

Detailed Study on Natural Language Processing Services.

Kunal Shah¹, Flavia Gonsalves²

¹Student, Mumbai Educational Trust, India

²Professor, Flavia Gonsalves, Mumbai Educational Trust, India.

Abstract – One key problem between human and computer interaction is the potential of computer to understand what a human wants. The AI assistant should be smart enough to understand the human behavior and react accordingly. There are various companies who have started integrating their product with Natural language capabilities to reduce redundant efforts. In this paper, we will perform a detailed and comparative study on various Natural language processing services.

Key Words: Natural Language Processing, Api.ai, Wit.ai, LUIS.

1. INTRODUCTION

Natural Language Processing (NLP) is a branch of Computer Science which has the ability of Computer Programming to understand natural language spoken by humans as is. NLPs are used to automate the tasks performed by humans. It can be used in various sectors like Education, Health Care, Multi-National Companies, etc. Human speech is however not precise and is often opaque where the structure can depend on various attributes like slang, regional jargon and complex combinations.

There are various Natural Language Processing services which can be applied to obtain necessary results.

A common feature of NLP systems is that they convert text input into formal representation of meaning such as logic (first order predicate calculus), semantic networks, conceptual dependency diagrams, or frame-based representations. [1]

Natural Language is sub divided in 2 types (i) Constrained Vocabulary and Grammar (ii) Unconstrained.

Constrained Vocabulary and Grammar (CG) is a methodological paradigm for Natural Language Parsing (NLP). Linguist-written, context dependent rules are compiled into a grammar that assigns grammatical tags ("readings") to words or other tokens in running text. Typical tags address lemmatization (lexeme or base form), inflexion, derivation, syntactic function, dependency, valiancy, case roles, semantic type etc. Each rule either adds, removes, selects or replaces a tag or a set of grammatical tags in a given sentence context. [2] These conditions can be tagged to any word in the sentence, either locally or even on a global level.

Unconstrained speech is when keywords are not separated from other words and grammar is not enforced on sentence. Some algorithms used for this task are:

- Sliding window and garbage model
- K-best hypothesis
- Iterative Viterbi decoding

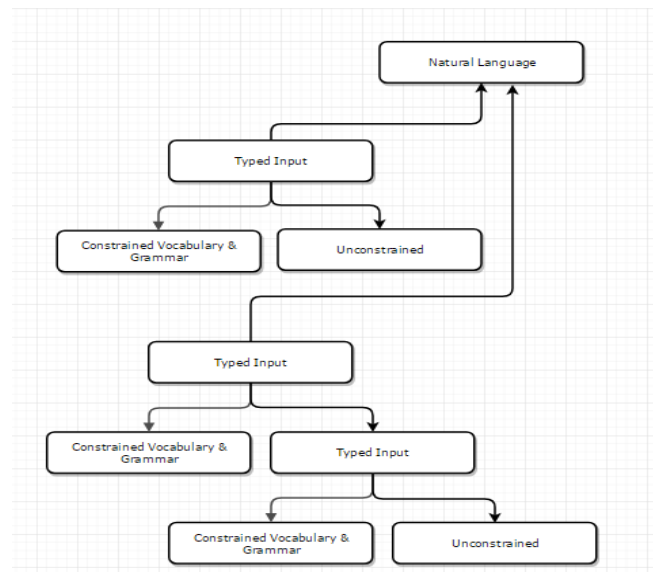


Fig -1: Types of Natural Language

In today's era there are various Natural Language services which are developed on top of complex algorithms. These make life of the developers easier and prevent the invention of wheel again. Some of these services are:

1. LUIS
2. Api.ai
3. Wit.ai
4. Lex
5. Watson

2. NATURAL LANGUAGE TERMINOLOGIES

There are some decisive key concepts which are common amongst major Natural Language Services mentioned below:

1. Utterance

An utterance is an input from user in textual or raw format that your algorithm needs to interpret. It can be a in form of a sentence or just keywords. For e.g.: If the user types "Order me a Pizza" or "Order me a large extravaganza", the algorithm needs to be smart enough to understand the utterance. Utterance cannot be predicted as every human has its own natural language of putting questions.

2. Intents

Intents are verbs in a sentence which represents the action that needs to be performed by the user. It is the task or need of that user needs to perform. Such as, if the utterance is "Order me a large pizza", then the corresponding intent can be "Order Pizza".

3. Entities

Entities are nouns present in an utterance. It represents the object on which action will be performed. An utterance can have multiple entities on which multiple operations need to be performed. For e.g.: "Order me a large extravaganza pizza" OR "Order me a large pepperoni pizza." In this case the type of pizza, "extravaganza" and "pepperoni" are mapped as entities. Examples of entities supported by NLPs are: Location, Number, Date, Distance, etc.

