# Natural Language Processing

## IST 664 / CIS 668

# Natural Language Processing Application Investigation Report

# Human Computer Interfaces and Dialogues, Including Chatbots

By,

Akshay Aravind (aaravind@syr.edu)
Laxmipriya Neha Majety (lmajety@syr.edu)
Sanjana Rajagopala (sarajago@syr.edu)
Vinutha Karanth (vkaranth@syr.edu)

# Table of Contents

# Abstract

Artificial Intelligence (AI) is a field of study in intelligent agents and has been in existence since the Turing machine and much beyond. There have been great advancements in AI over the past few decades. One of the major areas of AI is concerned with the interaction between computers and human (natural) languages. Most natural language processing (NLP) systems were based on complex sets of hand-written rules up to the 1980s. Starting in the late 1980s, however, there was a revolution in NLP with introduction of machine learning (ML) algorithms for language processing. A **chatbot** (talkbot, chatterbot, bot or interactive agent) is a computer program which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Most chatbots are either accessed via virtual assistants such as Google Assistant, Amazon Alexa, Apple Siri or via individual organizations' apps and websites. There are various platforms available to create a chatbot such as msg.ai, wit.ai, Dialogflow, Microsoft Language Understanding Intelligent Service (LUIS) etc. **Dialogflow** was chosen for this project because, the platform is very intuitive, the ease of creation of the chatbot was better when compared to other platforms, provides support for large set of languages and it provides integrations to various third-party apps. But most of the platforms follow the same concepts and terminologies.

Chatbot creation can be divided into design, building, analytics and maintenance. The chatbot *design* is the process that defines the interaction between the user and the chatbot. We as the chatbot designers defined the chatbot personality, the questions that will be asked to the patients (users), and the overall interaction. An important part of the chatbot design is also centered around user testing. Dialogflow provides out-of-the box techniques for testing and training. The process of *building* a chatbot involves two tasks: understanding the user's intent and producing the correct answer. In our case, the first task was to understand the user input and we have narrowed our inputs to a small set of questionnaires to demonstrate the NLP techniques. The responses are generated depending on the first task. Dialogflow provides very good *analytics* to monitor the usage of the chatbot and helps us improve accuracy and the overall user experience. We leveraged most of the features that Dialogflow provides and we even developed our own website and embedded our chatbot in it.

Our zest to learn and enjoy what we do made us go above-and-beyond to develop an application that will be deployed to the cloud on a voice command. We just have to say, "deploy NLP app" to **Google Assistant** on our phone and the code from our **GitHub** repository will be deployed on the cloud platform **Heroku**. Throughout the course of the project, we learnt a lot of new state-of-the-art technologies, got insights into the actual understanding of chatbots, explored various NLP techniques, understood the richness and power of NLP tools that are latest in the field and followed professional project management using **Trello**. We thoroughly enjoyed building our own chatbot from scratch and learning the advancements in the chatbots, NLP, machine learning and in general AI.

# 1. Introduction

**Human-Computer Interaction (HCI)** researches the design and use of computer technology, focused on the interfaces between people (users) and computers. The notion of an open-ended dialog between the user and computer likens human-computer interaction to human-to-human interaction. This is exactly what we achieved in our chatbot by taking mental illness as a use case and building a conversation with the user (patient). An important facet of HCI is the securing of user satisfaction. **Human-computer interface** can be described as the point of communication between the human user and the computer. The flow of information between the human and computer has several aspects: visual based, audio based, machine environment, input flow, output flow and feedback.

A **dialog system** or **conversational agent (CA)** is a computer system intended to converse with a human, with a coherent structure. **Chatbots** are typically used in dialog systems for various practical purposes and use sophisticated NLP systems. The criterion of intelligence depends on the ability of a computer program to impersonate a human in a real-time written conversation with a human judge, sufficiently well that the judge is unable to distinguish reliably between the program and a real human.

**Dialogflow** (formerly api.ai) is Google-owned developer of human-computer interaction technologies based on natural language conversations. Dialogflow helps us build natural and rich conversational experiences that gives users new ways to interact with the product by building engaging voice and text-based conversational interfaces powered by AI. Dialogflow incorporates Google's machine learning expertise and products such as Google Cloud Speech-to-Text and is backed by Google Cloud Platform which can easily scale millions of users. Using years of domain knowledge and NLU, Dialogflow analyzes and understands the user's intent and responds in the most useful way.

**Achelois** which means "the Greek moon goddess who washes away pain" - has been built using Dialogflow to understand the mental health of a patient and have a conversation with them by understanding their problems, suggesting remedies and trying to make them happy. We have also developed a simple website and Achelois is embedded in it. It can be found at **https://nlp-spring18-ui.herokuapp.com/**

# 2. NLP Techniques and Research

The three important NLP viewpoints with respect to chatbots and other dialogue systems are:

1. Natural Language Understanding (NLU)
2. Natural Language Generation (NLG)
3. Dialogue Processing

## 2.1 Natural Language Understanding (NLU)

NLU involves the process of extracting meanings from text inputs. In this direction, the basic design steps include:

- **Syntactic Parsing** - determines the function of each word (part-of-speech), the way words are related to each other, how they are grouped into phrases and how they can modify each other. The context-free

grammar (CFG) definition and parsers implementation are the common NLP techniques used in this step.

- **Semantic Parsing** - The role of a semantic parser is to extract the context-independent meaning of a written sentence. The discriminative methods such as support vector machines (SVM) and statistical methods such as decision trees and classification and regression trees (CART) are used to find the most probable parse tree that fits the sentence.
- **Contextual Interpretation** - refines the semantic interpretation by taking advantage of information at the discourse level and, removing remaining ambiguities such as anaphors, pronouns and ellipses. Discourse entity (DE) list maintains a set of constants referring to objects that have been evoked in previous sentences and can subsequently be referred implicitly.

