1 Acknowledgement

I would like to express my gratitude towards my guide Prof. A.M.Bhadgale of Computer Engineering Department, who has been very concerned and have adided for all the help essentials for the preparation of this work. He has helped me to explore this vast topic in an organised manner and provided me with all the ideas on how to work towards a research oriented venture.

I am thankful to Prof.M.V.Marathe, Head of Department, Computer Engineering, for the motivation and inspiration that triggered me for the seminar work.

Khushbhoo Mundada T150074241

(T.E. Computer Engineering)

2 Abstract

One of the goals of Artificial intelligence (AI) is the realization of natural dialogue between humans and machines. In recent years, the dialogue systems, also known as interactive conversational systems are the fastest growing area in AI. Many companies have used the dialogue systems technology to establish various kinds of Virtual Personal Assistants(VPAs) based on their applications and areas, such as Microsofts Cortana, Apples Siri, Amazon Alexa, Google Assistant, and Facebooks M. However, in this proposal, we have used the multi-modal dialogue systems which process two or more combined user input modes, such as speech, image, video, touch, manual gestures, gaze, and head and body movement in order to design the Next- Generation of VPAs model. The new model of VPAs will be used to increase the interaction between humans and the machines by using different technologies, such as gesture recognition, image/video recognition, speech recognition, the vast dialogue and conversational knowledge base, and the general knowledge base. Moreover, the new VPAs system can be used in other different areas of applications, including education assistance, medical assistance, robotics and vehicles, disabilities systems, home automation, and security access control.

Contents

3 Introduction

Spoken dialogue systems are intelligent agents that are able to help users finish tasks more efficiently via spoken interactions. Also, spoken dialogue systems are being incorporated into various devices such as smart-phones, smart TVs, in car navigating system. Also, Dialogue systems or conversational systems can support a wide range of applications in business enterprises, education, government, healthcare, and entertainment. Personal assistants, known by various names such as virtual personal assistants, intelligent personal assistants, digital personal assistants, mobile assistants, or voice assistants. Many companies have used the spoken dialogue systems to design their dialogue system device, such as Microsofts Cortana, Apples Siri, Amazon Alexa, Google Assistant, Samsung S Voice, Nuance Dragon, and Facebooks M. These companies used different approaches to design and improve their dialogue systems. There are many techniques used to design the VPAs, based on the application and its complexity. For example, Google has improved the Google Assistant by using the Deep Neural Networks (DNN) method which highlights the main components of dialogue systems and new deep learning architectures used for these components. Also, Microsoft used the Microsoft Azure Machine Learning Studio with other Azure components to improve the Cortana dialogue system.

In this proposal, we propose an approach that will be used to design the Next-Generation of Virtual Personal Assistants, increasing the interaction between users and the computers by using the Multi-modal dialogue system with techniques including the gesture recognition, image/video recognition, speech recognition, the vast dialogue and conversational knowledge base, and the general knowledge base. Moreover, our approach will be used in different tasks including education assistance, medical assistance, robotics and vehicles, disabilities systems, home automation, and security access control. To design the Next-Generation of Virtual Personal Assistants with high accuracy, we added some components to the original structure of general dialogue systems to change the general model to Multi-modal dialogue systems, such as ASR Model, Gesture Model , Graph Model, Interaction Model, User Model, Input Model, Output Model, Inference Engine, Cloud Servers and Knowledge Base.

4 Virtual Personal Assistant

A virtual assistant or intelligent personal assistant is a software agent that can perform tasks or services for an individual. Sometimes the term "chatbot" is used to refer to virtual assistants generally or specifically those accessed by online chat (or in some cases online chat programs that are for entertainment and not useful purposes). Some virtual assistants are able to interpret human speech and respond via synthesized voices. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal commands.

As of 2017, the capabilities and usage of virtual assistants are expanding rapidly, with new products entering the market and a strong emphasis on voice user interfaces. Apple and Google have large installed bases of users on smartphones. Microsoft has a large installed base of Windows-based personal computers, smartphones and smart speakers. Alexa has a large install base for smart speakers.