3. NATURAL LANGUAGE SERVICES

1) LUIS

LUIS refers to Language Understanding Intelligence Service. It is a powerful AI under the family of Microsoft's Cognitive Services. LUIS can process Natural Language by leveraging pre-designed models constructed by Microsoft and also has the capability to design user-specific models.

LUIS was launched in beta version and free of cost to use. It was introduced with Microsoft Bot Framework which acts as a middleware framework to deploy bots on various platforms like Skype, Slack, Facebook Messenger, etc.

It has a limitation on the number of transactions and also up to 5 direct requests per second. It uses machine learning algorithms to analyze the utterances. It breaks up the utterances into 'tokens' succinctly. LUIS provides the functionality of pre-built entities like intent.alarm, intent.calendar, intent.currency, intent.email, etc.

LUIS also provides the convenience to integrate with Microsoft's Azure Service effortlessly and also with other supported servers. LUIS allows creating the model using its web interface, and doesn't require any coding knowledge to build one.

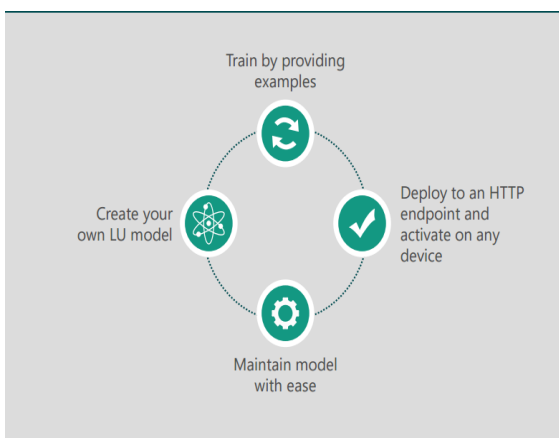


Fig -2: Lifecycle of LUIS[7]

2) Api.ai

Api.ai formerly known as 'SpeakToIt' is a Natural Language Processing service provided by Google Inc. Api.ai provides Speech to text as well as Text to Speech capabilities.

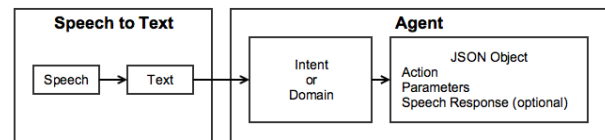


Fig -3: Architectural flow for Speech and Text[3]

It has the supports for Intents, Entities, Actions along with a focus area for 'Domains'.

Api.ai has a vast list of Pre-built entities like @sys.date, @sys.color, @sys.currency, @sys.units... etc. This pre-built entities can interpret the human behavior and responds back to user in a humorous tongue. Currently, prebuilt agents are available in English, Spanish, French, Japanese, Italian, Portuguese, Brazilian Portuguese, German, Russian, Chinese (Simplified), Chinese (Cantonese), and Chinese (Traditional). Api.ai also provides an integration module for 3rd party applications directly to Facebook messenger, Slack, Skype ... etc. Deploying the Api.ai applications is possible on Google's enterprise paid environment and can also be deployed on Heroku app or any other similar servers.

Furthermore, it also has the feature to maintain 'Contexts'.

Contexts are strings that justify the respective context of a user's request. It is helpful for differentiating phrases which may be vague or have different meanings depending on the user's preferences or geographic location, the current page in an app, or the topic of conversation.[3]

For example, if a user is listening a song of a specific genre. If the user has a music player application integrated with Api.ai then he may be interested in listening similar tracks of specific genre. The Api.ai in this case would be smart enough to understand the affinity of the user and may throw tracks to the user on similar grounds.

Or let's say you're a manufacturer of smart home devices, and you try to design an application that controls house hold devices. In this case, you may need help of embedded systems to connect your device with Arduino or Raspberry Pi Micro Controllers to talk with hardware. If the user says "Turn off the lights", the application will recognize the language and follow the command. If the user says "Turn it on", Api.ai would have maintained the context of "Lights" and will turn out the lights [3].

Contexts in Api.ai are tied with user-sessions. User can also manipulate with the life-span of a particular context. To refer to a parameter value that has been extracted in intent with a defined output context, use the following format in the 'VALUE' column: #context_name.parameter_name.