## 2.2 Natural Language Generation (NLG)

NLG involves the building of responses for the chatbots based on the processing done in the NLU step. (Sameera A. Abdul-Kader, 2015)

- **Artificial Intelligence Markup Language (AIML)** – an XML-compliant language designed to develop the AI flows in dialogue systems. Its purpose is to simplify the job of conversation modelling. The general structure of the AIML objects looks as follows:

```
<category>
   <pattern> User input </pattern>
   <template> Corresponding response to the user input </template>
</category>
```

The important elements of AIML are:

  - **Categories** – basic units of knowledge consisting of patterns and templates
  - **Recursion** – used to recursively match the other categories and thereby, simplify the complex grammatical forms. It is represented as <*srai*> tag
  - **Context** – the category tag uses the context <*that*> tag to refer to the previous input of the user
  - **Variables** – used to support the getting and setting of commonly used texts using <*get*> and <*set*> tags that are generally used to store proper nouns.
  - **Pronoun Swapping** – used for replacement of pronouns such as ‘you’re’ with ‘I’m’ and ‘your’ with ‘my’ etc.

- **Pattern matching** – mainly used in question-answering chatbot systems for simple statements and natural language enquiring.
- **Response generation** (Woudenberg, 2014)
  - Document planning – brakes the high-level communicative goals into structured representations of atomic communicative goals
  - Microplanning (Sentence planning) – is the phase where the number of generated clauses is decided in order to produce language with improved naturalness. Methods such as semantic grammars – reverse parsing are used in generation of unnatural proto-phrases.

  - Surface realization – the process of transforming the abstract structure obtained in the microplanning stage into surface linguistic structures by adding function words, inflecting words, determining word order etc.
- **Chat script** - the technique that helps when no matches occur in AIML. It concentrates on the best syntax to build a sensible default answer.
- **Markov chain** - used to build responses that are more probabilistically applicable and consequently are more correct. The idea of Markov chains is that there is a fixed probability of occurrences for each letter or word in the same textual data set.
- **Language tricks** - includes the sentences, phrases, paragraphs available in chatbots in order to add variety to the knowledge base and make it more convincing. The types of language tricks are: canned responses, typing errors and simulating key strokes, model of personal history and Non-Sequitur (not a logical conclusion). Each of these language tricks is used to satisfy a specific purpose and to provide alternative answers to questions.
- **Ontologies/Semantic networks** – consists of set of relationally and hierarchically interrelated concepts. The aim of using ontologies in a chatbot is to compute the relation to synonyms, hyponyms and other relations which are natural language concept names. The interconnection between these concepts can be represented in a graph enabling the computer to search by using particular rules for reasoning.

## 2.3 Dialogue processing

Aims at building a man-machine dialog based on turn-taking process wherein the information is transferred from one participant – the user and the other participant – dialog manager (DM). The process of dialogue management involves implementation of the interaction strategy to organize the sequence of system dialogues to achieve the common goal of the user and chatbot. The level of user satisfaction of a chatbot is heavily influenced by the concept of "Degree of Initiative" –

- **System-Led** – an initiative that is completely controlled by the system wherein it asks precise questions to the user and expects information or answers from the user
- **User-Led** – an initiative that is led by the user while the system is expected to provide information to the user queries without asking for more details
- **Mixed-Initiative** – an initiative in which both the participants interact cooperatively to achieve the conversation goal. For example, our health chatbot shares the control between the user and system.

**Relational databases** enable the chatbot to remember past conversations and make the conversation more continuous and meaningful by building knowledge bases. The Structured Query Language (SQL) allows generation of queries and query block nesting to save the conversation history. This mainly makes the search of a word and phrase match easier.

## 2.4 Types of chatbot models

In general, we have two types of chatbot models:

| | **Retrieval - Based** | **Generator - Based** |
|---|---|---|
| **Description** | Simple and pre-defined heuristic model | Smart model generating new response from scratch |
| **Grammatical errors in responses** | Rarely involves errors in grammar | Can be prone to grammatical errors |

| | | |
|---|---|---|
| **Behavior towards unseen cases** | Not handled | Handles efficiently |
| **Contextual responses** | Not considered | Considers context for response generation |
| **Length of conversation** | Long | Short |
| **Domain** | Open | Closed |

Table-1.0

## 2.5 Chatbot processing mechanism

Figure-1.0 shows a flowchart of the processing mechanisms for a chatbot.