Virtual assistants make work via:

1. Text (online chat), especially in an instant messaging app or other app
2. Voice, for example with Amazon Alexa on the Amazon Echo device, Siri on an iPhone, or Google Assistant on Google-enabled/Android mobile devices
3. By taking and/or uploading images, as in the case of Samsung Bixby on the Samsung Galaxy S8
4. Some virtual assistants are accessible via multiple methods, such as Google Assistant via chat on the Google Allo app and via voice on Google Home smart speakers.

Figure 1: The Structure of The Next-Generation of Virtual Personal Assistants

Virtual assistants use natural language processing (NLP) to match user text or voice input to executable commands. Many continually learn using artificial intelligence techniques including machine learning.

To activate a virtual assistant using the voice, a wake word might be used. This is a word or groups of words such as "Hey Mycroft", "Alexa", "Hey Siri" or "OK Google".

5 Dialogue System

A dialogue system is a computer program that communicates with a human user in a natural way. [1] The dialogue System provides an interface between the user and a computer-based application that permits interaction with the application in a relatively natural manner. The System can be CUI, GUI, VUI and multi model etc. it can be used in telephones, PDA systems, cars, robot systems and web browsers. A text based dialogue system is in which we chat with the system. A spoken dialogue systems is defined as a computer systems that human interact on a turn-by-turn basic and in which spoken natural language interface plays an important part in the communication. [2] A multimodal dialogue systems are those which are dialogue systems that process two or more combined user input modes - such as speech, pen, touch, manual gestures, gaze, and head and body movements - in a coordinated manner with multimedia system output. [3] Different Dialogue Systems have different architectures but they have same set of phases which are Input Recognition, Natural Language Understanding, Dialogue Management, Response Generation and Output Renderer.

Many companies have used the spoken dialogue systems to design their dialogue system device, such as Microsofts Cortana, Apples Siri, Amazon Alexa, Google Assistant, Samsung S Voice, Nuance Dragon, and Facebooks M. These companies used different approaches to design and improve their dialogue systems. There are many techniques used to design the VPAs, based on the application and its complexity. For example, Google has improved the Google Assistant by using the Deep Neural Networks (DNN) method which highlights the main components of dialogue systems and new deep learning architectures used for these components [16]. Also, Microsoft used the Microsoft Azure Machine Learning Studio with other Azure components to improve the Cortana dialogue system

5.0.1 Components

Components of Dialogue System A Dialogue system has mainly seven components. These components are following:

1. Input Decoder
2. Natural Language Understanding
3. Dialogue Manager
4. Domain Specific Component
5. Response Generator
6. Output Renderer

Input Decoder : component is the one which recognizes the input. It converts the input to the simple text. This component is present only in which are not text base dialogue systems. This component involves conversion of spoken sound (user utterances) to text (a string of words). This requires the knowledge of phonetics and phonology. Phonetics is branch of linguistic which deals with the sound of speech and their production, combination, description and representation by written symbols. Phonology is study of speech sound in language or a language with reference to their distribution and patterning and to tacit rules governing pronunciation. For this purpose speech Recognition is needed. There are many systems available for this purpose. These are called Automatic Speech Recognition (ASR), Computer Speech Recognition or simply Speech to Text (STT). Besides speech the dialogue system can have other inputs like gesture, handwriting etc.

Natural Language Understanding : As the name suggest this unit try to understand what user want to tell. It converts the sequence of words into a semantic representation that can be used by the dialogue manager. This component involves use of morphology, syntax and semantics. Morphology is

