Earlham COLLEGE

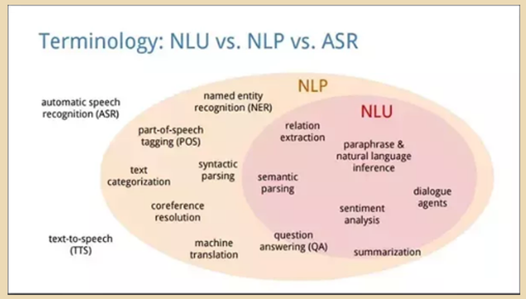

Figure 1: Visualization of the scope of NLP.[1]

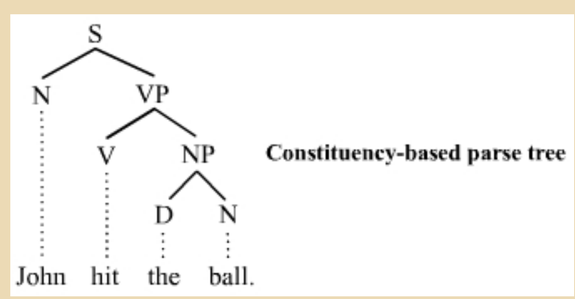

Figure 3: An example of a Parse Tree for the sentence *"John hit the ball"*.[2]

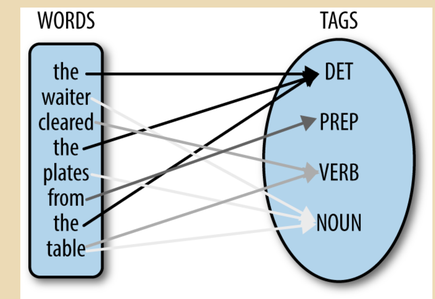

Figure 4: An example of Part-Of-Speech Tagging.[3]

A dialogue-driven ChatBot based on a NLIDB System

Computer Science Senior Capstone Project | Jon Abdulloev

SUMMARY

- Many resources and opportunities in a community go unnoticed and are not availed of by a majority the community.
- If the providers and the consumers of resources are more efficiently connected, then a more sustainable use of a community's resources will be achieved.
- A relational database would make an effective system to keep track of available resources.
- However, not every member of a community would have the correct SQL (Structured Query Language) background to communicate with a relational database.
- The proposed system in this presentation is a Natural Language Interface to Database (NLIDB) system whose purpose is to convert natural language (English) into SQL while interacting with users in the most user-friendly way possible.
- If any member of the community is easily able to access the database then the expected outcome is an equitable distribution of resources which would make for a more sustainable use of a community's resources.

NATURAL LANGUAGE PROCESSING

There has been a significant amount of research on Natural Language Processing (NLP) which has been largely motivated by its enormous applications. Some of the well-known systems that use NLP techniques include

- Siri from Apple
- IBM Watson
- Wolfram | Alpha.

NLP is a broad topic with various subtopics as shown in Figure 1.

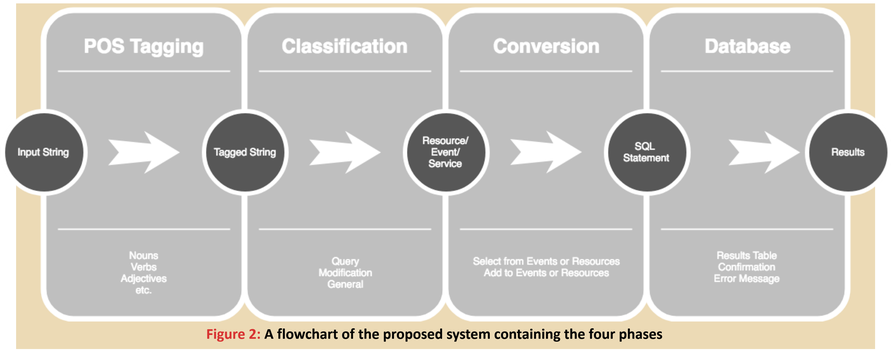

Figure 2: A flowchart of the proposed system containing the four phases

METHOD AND PROCESS

There are four types of sentences in the English language:

1) Declarative (e.g. *"I have a car"* or *"I can help with Statistics")*
2) Imperative (e.g. *"Are there rides?"* or *"Can people help me?")*
3) Interrogative
4) Exclamatory

We focused on parsing simple declarative and interrogative sentences that belong to contexts relating to the querying and adding of information about services (such as tutoring and rides), sharable items (such as books, cars, pens) and events. The program consists of 4 phases:

1. First phase tags each word of the input with its part-of-speech.
2. Second phase uses the tagged sentence from the first phase to classify the sentence as either a database query, a database modification, or a general sentence.
3. Third phase then converts the input into a SQL command.
4. Final phase involves returning the results of running the generated SQL command. If the SQL command is a query (on either the *Resources* or the *Events* tables) and there are entries in the database, then a table of the results is returned. However, if there are no entries in the database for that specific query, then a follow up message is returned. If the SQL command is a modification (to either the *Resources* or *Events* tables) then a simple confirmation is returned.

RESULTS

- The challenges of natural language ambiguities (including Syntactic and Word Sense Ambiguities) are resolved in the second phase of the program, classification.
- This is a crucial phase of the system which largely determines if the correct SQL statement is generated from the natural language sentence inputted.
- A test dataset was produced based on a survey of how a group of users interacted with the system during user testing.
- The demographics of the test user group was a multicultural college student body of ages 18-24.
- An iOS mobile application connected to a Python3 Flask server was used as the user interface to test the proposed system (*see Acknowledgements*).
- Based on the results, the recall, precision and F1 Scores of each of the dialogue trees were calculated.

Dialogue Tree	Precision	Recall	F-Measure
Tutorial	1.00	0.83	0.91
QueryResource	0.92	0.91	0.91
QueryEvent	1.00	0.85	0.92
ModifyResource	1.00	0.89	0.94
ModifyEvent	1.00	0.85	0.92
AddEvent	0.83	0.82	0.82
Average	**0.96**	**0.86**	**0.90**

Table 1: The Precision, Recall and F-Measure of each dialogue tree as well as the averages of those metrics.

CONCLUSION AND FUTURE WORK

The system performed significantly well. The average recall was 96% and the average precision was 86%. These two metrics contributed to a successful average F-measure of 90%. For future work, it would be interesting to

- use Machine Learning tools (e.g. Word2vec) to associate synonyms together, like *'car'* and *'ride'*.
- capture complex compound sentences by parsing the input and producing a Parse Tree (example in Figure 3).

REFERENCES:

[1]https://www.quora.com/What-is-the-difference-between-natural-language-processing-NLP-and-natural-language-understanding-NLU
[2]https://en.wikipedia.org/wiki/Parse_tree
[3]http://naviglinlp.blogspot.com/2017/04/lecture-7-part-of-speech-tagging.html

ACKNOWLEDGEMENTS: James Tran '18 built the iOS mobile application which combined with the Python3 Flask server built by Lam Nguyen '18 allowed for a user interface to test the proposed system. Furthermore, the rest of the NadaBot team assisted in the collection of user feedback (by arranging for the user testing of the proposed system through the application) which was vital for the improvement of the proposed system. Furthermore, my professor David Barbella guided me in my NLP research and I would like to thank him very much.