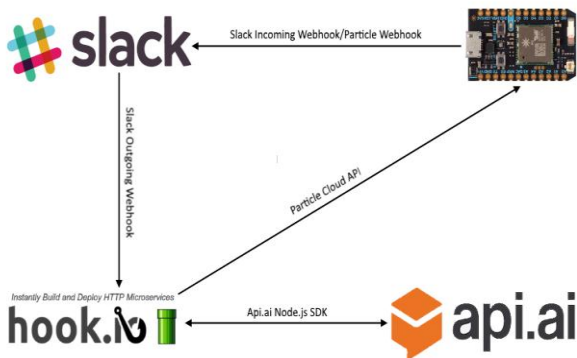


Fig -4: Api.ai integration with Slack and Arduino[3]

User also has the flexibility to use the Match mode. There are 2 types of match mode provided by Api.ai

- Hybrid (Rule Based and Machine Learning).
- Machine Learning only.

3) Wit.ai

Wit.ai is a NLP service provided by Facebook. It can help you parse a message into structured data (Understand) or predict the next action your bot should perform (Converse). The recipes are designed to help you with both the Understand and Converse capabilities by looking at common problems and recipes to address them [4]. Wit.ai supports more than 50 languages which include English, Polish, Russian, etc. Wit.ai is completely free of cost without any limitations on requests, intents or entities.

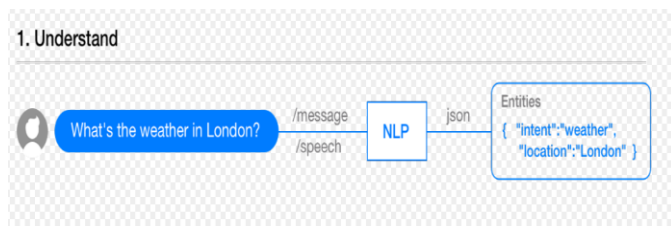


Fig -5: Simple conversational flow for Wit.ai[4]

Converse allows the user to build conversational app. It allows you to predict the response of the app at given state in the context.

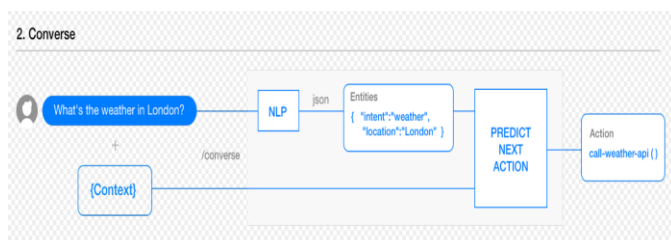


Fig -6: Contextual flow for Wit.ai[4]

Wit.ai also provides the combination of text and speech recognition. Just like Api.ai it also has its own Machine learning algorithm which will constantly revise itself with changes in model. The key elements of Wit.ai are similar to other NLP services except actions here are separated to use combined operations. It provides the functionality of pre-built entities like currency, temperature, numbers, etc. Unlike LUIS and Api.ai it cannot be integrated succinctly with 3rd party messaging software's like Facebook messenger or skype. The user can create a web service Api and then hook up with application to leverage the services. There are also server less messenger boilerplate services which can be used to create gateway between Wit.ai and other Social networks.

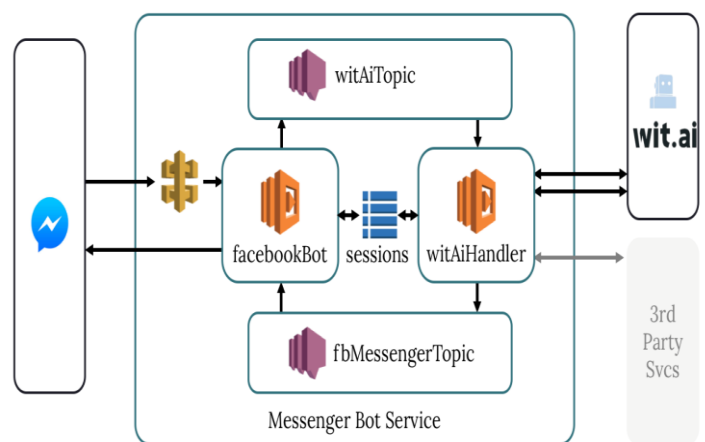


Fig -7: Wit.ai Boilerplate Architecture[5]

Wit.ai provides SDK's for iOS, Android, Node.js, Raspberry PI, Python, C, etc. It also has its JavaScript plugin for front end developers.

Wit.ai also allows the user to build a flow-based bot, where the bot runs a 2 question survey. Based on the answer of 1st question, the bot will append additional questions to the conversations.