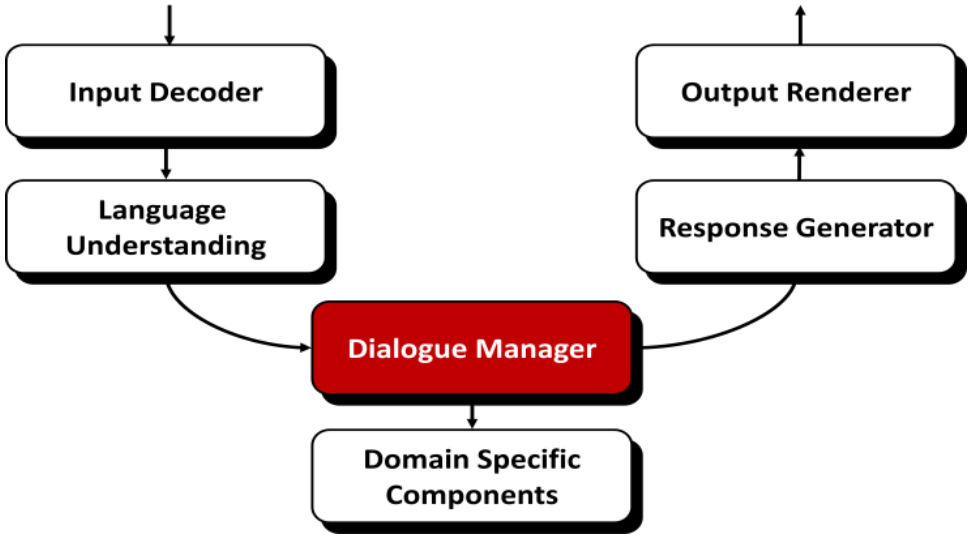

Figure 2: The Structure of The Next-Generation of Virtual Personal Assistants

the study of the structure and content of word forms. After identifying the keywords and forming a meaning it provide it to dialogue manager.

Dialogue Manager The Dialogue Manager manages all aspects of the dialogue. It takes a semantic representation of the users text, figures out how text fits in the overall context and creates a semantic representation of the system response. It performs many tasks these are:

1. Maintains the history of dialogue
2. Adopts certain dialogue strategies
3. Deal with malformed and unrecognized text

Retrieve the contents stored in files or database For these tasks dialogue manager has many components these components are:

1. Dialogue Model
2. User Model
3. Knowledge Base
4. Discourse Manager

Domain Specific Component : The Dialogue Manager usually needs to interface with some external software such as a database or an expert system. The query or plans thus have to be converted from the internal representation used by the dialogue manager to the format used by the external domain specific system (e.g. SQL). This interfacing is handled by the domain specific components. This can be handled by Natural Language Query Processing system. This system generate SQL query from natural language.

Response Generator This component involves constructing the message that is to be given by the user. It takes decision regarding what information should be included, how information should be structured, choice of words and syntactic structure for message. Current systems use simple methods such as insertion of retrieved data into predefined slots in a template.

Speech Generation It translates the message constructed by the response generation component into spoken form. For speech generation two approaches may be used. The first approach is to use prerecorded canned speech may be used with spaces to be filled by retrieved or previously recorded samples e.g. Welcome, how can I help you. The secondapproach is use text to speech synthesis. In this speech is generated of text. It is called Contaminative Speech Synthesis, Text to Phoneme conversion and Phoneme to speech conversion or Text to Speech

6 Basic technologies to build a voice assistant

Voice/speech to text (STT) is the process of converting speech signal into digital data (e.g., text data). The voice may come as a file or a stream. You can use CMU Sphinx for its processing.

Text to speech (TTS) is the opposite process that translates text / images in a human speech. It is very useful when, for instance, a user wants to hear the correct pronunciation of a foreign word. Text to speech (TTS) is the opposite process that translates text / images in a human speech. It is very useful when, for instance, a user wants to hear the correct pronunciation of a foreign word.

Intelligent tagging and decision making serve for interpreting the user's request. For example, the user may ask: 'What do I watch tonight?'. The technology will tag the top-rated movies and suggest you a few according to your interests. The AlchemyAPI may help you build AI assistant that can cope with this task.

Image recognition is an optional but very useful feature. Later, you can use it for developing multimodal speech recognition. Have a look at OpenCV if you want to create an AI assistant with this feature under the hood.

Noise control : The noises from cars, electrical appliances, other people talking near you make the user's voice unclear. This technology will reduce or totally eliminate the background noise that prevents a correct voice recognition. If you want to build your own personal assistant, this feature can serve as a good addition which will enhance the overall user experience.

Voice Biometrics is a very important option security feature which you should take into account to create your own AI assistant. Thanks to this feature, the voice assistant may identify who is talking and whether it is necessary to respond. Thus, you may avoid a comic situation that happened to Siri and Amazon Alexa when they lowered the temperature in a house and even turned off someone's thermostat by hearing a relevant command from the TV speakers.