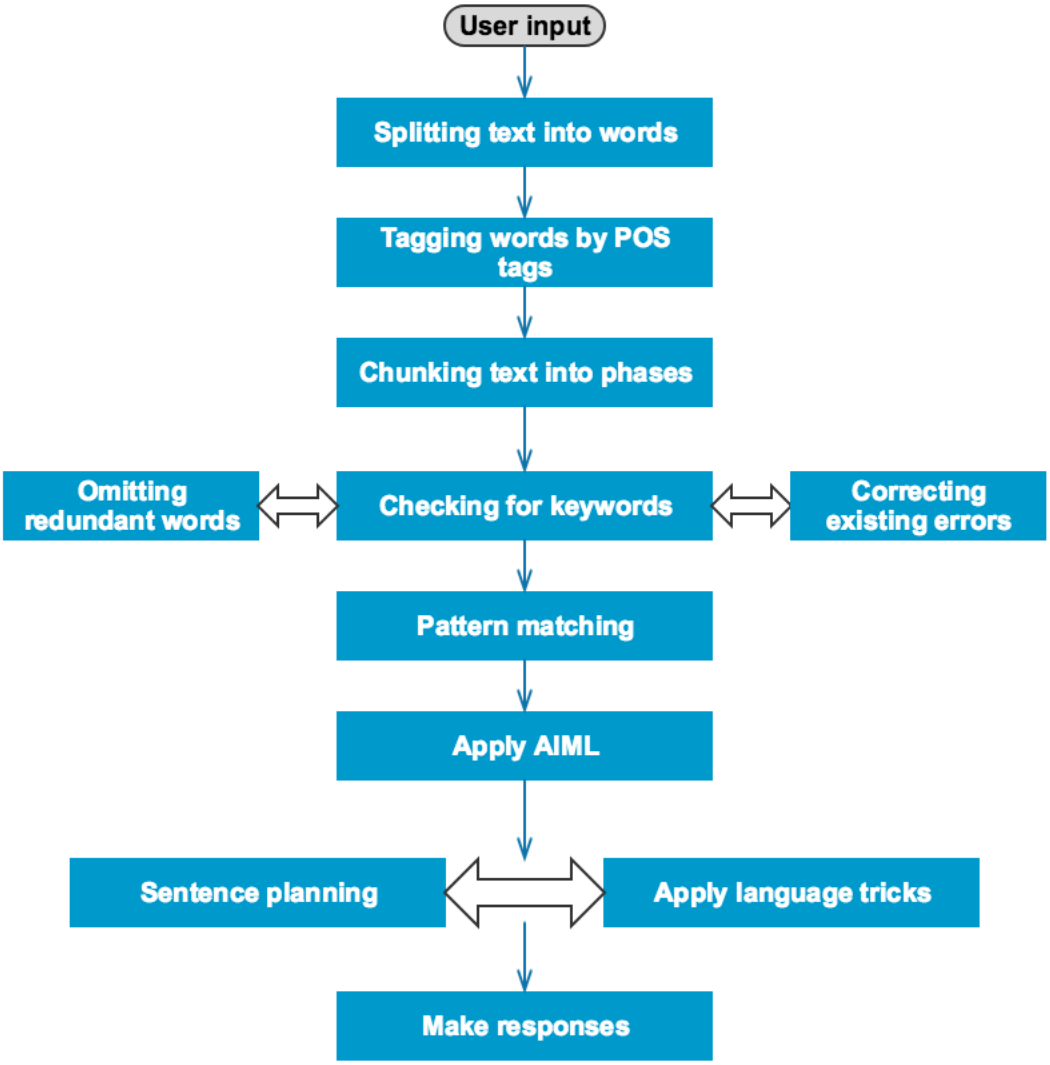


Figure-1.0

# 3. Chatbots

Chatbots and virtual assistants are becoming part of our daily life and they are also becoming smarter every day. There is a lot of scope, many inventions to be made in this domain and the use cases are plenty. This

was one of our motives to choose this topic. We explored the various platforms and tried our hand in three of the famous ones: wit.ai, luis.ai and Dialogflow. Table-1.1 summarizes these three platforms.

## 3.1 Comparison chart

(Javier, 2015)

| Platforms | wit.ai | luis.ai | Dialogflow |
|---|---|---|---|
| Description | • Allows to easily create text or voice-based bots that humans can chat with on their preferred messaging platform.<br>• It allows to understand user input after certain training, identify intents, extract entities, and predict what the bot should do based on the current context and user query. | • Introduced by Microsoft. The Microsoft Bot framework helps us to build, test and deploy bots for many well-known platforms such as Facebook, Slack etc.<br>• There are some pre-built domains that we can import to our chatbot together with its entities, intents and utterances. | • Formerly known as api.ai, this Google acquired platform is popular for building conversational interfaces.<br>• Offers more than 30 pre-built agents such as navigation, hotel booking, small talk, weather, news etc. |
| Pros | • The concept of story is really powerful and allows controlling the conversation flow using branches and conditions on actions.<br>• An "Inbox" exists, where the requests that could not be processed by the chatbot are listed, so the developers can train the bot. | • More feasible for professional purposes and .NET developers.<br>• The enterprise version can be integrated with other application provided by Microsoft Azure to enhance functionalities. | • One-click integration with several platforms like Slack, Telegram, Facebook Messenger, Twitter, Amazon Alexa, Google Assistant etc.<br>• Built-in voice interface and support for Node.js, Ruby, Android, iOS, Python etc. SDKs. |
| Cons | • wit.ai has integration with Facebook messenger only. Stories are in beta.<br>• There are cases where it is difficult to control the flow of the conversation and the bot tends to misunderstand the user requests. | • luis.ai has 10,000 transactions for each month and is priced after that. | • Priced for enterprises requiring high utilization of Dialogflow<br>• It is impossible to block the matching of an intent if a context is present.<br>• The training section is still in beta |

Table-1.1

Figure-1.1 and Figure-1.2 shows the screenshots of wit.ai and luis.ai console.

