A Chatbot Framework For Yioop

By
Harika Nukala

Advisor
Dr. Chris Pollett

Committee Members
Dr. Robert Chun
Dr. Leonard Wesley
Agenda

- Problem statement
- Introduction
- Background
- Design and Architecture
- Chatbot API
- Language Understanding API
- Integration
- Experiments
- Conclusion and Future Work
- References
The goal of the project is to design and develop a chatbot framework which provides a facility for developers to create chatbots in Yioop.

- Yioop is an open source, PHP search engine and discussion board portal.

- Simply explained Chatbot Framework is where bots are built, and where their behavior is defined.

- Bot development frameworks are software frameworks that abstract away much of the manual work that is involved in building chatbots.
Introduction

• A chatbot is an artificially intelligent computer program that can converse with humans on messaging platforms.

• An artificial intelligent chatbot can understand user’s intention and respond accordingly.

• A developer must have expertise and special knowledge in few areas like machine learning and artificial intelligence to create such chatbots.

• To make it easier, companies like Facebook and Microsoft introduced bot frameworks for developers to build and connect bots to users in a conversational interface/channels.

• The aim of my project is to create a Chatbot Framework for Yioop, with a similar kind of end goal in mind.
Key Concepts

• **Expression**
  An Expression is the textual input from the user, that a bot needs to interpret. It is a sentence, like “Book me a ticket to Paris” or “Get me weather of San Jose”.

• **Intent**
  An intent represents actions the user wants to perform. It is a purpose or goal expressed in a user's input, such as book a flight, get weather update, or reserve a hotel room.

• **Entity**
  An entity is relevant to a user’s intent. In the expression “Book me a ticket to Paris”, “Paris” is an entity of type location.
• Facebook and Microsoft have large-scale bot frameworks designed and developed to produce a mass number of chatbots.

• All developers face the same kind of problems while creating bots. They are: bots require basic input and output; they must have language understanding skills; and they must be able to connect with the user in a conversational interface.

• These frameworks provide tools that help developers to build bots and define their behavior using a programming language.
• Facebook Messenger platform provides Receive and Send API that allows developers to make bots interact with businesses.

• The Facebook Bot Engine (Wit.ai) depends on concept of Machine Learning.

• User feed the Bot engine with sample conversations so that it can handle many different variations of the same questions.

• Wit.ai parses a message into structured data (Understand) or predict the next action your bot should perform (Converse).
Preliminary Work Summary (Contd.)

6. Response
“The weather is 49 degrees and cloudy in Seattle”

1. User input
“what is weather in Seattle”

2. Forward input to process

3. Intent: “getWeather”
Entities: “Seattle”

4. REST

5. Results
49, cloudy

Yahoo Weather API

Chatbot Web service

NLP service
Wit.ai

Conversational Channel
Facebook Messenger

Bot Routing service
• Microsoft announced its Bot Framework roughly at the same time as Facebook.

• Microsoft provides an SDK that can be viewed as two components:
  1. Bot Connector, the integration Framework
  2. LUIS.ai, the natural language understanding service component
Both wit.ai and luis.ai are external web services which are consumed as a backend service in chatbot applications using a REST call. This makes interactions between users and bots slower.

We created our own Language processing API in Yioop to make conversations faster between users and bots.
Architecture

User

Presentation Layer
(Yioop discussion group)

Service Layer
(Chatbot API + Language Understanding API)

Data Layer
(Bot Knowledge storage)

Bot web Service

Looks like Robot 😊
Sample Knowledge base

### Table 1: Expression

<table>
<thead>
<tr>
<th>EXPRESS_ID</th>
<th>EXPRESSION</th>
<th>INTENT_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>what is the weather in &amp;location</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>get me weather of germany</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>get me weather of Delhi</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>what is stock price of &amp;company</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>get me weather of seattle</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>get me stock price of apple</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table 2: Entity

<table>
<thead>
<tr>
<th>ENTITY_ID</th>
<th>ENTITY_NAME</th>
<th>USER_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>location</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>company</td>
<td>3</td>
</tr>
</tbody>
</table>

### Table 3: Intent

<table>
<thead>
<tr>
<th>INTENT_ID</th>
<th>INTENT</th>
<th>USER_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>getWeather</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>getStockPrice</td>
<td>3</td>
</tr>
</tbody>
</table>
Chatbot API

- The Chatbot API allows developers to create chatbots and configure them in Yioop.