Speech compression : With this mechanism, the client side of the applications will resize the voice data and send it to the server in a succinct format. It will provide a fast application performance without annoying delays. To implement this mechanism, you can use G.711 standard.

Voice interface is what the user hears and sees in return to his or her request. For the voice part, you will need to pick up the voice itself, set the rate of speech, the manner of speaking, etc. For the visual part, you will have to decide on the visual representation that a user is going to see on the screen. If reasonable, you can skip it at all and make your own AI assistant without these adjustments.

7 Automated Speech Recognition

Speech recognition is invading our lives. Its built into our phones, our game consoles and our smart watches. Its even automating our homes.

1. Turning Sounds into Bits : The first step in speech recognition is -we need to feed sound waves into a computer.But sound is transmitted as waves. How do we turn sound waves into numbers? Sound waves are one-dimensional. At every moment in time, they have a single value based on the height of the wave.

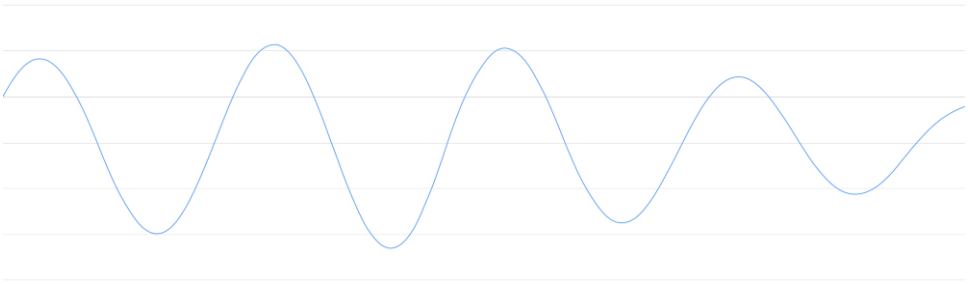

Figure 3: one dimensional sound waves

2. To turn this sound wave into numbers, we just record of the height of the wave at equally-spaced points:

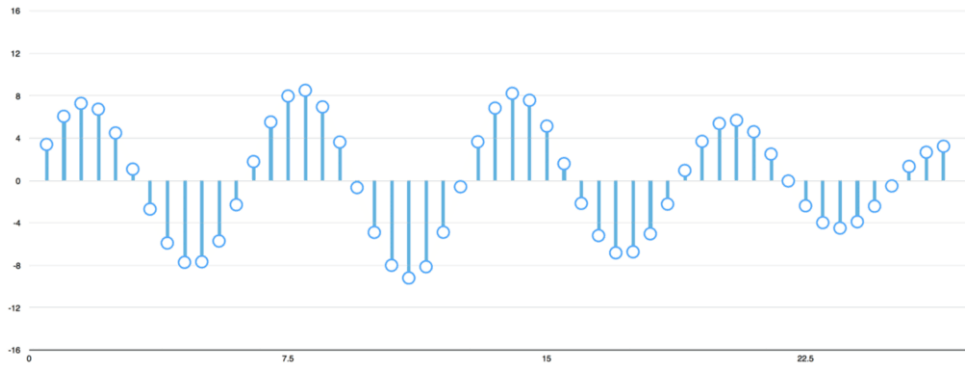

Figure 4: Sampling

This is called sampling. We are taking a reading thousands of times a second and recording a number representing the height of the sound wave at that point in time. Lets sample our Hello sound wave 16,000 times per second. Heres the first 100 samples:

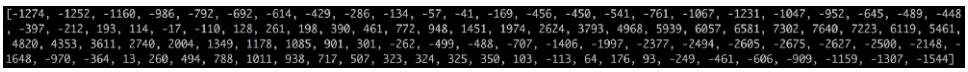

Figure 5: First 100 sample points of the word "Hello"

Sampling is only creating a rough approximation of the original sound wave because its only taking occasional readings. Theres gaps in between the readings so some data is lost.

But thanks to the Nyquist theorem, we know that we can use math to perfectly reconstruct the original sound wave from the spaced-out samplesas long as we sample at least twice as fast as the highest frequency we want to record.