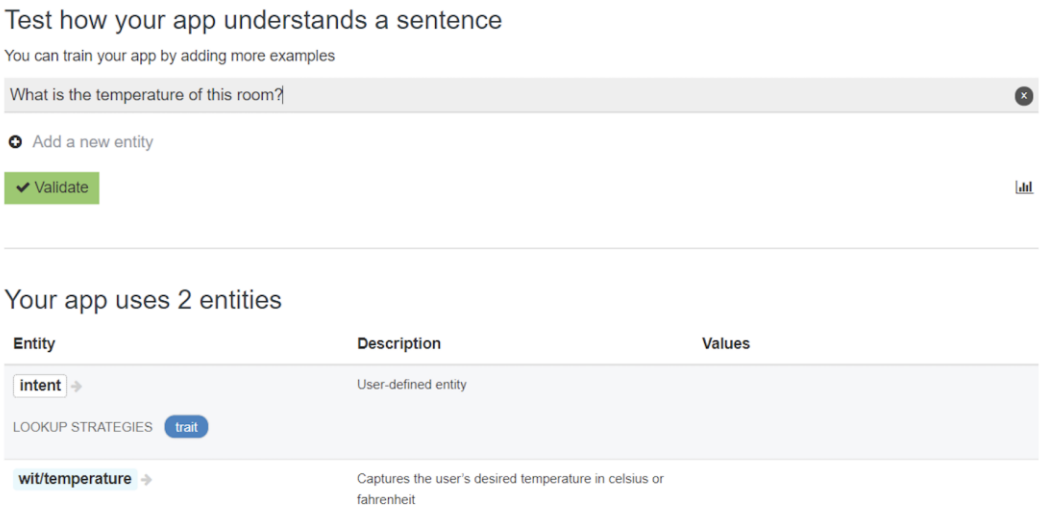


Figure-1.1

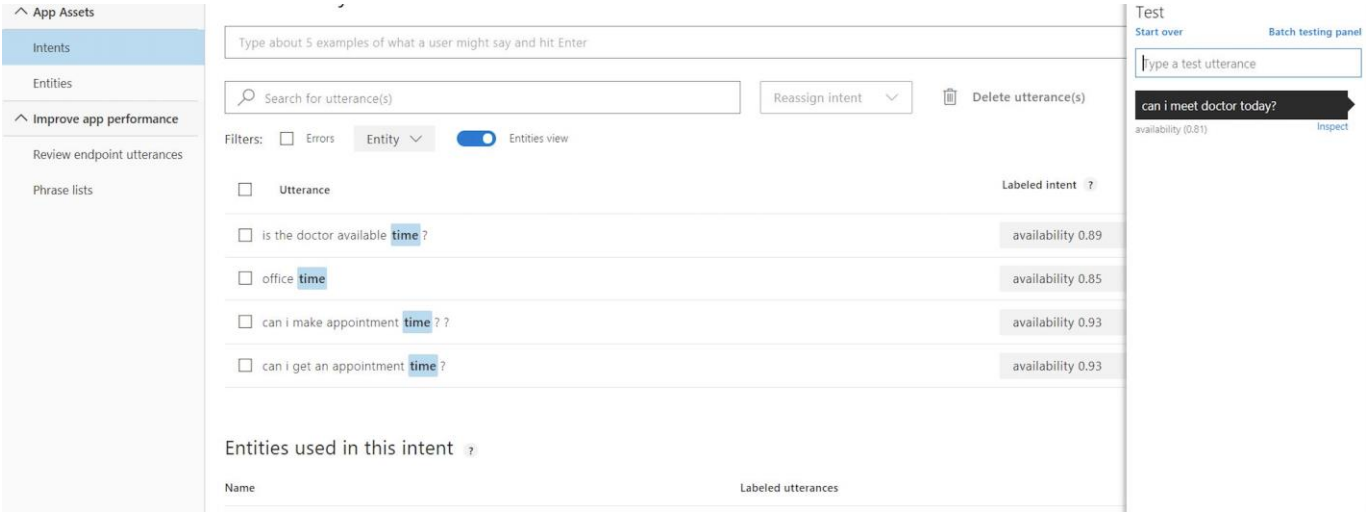


Figure-1.2

# 4. Healthcare Chatbots

Healthcare domain was an interesting and challenging area for us to target. There are already chatbots available in areas like flight booking, hotel management, food ordering services, home automation etc. Healthcare is one of the areas where there is a lot of research and opportunities with respect to utilizing chatbots. Most of the chatbots available in this domain deal with hospital management like scheduling an appointment, getting to know the availability of the doctors, managing prescriptions, heart rate and blood pressure monitoring, reporting symptoms and illness etc. Some of the healthcare chatbots in these domains are Your.MD, Florence and GYANT etc. We wanted to build a virtual doctor which mimics a real doctor in terms of conversation, treatments and suggestions to improve the mental health of patients.

# 5. Dialogflow

**Achelois** is the chatbot or the agent that we have built using Dialogflow and can be found in action at **https://nlp-spring18-ui.herokuapp.com/**. We leveraged most of what Dialogflow has to offer to build Achelois. It understands the feelings and problems of a patient struggling with their mental health and provides suggestions to make their life better and happy. The scope of the problems is vast, but we have narrowed to a small set such as getting to know the mood of the patient, hear about any incidents that have had happened, appetite problems etc. (Build natural and rich conversational experiences, n.d.)

The process a Dialogflow follows from invocation to fulfillment is similar to someone answering a question, with some liberties taken of course. In order to start a conversation with an agent, the user needs to invoke the agent. A user does this by asking to speak with the agent in a manner specified by the agent's developer. Figure-1.3 shows how we begin the conversation with Achelois. For the agent to understand what the user said, it needs examples of how the same text can be said in different ways. These are added to training phrases. Figure-1.4 shows the user expressions that we have configured for the welcome intent. Note that, we do not have "Hi, good morning" as single user text in

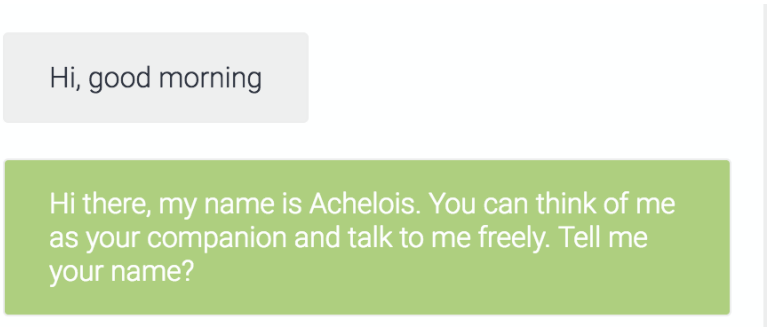


Figure-1.3

Figure-1.4. But the agent is trained using NLP and machine learning techniques such as word embedding, user data, lexical synonyms, bag-of-words, synonym detection, regular expressions, tokenization, tags, supervised learning etc. and hence was able to detect what the user said.