- Users can interact with a chatbot from any group thread that a bot is configured to work for.

- This API acts as a connector and routes messages between users and chatbot.

- This API provide web based console so that developers add knowledge to specific bots.
The bot users have many of the same qualities as the normal users.

The main difference is the bot users are controlled programmatically by bot unique token and bot callback URL.
To verify the authenticity of the token sent by the chatbot API, the bot application has a procedure that extracts the token from the HTTP POST request, parses the token, verify its contents, and verify its signature.
An intent represents a task or action the user wants to perform.

Intents match user requests with the actions that should be taken by chatbot, so one must add intents to help chatbot understand user requests and respond to them.
Add an Entity

- Entities are the key information for a domain-specific chatbot.

- An entity is a collection of similar objects like a location, person’s name, number.
- Admin [Bot Story]

Edit an Entity / Add an Entity value

- Entity values are like instances of a class.

- For example, *location* is an entity and *San Jose, Chicago, Seattle* are the entity values of location entity.
Add an Expression

• Expressions are sentences representing examples of user queries that a chatbot is expected to receive and interpret.

• Expressions are labeled in terms of intents and entities.

• Entities are detected and labeled automatically if the chatbot has already learnt.
The language understanding API allows bots to understand the user query and extract meaningful information out of it.

It predicts what bot should do based on the user query.

In short, there are two things that the Language understanding API can possibly do —

- Intent Classification using TF-IDF algorithm
- Entity Extraction using pattern matching technique
Basic NLP Analysis

Execution Flow

- Tokenization
  (tokenize)
- Removing stopwords
  (stopwordsRemover)
- Stemming
  (stemTerms)
- Intent Classification
  (TF-IDF)
- Entity Extraction
  (patternMatching)

Language processing

Input: Raw text

Output: Intent and Entities
Classify Intent (TF-IDF)

- More frequent terms in a collection of sentences of an intent are more important, i.e. more indicative of the intent.

<table>
<thead>
<tr>
<th>Intent</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetWeather</td>
<td>What is weather in &amp;location</td>
</tr>
<tr>
<td>GetWeather</td>
<td>How is weather in &amp;location</td>
</tr>
<tr>
<td>GetWeather</td>
<td>Is it rainy in &amp;location</td>
</tr>
<tr>
<td>BookFlight</td>
<td>Book flight from &amp;source to &amp;destination</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intent</th>
<th>Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetWeather</td>
<td>weather</td>
<td>2</td>
</tr>
<tr>
<td>GetWeather</td>
<td>rainy</td>
<td>1</td>
</tr>
<tr>
<td>BookFlight</td>
<td>Book</td>
<td>1</td>
</tr>
<tr>
<td>BookFlight</td>
<td>flight</td>
<td>1</td>
</tr>
</tbody>
</table>
One measure of how important a word may be is its \textit{term frequency} (tf), how frequently a word occurs in a collection of sentences of an intent.

\[ tf_{t,S} = \text{frequency of term } t \text{ in a set of sentences } S \]

\[ tf_{t,S} = \sum_{s \in S} tf_{t,s} \]

\[ df_t = \text{intent frequency of term } t = \text{number of intents containing term } t \]

\[ idf_t = \text{inverse document frequency of term } t, \]

\[ = 1 + \log \left( \frac{N}{1 + df_t} \right) \]

\[(N: \text{total number of intents)}\]
A typical combined term importance indicator is $tf-idf$:

$$tf - idf_{t,S} = tf_{t,S} \times idf_t$$

A term occurring frequently in the intent but rarely in the rest of the collection has high weight.