3. Pre-processing our Sampled Sound Data We now have an array of numbers with each number representing the sound waves amplitude at 1/16,000th of a second intervals. We could feed these numbers right into a neural network. But trying to recognize speech patterns by processing these samples directly is difficult. Instead, we can make the problem easier by doing some pre-processing on the audio data. Lets start by grouping our sampled audio into 20-millisecond-long chunks. Heres our first 20 milliseconds of audio (i.e., our first 320 samples): Plotting those numbers as a simple line graph gives us a rough approximation of the original sound wave for that 20 millisecond period of time:

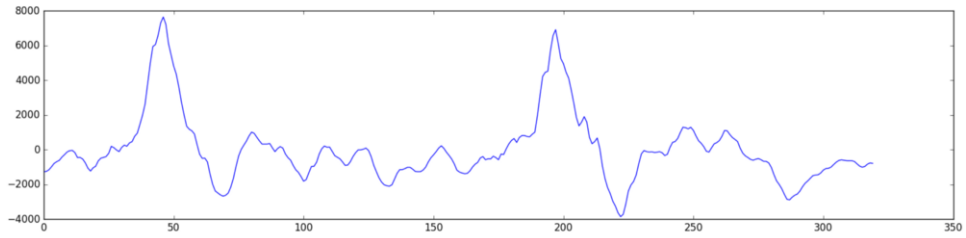

Figure 6: First 100 sample points of the word "Hello"

This recording is only 1/50th of a second long. But even this short recording is a complex mish-mash of different frequencies of sound. Theres some low sounds, some mid-range sounds, and even some high-pitched sounds sprinkled in. But taken all together, these different frequencies mix together to make up the complex sound of human speech. To make this data easier for a neural network to process, we are going to break apart this complex sound wave into its component parts. Well break out the low-pitched parts, the next-lowest-pitched-parts, and so on. Then by adding up how much energy is in each of those frequency bands (from low to high), we create a fingerprint of sorts for this audio snippet.

4. Fourier Transform We do this using a mathematic operation called a Fourier transform. It breaks apart the complex sound wave into the simple sound waves that make it up. Once we have those individual sound waves, we add up how much energy is contained in each one. The end result is a score of how important each frequency range is, from low pitch (i.e. bass notes) to high pitch. Each number below represents how much energy was in each 50hz band of our 20 millisecond audio clip: If we repeat this process on every 20 millisecond chunk of audio, we end up with a spectrogram (each column from left-to-right is one 20ms chunk):

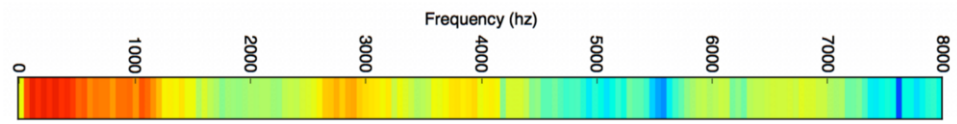

Figure 7: Spectrograph

A spectrogram is cool because you can actually see musical notes and other pitch patterns in audio data. A neural network can find patterns in this kind of data more easily than raw sound waves. So this is the data representation well actually feed into our neural network.

5. Recognizing Characters from Short Sounds Now that we have our audio in a format thats easy to process, we will feed it into a deep neural network. The input to the neural network will be 20 millisecond audio chunks. For each little audio slice, it will try to figure out the letter that corresponds the sound currently being spoken.