We do not need to input all possible user inputs, but only need to put in few sentences or words and the powerful algorithms in Dialogflow takes care of the rest. We observe a lot of other techniques here in this small example. The case of the words is not taken into consideration, but punctuation and special characters are. The Dialogflow agent needs to know what information is useful for answering the user's request.

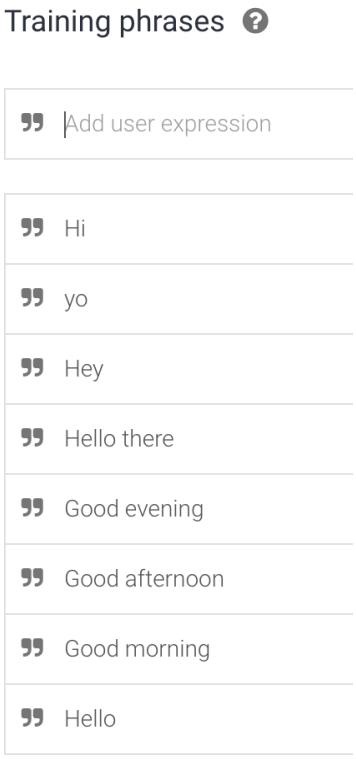


Figure-1.4

## 5.1 Agents

Agents are best described as NLU modules. These are included in the app to transform natural user requests into actionable data. Agents can also be designed to manage a conversation flow in a specific way. There are a number of pre-built agents available in Dialogflow.

## 5.2 Intents

An intent represents a mapping between what a user says and what action should be taken. Intent interfaces have the following sections:

- Training Phrases
- Action
- Response
- Contexts

Figure-1.5 shows the intents in our agent and Figure-1.6 shows training phrases for "1.0 – getMood" intent. There are around 40 training phrases for this intent. The reason is, getting the mood of a person is a very diverse question. And the possibilities on how a user might respond depends on each individual. Hence, we have a lot of phrases here and the NLP and ML algorithms train these sentences and can understand similar sentences from the user even if there are typos. More the training phrases, more the agent learns. The system detects the correspondence between the words (or phrases) and existing developer and system entities and highlights such words and phrases. It also automatically assigns a parameter name to each detected entity. Figure-1.7 shows the action and parameters for getMood intent. The colors in Figure-1.6 correspond to the entities shown in Figure-1.7. For example, the word awful is negative and is tagged with negativeFeelings entity.

