Abstract—This report describes my capstone project Smart Travel System which works as an interactive virtual travel assistant. It uses cognitive computing and natural language processing to provide a personalized travel experience. It also creates a detailed itinerary aligned to user preferences based on the number of vacation days, with information about the travel destinations including distance and driving time to the destinations from current location, full address of the destination, restaurants near the destination.

Index Terms—Travel; Cognitive Computing; Natural Language Processing, Data Management

I. INTRODUCTION

In today's fast paced life, it is important to make the best usage of available time. Even on vacations, visitors want to cover maximum attractions fitting to their individual travel tastes. My project Smart Travel System serves as multipurpose virtual travel assistant. It provides user with a personalized travel experience based on his/her tastes in traveling. It will suggest travel destinations based on what the user of the system wants to achieve from his vacation. The system takes user input in the form of text, speech or an interactive questionnaire and suggests travel attractions based on the obtained user preference. It also makes the task of planning out the vacation easy using detailed itinerary creation functionality depending on the number of vacation days of the user. It provides distance in miles from current location and the time required to drive to these destinations. Smart Travel system uses techniques of data management, cognitive computing, natural language processing, web service interaction and application development to give a smart and user-friendly experience.

II. BACKGROUND

The whole system is based on the concept of detecting user preference and suggesting travel destinations aligned to these preferences. User Preference denotes the type of travel experience an user desires for. This system considers seven types of user preferences. These preferences are called 'preference tag(s)'. The seven preference tags considered are:

- Adventure
- Art and Culture
- Family
- History
- Kid Friendly
- Nature
- Sports

For e.g. A user loves to hike and camp. This would denote the user has a strong Preference for the tag 'Adventure' and a weak preference to the tag 'Nature'. The reason Family and Kid Friendly tags are separate is because they might not always coincide. Places such as Parks are kid friendly but not necessarily best for a family trip. Places like halls and other such places where ceremonies or parties could be held would be tagged with the 'Family' preference tag. These places may or may not be Kid Friendly.

III. DATA COLLECTION AND PREPARATION

The first step of developing this travel system was to find a database consisting of travel attractions. After extensive research it was found that a database appropriate for use was not readily available, creation of one was the next step. Since creating a database for the whole world was impractical, focusing on areas in and around Rochester came out to be the best solution. After researching travel destinations near Rochester area using Google Search[1], a list was made manually, by me, consisting of 27 different tourist attractions providing a variety of travel experiences. The initial database consisted of travel destination names and a link to their web page[1]. These travel locations were then labeled according to the type of travel experience they provide. These labels are called 'preference tag(s)'. Further the database was normalized and divided into two different databases for ease of use. For preference analysis, each preference tag is related to a database consisting of synonyms and words related to those preference tags. These amount to seven different files, each for every preference tag. Every list contains 30-40 words. These words were collected from online websites[2] [3]. All these words are singular. Plurality is handled in the systems algorithm.

IV. MODES OF WORKING

The Smart Travel System has three different modes of working. This makes the system accessible to every type of usage mode.

A. Preference Profile

In this mode of working, the user is presented with a quiz consisting of ten questions, answers of which will indicate the user’s tastes in traveling. Here user is presented with multiple choice questions and he/she must select an option that he/she relates to the most as seen in figure 1. Each selected option contributes a few points to one or more of the preference tags. In this manner, a preference profile is created for the user. The quiz is constructed such that there will always be a strong preference denoted in the user’s answers. This means that there will never be a tie between any of the preference tags. The final strong preference is sent to the system for further processing.
Please enter mode - (P)reference Profile, (S)speech or (T)ext:

Input number of days you are visiting:
Please select the best alternative:

Q1. Do you like going For Hikes?
   1. Not interested
   2. Sounds good
   3. Love it!

Q2. Do you like forests and animals?
   1. Not interested
   2. Sounds good
   3. Love it!

Q3. How does admiring our rich historical past sound?
   1. Not interested
   2. Sounds good
   3. Love it!

Q4. Do you enjoy playing sports?
   1. Not interested
   2. Sounds good
   3. Love it!

Q5. Are you interested in paintings and art pieces?
   1. Not interested
   2. Sounds good
   3. Love it!

Q6. Which personality appeals the most to you?
   1. Dear Drillz
   2. Cristiano Ronaldo
   3. M F Hussain

Fig. 1. Preference Profile Creation Questionnaire

Please enter mode - (P)reference Profile, (S)speech or (T)ext:

Input number of days you are visiting:
Activating Speech mode...
Recording started. Please speak now!
finished recording
Speech to Text output received: animals and plants

Fig. 2. Speech Input Mode

B. Speech Input

In this mode of working, audio is recorded from the user in real time and saved to a wav file. This recorded audio file is sent to IBM Watson Speech to Text API[4]. As this audio is recorded and stored in a file, it is played back to the user to make the system user friendly. IBM Watson extracts text from this audio file and provides output in the form of text. This text is then used for interpreting the preference indicated by the user using the systems preference analysis algorithm. In figure 2, we can see an example of this mode where the audio is detected and displayed in textual format. Audio from the user is recorded for five seconds.