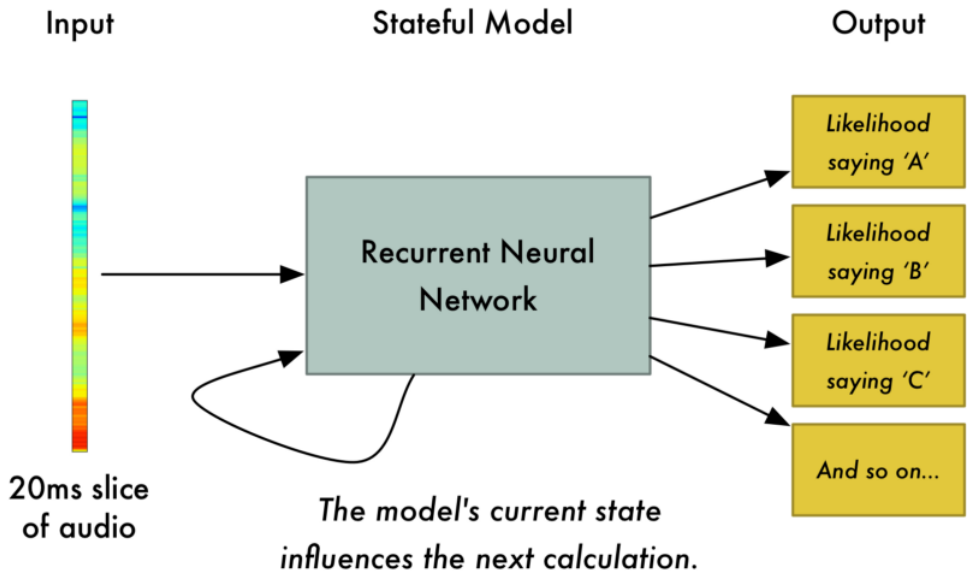

Figure 8: Neural Network Model

Well use a recurrent neural networkthat is, a neural network that has a memory that influences future predictions. Thats because each letter it predicts should affect the likelihood of the next letter it will predict too. For example, if we have said HEL so far, its very likely we will say LO next to finish out the word Hello. Its much less likely that we will say something unpronounceable next like XYZ. So having that memory of previous predictions helps the neural network make more accurate predictions going forward.

After we run our entire audio clip through the neural network (one chunk at a time), well end up with a mapping of each audio chunk to the letters most likely spoken during that chunk. Heres what that mapping looks like for me saying Hello:

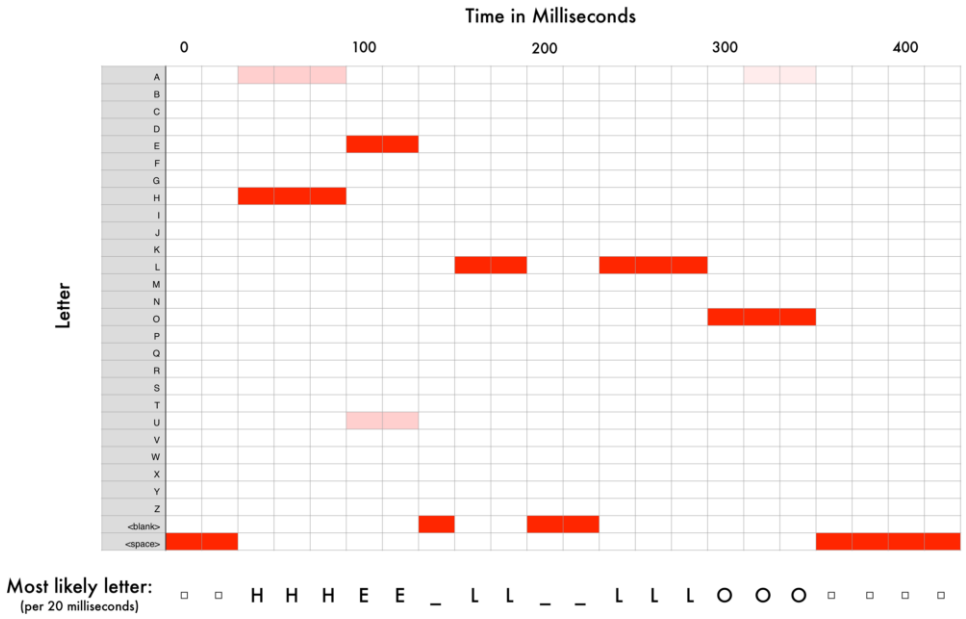

Figure 9: Graph of Time VS Frequently occuring letters

We have some steps we follow to clean up this output. First, well replace any repeated characters a single character:

(a) HHHEE-LL-LLLOOO becomes HE-L-LO

(b) HHHUU-LL-LLLOOO becomes HU-L-LO

(c) AAAUU-LL-LLLOOO becomes AU-L-LO

Then well remove any blanks:

(a) HE-L-LO becomes HELLO

(b) HU-L-LO becomes HULLO

(c) AU-L-LO becomes AULLO

That leaves us with three possible transcriptionsHello, Hullo and Aullo. If you say them out loud, all of these sound similar to Hello. Because its predicting one character at a time, the neural network will come up with these very sounded-out transcriptions. Of our possible transcriptions Hello, Hullo and Aullo, obviously Hello will appear more frequently in a database of text (not to mention in our original audio-based training data) and thus is probably correct. So well pick Hello as our final transcription instead of the others.