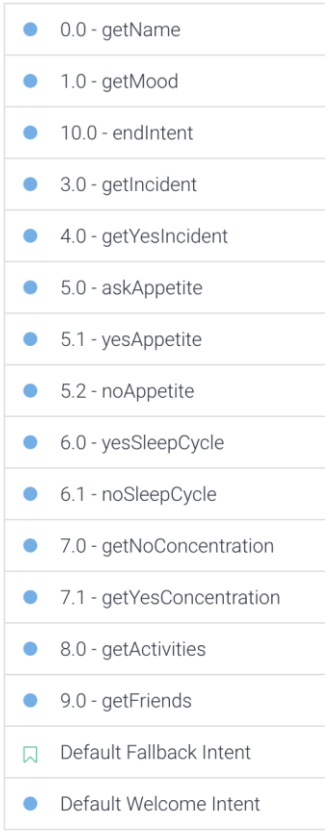


Figure-1.5

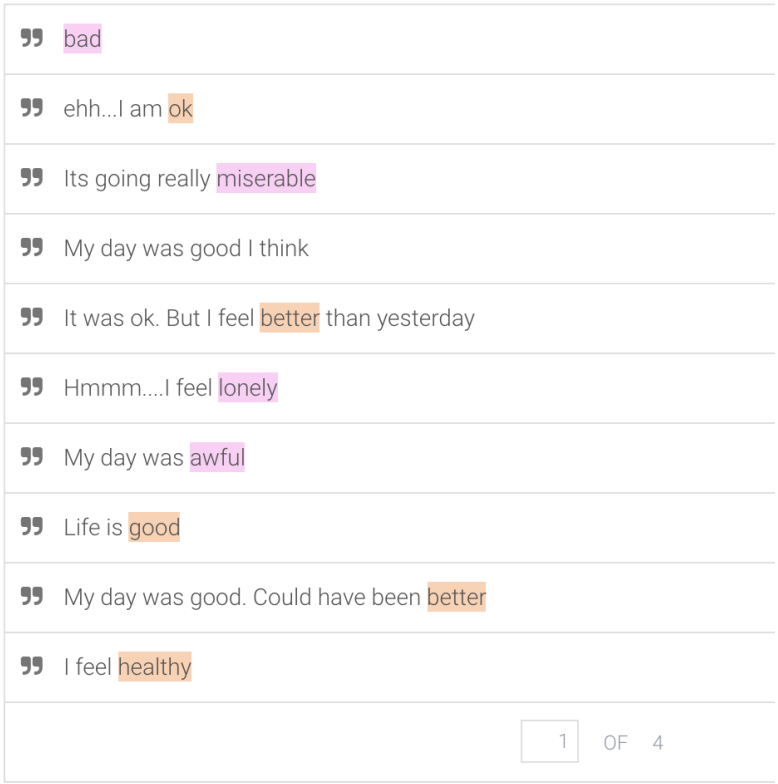


Figure-1.6

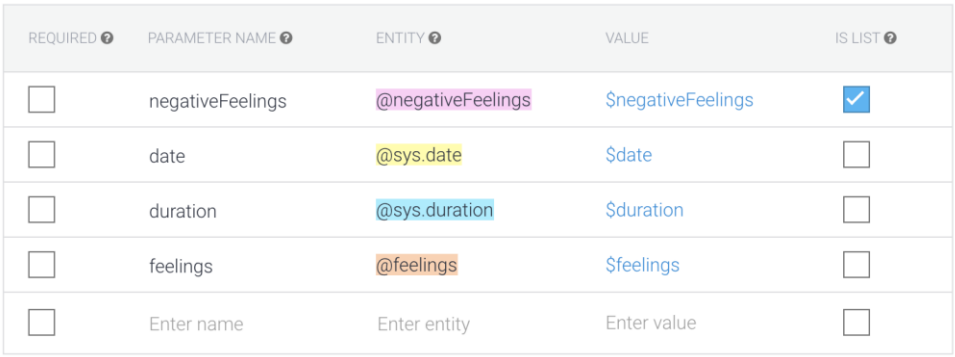


Figure-1.7

Figure-1.8 shows the agent's responses which will be provided by the agent when the intent is triggered. One of the responses is randomly selected and this in turn keeps the conversation going. Different responses will make the agent more human-like.

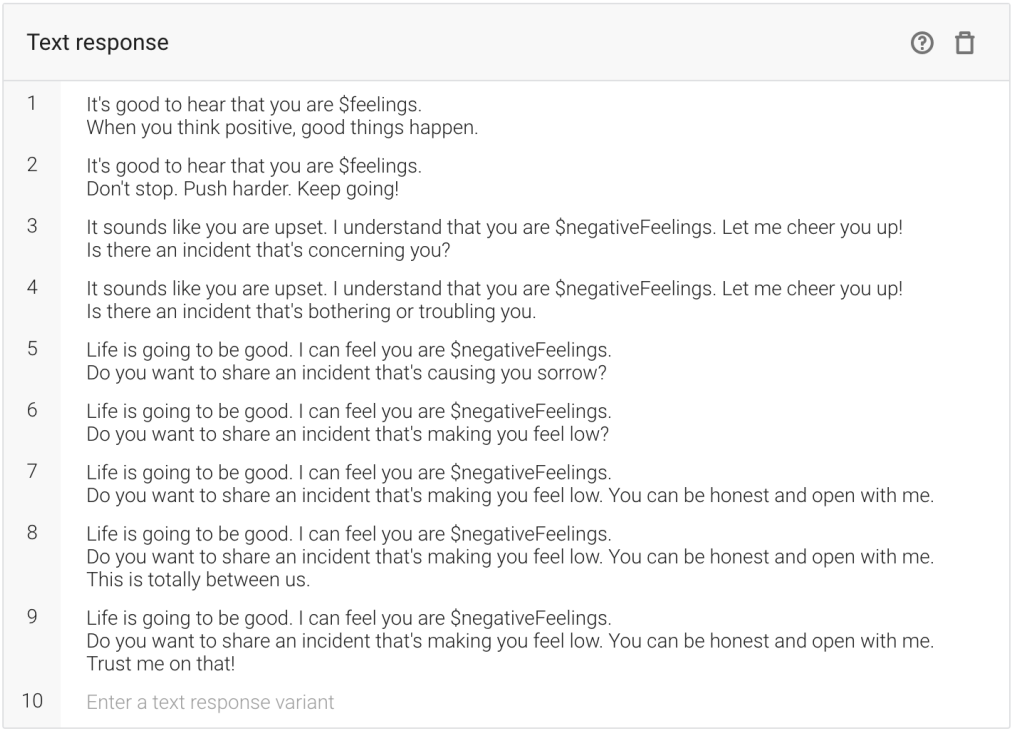


Figure-1.8

## 5.3 Entities

Entities are powerful tools used for extracting parameter values from natural language inputs. Any important data we want to get from a user's request, will have a corresponding entity. There are three types of entities: system (defined by Dialogflow. Example: given names, address, phone numbers, color, temperature etc.), developer (defined by a developer), and user (built for each individual end-user). Figure-1.9 shows the entities for "activities". When the bot wants to learn about the activities that the user likes to do, there are innumerable possibilities and we have defined a few as shown in the figure. But, Dialogflow is intelligent enough to understand all the synonyms and continuously learns more synonyms.

| | |
|---|---|
| reading | reading books, reading novels, reading comics, reading newspaper |
| playing | playing, playing basketball, playing volleyball, tennis, cricket, badminton, ping-pong, foosball, UNO, cards |
| outdoor | outdoor, skiiing, sky-diving, scuba-diving, ice-skating, kayaking, rafting, swimming, travelling, hiking, cycling |
| art | art, singing, dancing, listening to music, playing violin, music, playing guitar, playing keyboard, painting, cooking |
| entertainment | entertainment, watching movies, watching tv series, watching tv, video games, mobile games, browsing |
| technical | technical, programming, coding, solving puzzles, solving math problems |
| miscellanous | miscellanous, playing with kids, trying new things |
| pets | pets, dogs, cats, fish, birds, hamsters, puppies, kittens |
| | Click here to edit entry |

Figure-1.9

Figure-2.0 shows the “activities” entity usage in the conversation. Although, there is no such activity as “I like playing computer games” in our entity list, Dialogflow recognizes that “playing computer games” is an activity that a person would like to do.