Bot: "Do you watch sports on TV?"
 User: "No"
 Bot: "Ok. And do you watch sports online?"
 User: "Yes"

But if the user answers "Yes" the bot must ask another question:

Bot: "Do you watch sports on TV?"
 User: "Yes"
 Bot: "Which sport do you watch the most?"
 User: "Soccer"
 Bot: "Ok. And do you watch sports online?"
 User: "Yes" [4]

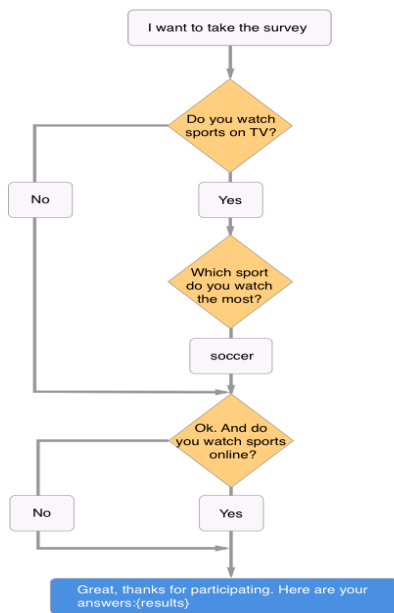


Fig -8: Flow-based approach for Wit.ai[4]

4. BENCHMARKING THE NATURAL LANGUAGE UNDERSTANDING SERVICES

In this section we discuss the benchmark conducted across multiple Natural Language Understanding Services. Beyond performance, the ability to operate offline can be a game-changer for use cases like connected cars, where internet connectivity cannot be guaranteed.[6] Privacy is a concern for users. Thereby, sending data to cloud might not always be an option. Privacy can be critical in some prone sectors like banking, healthcare, etc. In the graph below, the dots represent the aggregated precision and also recall scores for all slots of specified intents.

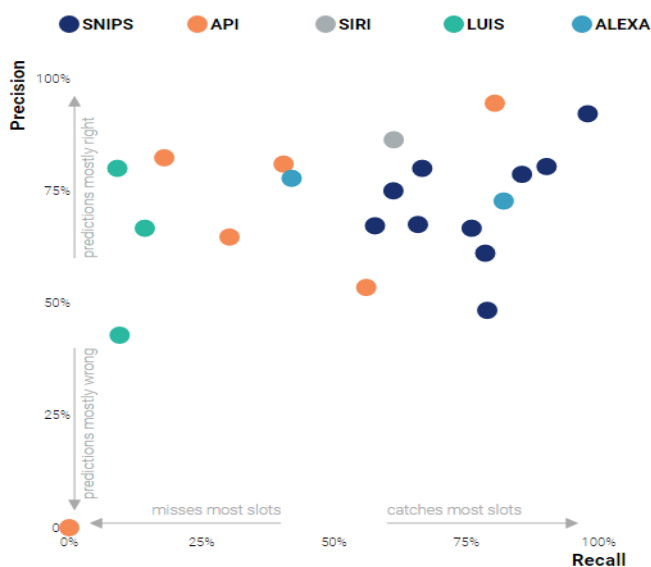


Fig -9: Aggregated performance of NLPs[6]

5. CONCLUSION

Our personal preference goes to Api.ai from Google as they provide all the necessary elements for building conversations as they have the feature of fallback entity which handles the conversation very well. They also have free tier, so you don't have to worry about the price. Anyway, we would recommend you store all data needed for your model in a structured way in your own code repository. It means that you can retrain the model when you want and even switch the providers when you wish. You wouldn't want yourself to be in a situation where company faces downfall and you will need to start training from scratch. For the end-to-end solutions that require less code, we think Api.ai is the way to go. This is also a good option if you need embedded capabilities, avoiding dependence on an internet connection.

6. ACKNOWLEDGEMENT

The acknowledgement is a small effort to express my gratitude to all those who have assisted us during the course of preparing this paper. I am greatly indebted to express immense pleasure and sense of gratitude towards my guide and mentor Prof. Flavia Gonsalves for her constant support and valuable encouragement.

REFERENCES

- [1] H. Feili. Natural Language Processing Projects. [PowerPoint] Sharif UT. Tehran. Iran, 2003M. Young, The Technical Writer's Handbook.
- [2] World of VISL - Constraint Grammar (http://visl.sdu.dk/constraint_grammar.html)
- [3] Contexts - Api.ai (<https://docs.api.ai/docs/concept-contexts>)
- [4] Recipes for Conversational apps - Wit.ai (<https://wit.ai/docs/recipes#overview-link>)
- [5] Serverless messenger bot boilerplate (<https://github.com/SC5/serverless-messenger-boilerplate>)
- [6] Benchmarking Natural Language Understanding Systems. (<https://snips.ai/content/sdk-benchmark-visualisation/>)
- [7] Language Understanding Intelligent Service - LUIS (<https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/Home>)