8 Hidden Markov Model

Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states.

The hidden Markov model can be represented as the simplest dynamic Bayesian network. The mathematics behind the HMM were developed by L. E. Baum and coworkers. HMM is closely related to earlier work on the optimal nonlinear filtering problem by Ruslan L. Stratonovich, who was the first to describe the forward-backward procedure.

In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters, while in the hidden Markov model, the state is not directly visible, but the output (in the form of data or "token" in the following), dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore, the sequence of tokens generated by an HMM gives some information about the sequence of states; this is also known as pattern theory, a topic of grammar induction.

The adjective hidden refers to the state sequence through which the model passes, not to the parameters of the model; the model is still referred to as a hidden Markov model even if these parameters are known exactly.

Hidden Markov models are especially known for their application in reinforcement learning and temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics.

A hidden Markov model can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other. Recently, hidden Markov models have been generalized to pairwise Markov models and triplet Markov models which allow consideration of more complex data structures and the modeling of nonstationary data.

9 Text to Speech

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written words on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s. Overview of a typical TTS system

Automatic announcement Menu A synthetic voice announcing an arriving train in Sweden. Problems playing this file? See media help.

A text-to-speech system (or "engine") is composed of two parts:[3] a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-endoften referred to as the synthesizerthen converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

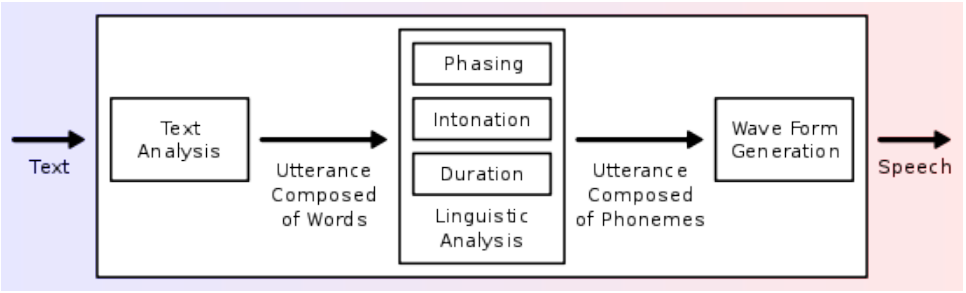

Figure 10: Overview of a typical TTS system

10 Conclusion

This proposal introduces the structure of Next-Generation of Virtual Personal Assistants that is a new VPAs system designed to converse with a human, with a coherent structure. This VPAs system has used speech, graphics, video, gestures and other modes for communication in both the input and output channel. Also, the VPAs system will be used to increase the interaction between users and the computers by using some technologies such as gesture recognition, image/video recognition, speech recognition, and the Knowledge Base. Moreover, this system can enable a lengthy conversation with users by using the vast dialogue knowledge base. Moreover, this system can be used in different tasks such as education assistance, medical assistance, robotics and vehicles, disabilities systems, home automation, and security access

control. Also, it can be a satisfactory solution that can be used by applications, such as responding to customers, customer service agent, training or education, facilitating transactions, online shopping, travelling information, counseling, tutoring system, ticket booking, remote banking, travel reservation, Information enquiry, stock transactions, taxi bookings, and route planning etc. In the end, to achieve the final stage and all these improvements to the new system with high accuracy, we need funding from an organization that will work with us to improve the system by funding the new hardware devises that have high accuracy, as well as the tools and cloud severs that we will need for testing the new system.

11 References

1. https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8.
2. Suket Arora 1, Kamaljeet Batra, Sarabjit Singh. Dialogue System: A Brief Review. 2013.