## 5.4 Contexts

Contexts represent the current context of a user’s request. This is helpful for differentiating phrases which may be vague or have different meanings depending on the user’s preferences, location or the topic of conversation. Contexts are designed for passing on information from previous conversations and can be used to manage the conversation flow. Each intent can have several input and output contexts. The output contexts from an intent will act as an input context to another intent. Figure-2.1 shows an example of the context for “5.0 – askAppetite” intent. Based on the user’s response, the decision tree can branch in different ways and go to respective intents based on the output context.

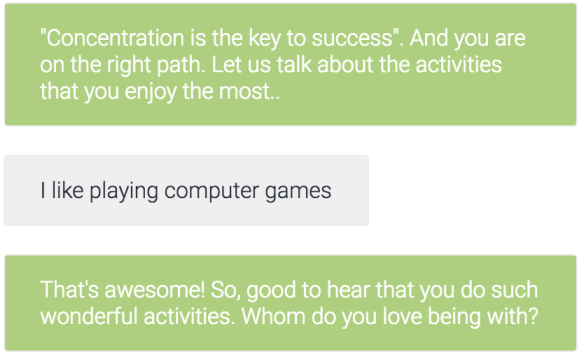


Figure-2.0

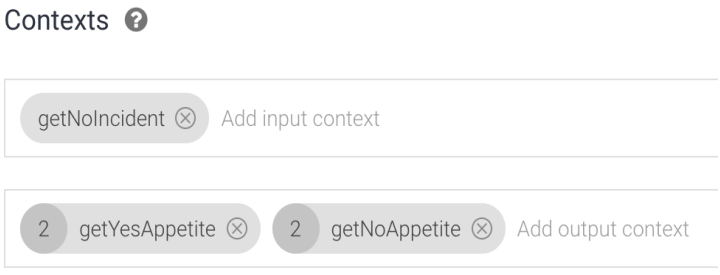


Figure-2.1

## 5.5 Decision tree of Achelois

Agents work based on decision trees and at each step of the conversation flow, it can have multiple paths to proceed and determines the best path based on the previous outcome and input. Figure-2.2 shows a small part of the decisions taken by Achelois. This figure covers a portion of the illness such as weight loss, excessive sleep and interested activities.

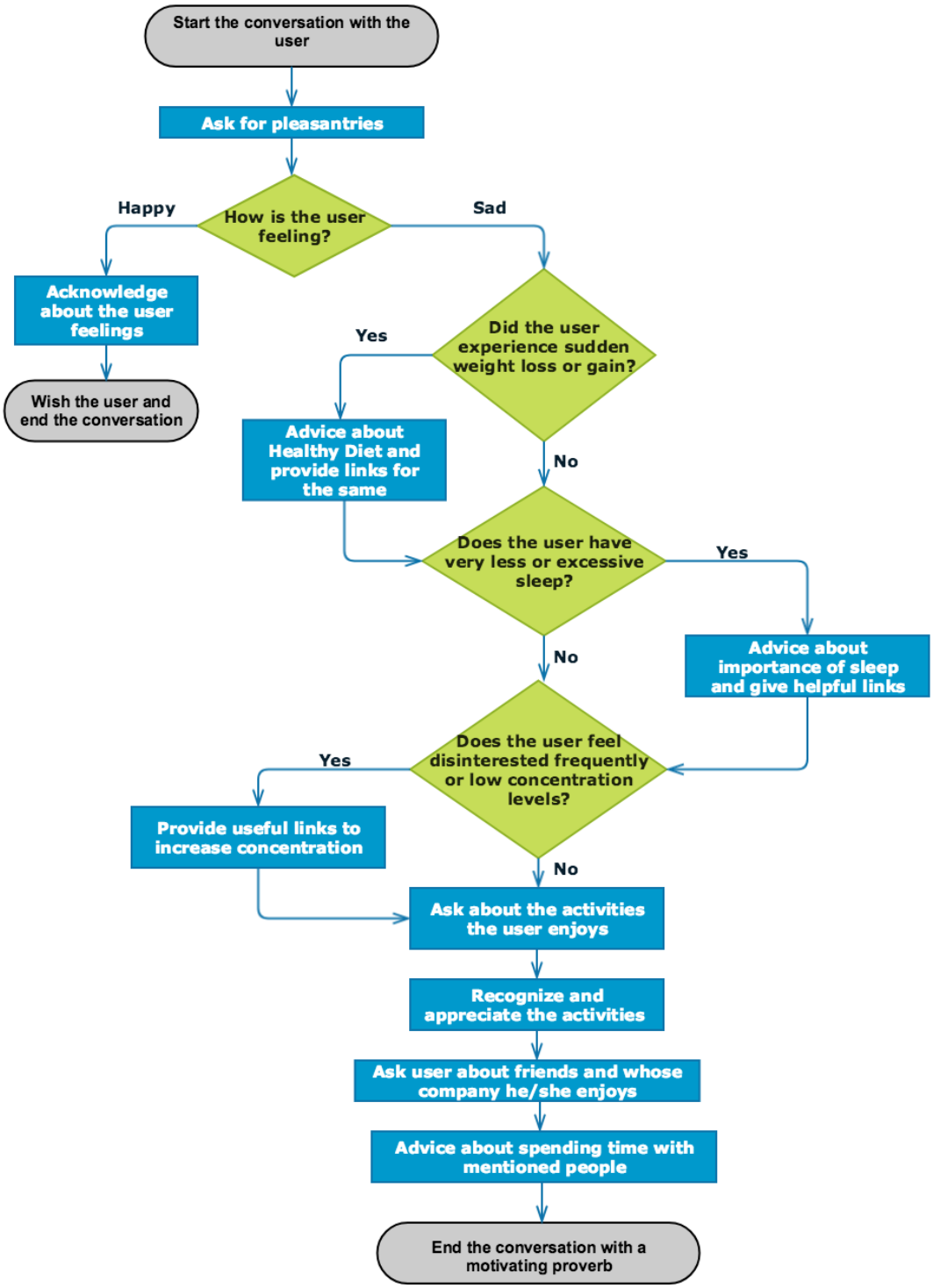


Figure-2.2

# 6. A Simple Yet Ingenious Web Application

We couldn't stop at just building a chatbot and learning the techniques behind it. We were curious to explore other applications in NLP and HCI. So, we developed a web application that could be deployed to the cloud by just a voice command. The web application is a simple intuitive single page website that has Achelois embedded in it.