The score of each intent is the sum of $tf-idf$ weight of each query term in set of sentences.

$$Score(q, S) = \sum_{t \in q} tf - idf_{t,S}$$

The top scored intent is returned as classified intent.
• The entities are key information to perform an action.

• These entities are extracted from user’s request by using matching keywords or the pattern matching technique (similar wording pattern from an existing database).

• For example, when a user asks, “How is the weather in San Jose”. The expression we have in knowledge base is generalizes as “what is weather in (.* )”.

• Pattern matching technique is applied on both sentences and extract “San Jose” as entity value.
Experiments – Testing and Use Cases

• After integrating the chatbot framework in the Yioop search engine, we performed some experiments to see how chatbot works and how they interact with users in Yioop discussion groups.

• To evaluate the chatbot framework, we need to create chatbot applications. I created a Weather Bot and Stock Bot services with unique bot token for authenticity.
Use Case – Weather Bot

• I created a Weather Bot application with unique bot token for authenticity.

• The Weather Bot service performs action that intend to give weather updates for user requested location.

• This bot service calls the yahoo weather service which is an external API to get weather information.

• The bot service URL and unique bot token must be provided as Bot token and Bot callback URL in weather bot account in order to have a conversation with users in Yioop discussion groups. These bots must be added to the Yioop groups the user chosen to interact.

• The bot has learnt two expressions with getWeather intent.
Use Case – Weather Bot (Contd.)

Account Details

Username: weatherbot
First Name: weather
Last Name: bot
Email: 
Bot User
Bot Unique Token: 1234
Bot Callback URL: http://localhost/WeatherBot
Password: ...

Intent | Entity | Expression

Intent
Expression
getWeather

Bot User has following Entities:
location

Expression | Name | Actions
Tell me weather in &location | getWeather | Edit | Delete
what is the weather in &location | getWeather | Edit | Delete
Use Case – Weather Bot (Contd.)

Conversation between a user (ramya) and a weather bot (weatherbot)

---

ramya

@weatherbot what is weather in seattle

---

weatherbot

The weather is 62 and partly cloudy in seattle.

---

ramya

@weatherbot what is weather in San Jose

---

weatherbot

The weather is 66 and clear in san jose.
The Entity knowledge base is updated with new entity values ‘Seattle’ and ‘San Jose’
Use Case – Stock Bot

• Similar to the Weather Bot, we created a Stock Bot application with unique bot token for authenticity and respective bot account.

• The Stock Bot service calls the yahoo finance, an external API to get the current stock price of the user asked stock symbol.
Use Case – Stock Bot (Contd.)

Account Details

Username: stockbot
First Name: stocksbot
Last Name: bot
Email: 
Bot User
Bot Unique Token: 3456
Bot Callback URL: /lhost/StockBot.php
Password: ...

Intent | Entity | Expression

Intent
Expression
Bot User has following Entities:
company

<table>
<thead>
<tr>
<th>Expression</th>
<th>Name</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>get me stock price of &amp;company</td>
<td>getStockPrice</td>
<td>Edit, Delete</td>
</tr>
<tr>
<td>what is stock price of google</td>
<td>getStockPrice</td>
<td>Edit, Delete</td>
</tr>
</tbody>
</table>
Conversation between a user (ramya) and a stock bot user (stockbot)

---

ramya

@stockbot can i get stock price of google

---

stockbot

The current stock price of google ("GOOG") is $941.99 and percentage change is +0.46%.

---

ramya

@stockbot get me stock price of amazon

---

stockbot

The current stock price of amazon ("AMZN") is $967.55 and percentage change is +0.88%.
Conclusion and Future Work

• In this project, a new module called Chatbot Development Framework is developed for Yioop.

• This bot framework is useful for developers to create new chatbots, train them quickly in the Yioop social site. Yioop bots are more like human when it interacts with users and makes conversational experience more enjoyable.

• I am excited to provide a preliminary availability of the chatbot Framework for Yioop. This framework can be extended and improved by adding tools or facilities like conversation state management, scheduled conversations, language specific bots, a bot search and bot recommendations.
References