C. Text Input

This mode of working has been added for ease of use. User can input any length of text describing his likes and dislikes into the system. This input is further processed using preference analysis and user preference is detected. This mode will also ensure that the system works when the IBM speech to text server is not available or if there are issues with the user’s Internet connection. Also, during situations when the speech to text API is not able to detect user speech correctly, or if the user is operating from a noisy place, text mode will be able to provide accurate information.

V. Algorithm and System Flow

The application starts by first asking the user the type of mode he wants to use. After the user selects this information, the application asks the user for the number of days he/she is on vacation. This information will be used later on for making the itinerary. After this step, the user will either present a questionnaire, if the selected mode is Preference Profile Mode, or record audio, if the selected mode is Speech Mode, or ask for text input.

If Preference Mode is selected, the user is presented with the quiz. At the end of the quiz, the strong preference of the user is obtained by aggregating the points that the user’s answers have contributed to the seven different tags. InSpeech mode, the recorded audio is sent to IBM Watson using a https web service post call. Output is received from the web service in the form of json which is then decoded and textual output is extracted from. This output from IBM Watson will be now used as an input to the system’s NLP algorithm. This can be used to do speech synthesis in real time. Similarly, the text mode give us a text input.

The system uses a bag of words approach to decipher input text. The input sentence is divided into tokens. These tokens consist of single words. Punctuations are removed from the tokens. Now the data is ready to be processed. Every token is run through the seven different preference tag synonyms and related words list. As a match is found, a point is contributed to the related preference tag. As these tokens are parsed, points accrue for various preference tags. While parsing tokens which are in a plural format, they are converted to singular format by removing the ending letter ‘s’ and ‘es’ and parsed again to look for matches. At the end, the preference tag(s) with the maximum number of points is established to be the user’s travel preference. This preference is used to go through the database for travel attractions, and the places tagged with this preference are selected to be shown in the output.

The next step in this process is Itinerary creation. Using the number of vacation days, and the output received from
the NLP algorithm about the selected travel attractions, an itinerary is created. The travel attractions are equally divided into the number of days available. This is then displayed as figures 4 and 5.

Also, when the total number of travel attractions are less than the total number of vacation days, after displaying the attractions in the itinerary, a list of some more places to visit is shown. This list consists of one member of each preference tag so as to give a all-inclusive experience to the user(figure 6).

At the back end, IP address of the user is used to identify user location. This information is then used to call Google distance API[5] to calculate distance of the user from the travel location. This feature is handy as it will help the user plan his trip when he is on it. In case something closes down, he can refer to this information to decide if he wants to go to the next nearest possible attraction. I have also added driving time from current location to help the user by providing more information.

Google Geocode API[6] is useful in converting latitude longitude coordinates to user readable address. Also, while making the distance call, current address in words is required. User IP gives an address in the form of latitude and longitude. Google Nearby API[7] is used to get the list of places to eat near every travel location. This can again help the user plan out his trip. All Google APIs[6] [5] [7] need https post calls and outputs are received in json format. Both the IBM Watson and all of the Google APIs need a apikey for usage. This can be created by logging into the websites and obtaining API keys.

VI. ISSUES FACED

Unavailability of database is a major issue in scaling this system to a larger space. Database will have to be created manually. Once this is done, the current system can be scaled easily to handle big data. Another issue that I faced was the inability of IBM Watson to decipher accents. It could not understand what I was saying as opposed to when I used Americanized versions of the same words, where it performed very well. This was especially an issue when using one word audio inputs. It was able to understand sentences much better as it could make out the context.
VII. Ethical Issues

While making the synonyms and related words list for the preference tag 'Adventure', I came across a lot of words which indicated self harm or risk to life. Some of these words were 'hazardous', 'risky', 'gamble', 'jeopardize' etc. I removed such kind of words from the final list but it made me think that there should some sort of mechanism to detect such intentions of the users. Detecting these intentions could help us safe guard the user by either giving warnings or informing authorities.

VIII. Conclusion and Future Work

Smart Travel System acts as a multi-functional virtual travel assistant. The final output gives a personalized travel itinerary with distance and driving time from current location and places to eat near the travel attraction. It is easy to use and gives a lot of information to help the traveler make informed choices. In future, information about opening and closing times of the attractions could be added. Also, user reviews or rating of the place could be added. Integrating this system into a android or ios app would further make it easier to use.

ACKNOWLEDGMENT

I would like to thank Professor Carol Romanowski for guiding me through out the development of the project and Professor Reynold Bailey for helping me with my milestone and capstone presentations.

REFERENCES