## 6.1 How it works

1. Just say "Deploy NLP app" to Google Assistant on the phone (Google Assistant can figure out similar sentences as well!).
2. IFTTT has plenty of integrations and we have integrated Google Assistant and Slack to IFTTT. When there is a deploy command to Google Assistant, IFTTT intervenes and posts a message on Slack. This can be any channel or the Slack bot.
3. We have two very simple Django applications developed. The first application just listens to a channel or the Slack bot in real-time.
4. When there is a deploy post on Slack, our application picks it up and performs an empty commit of the other Django application to our GitHub repository.
5. The second Django application hosted at **https://nlp-spring18-ui.herokuapp.com/**, is the web application that has Achelois in it and few other things. This web application is deployed on Heroku.
6. Our Heroku app is configured in such a way that, whenever there is a push to our GitHub repository, Heroku initiates an automatic deployment.

Figure-2.3 shows the above steps in the pictorial representation.

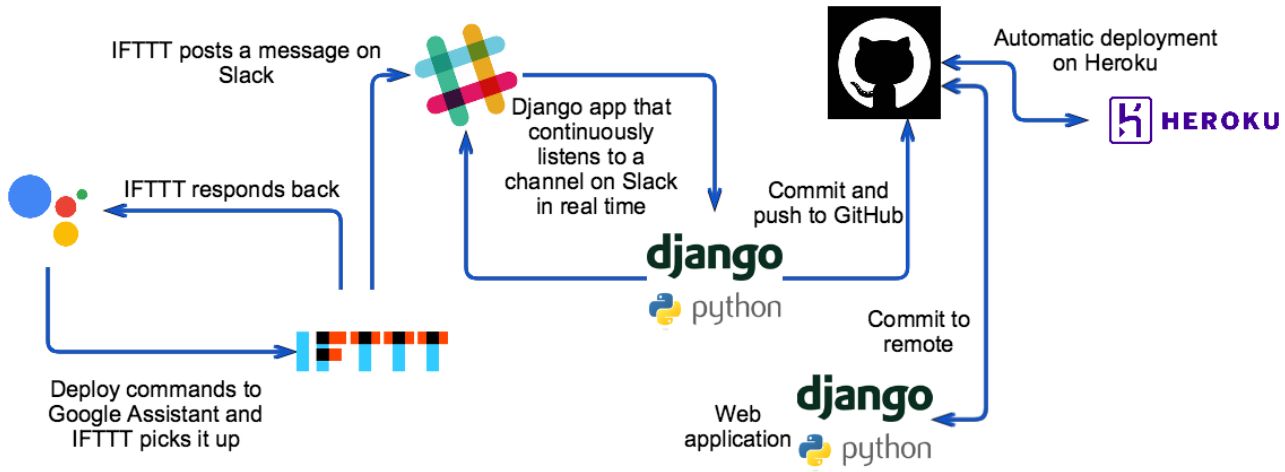


Figure-2.3

# 7. Conclusion and Future Work

Chatbots in healthcare domain is very young and the scope for innovation is broad. A simple question like "How is your sleep cycle?" can have a lot of different responses from a user in contrast to questions about weather or flight information. We wanted to build the bot in small pieces to cover a very tiny set of mental illness. Using this idea, there is a lot of scope to improve Achelois and take it to the next level. One of the

challenges with respect to healthcare domain in general is that the data is sparse. Even if large datasets are available, it is difficult to feed them to the bot as the responses from patients are unpredictable. Just a decade ago, chatbots and virtual assistants were hardly used. Now everyone has an assistant built into their phone which make our lives easier. We certainly believe that chatbots have a lot of potential and the research trends and the creativity especially in healthcare is going to be a game changer.

# 8. References

(n.d.). Retrieved from https://dialogflow.com/

Abdul-Kader, S. A. (n.d.).

*Build natural and rich conversational experiences*. (n.d.). Retrieved from Google DialogFlow: https://dialogflow.com/

Javier. (2015). *Building a Chatbot: analysis & limitations of modern platforms*. Retrieved from tryo labs: https://tryolabs.com/blog/2017/01/25/building-a-chatbot-analysis--limitations-of-modern-platforms/

Sameera A. Abdul-Kader, D. J. (2015). *Survey on Chatbot Design Techniques in Speech Conversation Systems.* International Journal of Advanced Computer Science and Applications.

Woudenberg, A. v. (2014). *Chatbots and Dialogue Systems: A Hybrid Approach.* Open University of the Netherlands.

https://docs.microsoft.com/en-us/azure/bot-service/dotnet/bot-builder-dotnet-manage-conversation-flow

https://blog.algorithmia.com/introduction-natural-language-processing-nlp/

https://wit.ai/

https://www.luis.ai/home

http://medicalfuturist.com/chatbots-health-assistants/

https://www.techemergence.com/chatbots-for-healthcare-comparison/

# 9. Source Code, Website and Video

The source code of the web application is available on GitHub:
https://github.com/Akshay23504/nlp-spring18-ui

The website URL is:
https://nlp-spring18-ui.herokuapp.com/

The following video demonstrates Achelois in action:
https://www.dropbox.com/s/rc7efr0r1ovukwo/Achelois_video_demo.mov?dl=0