pages 53–60.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. pages 583–593.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. *Proceedings of HLT-NAACL*.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2015a. A survey of available corpora for building data-driven dialogue systems. *CoRR*, abs/1512.05742.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015b. Hierarchical neural network generative models for movie dialogues. *CoRR*, abs/1507.04808.

P Srikanth and Kavi Narayana Murthy. 2019. Named entity recognition for telugu.

Vladimir Svetnik, Andy Liaw, Christopher Tong, John Culberson, Robert P Sheridan, and Bradley Feuston. 2003. Random forest: A classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43:1947–58.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR*, abs/1506.05869.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

Shuo Xu. 2018. Bayesian nave bayes classifiers to text classification. *Journal of Information Science*, 44(1):48–59.

V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington. 2000. Jupleter: a telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1):85–96.