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

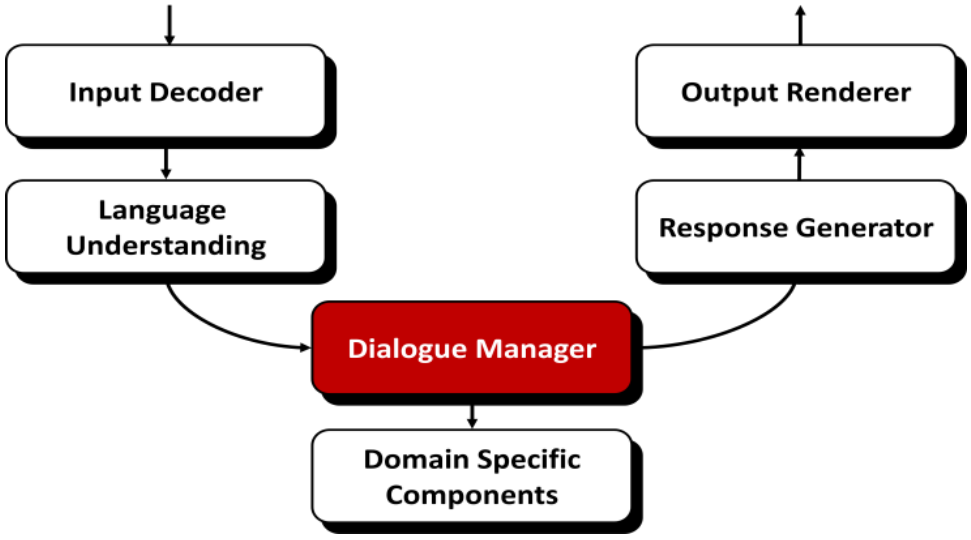


Figure 2: The Structure of The Next-Generation of Virtual Personal Assistants

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 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.

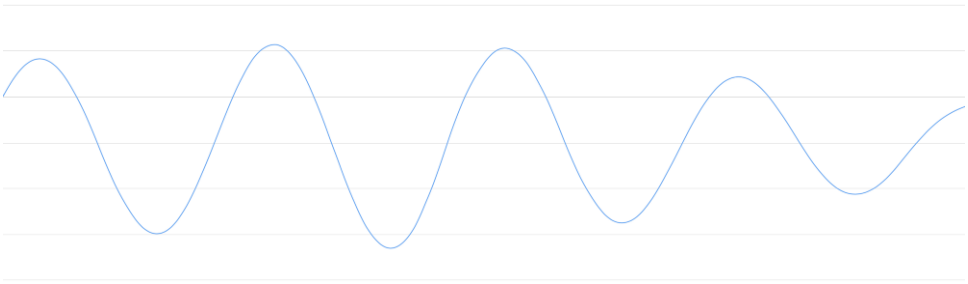


Figure 3: one dimensional sound waves

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

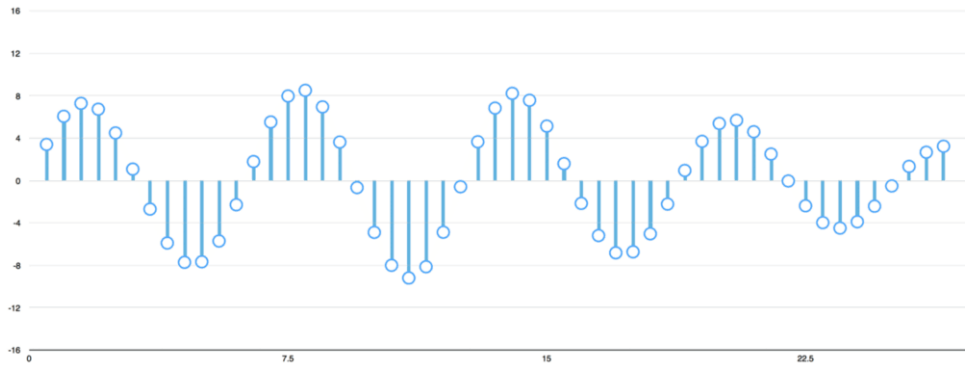


Figure 4: Sampling

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:

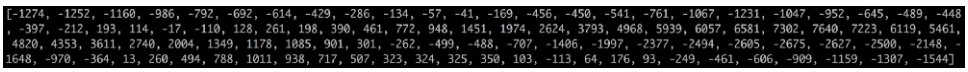


Figure 5: First 100 sample points of the word "Hello"

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:

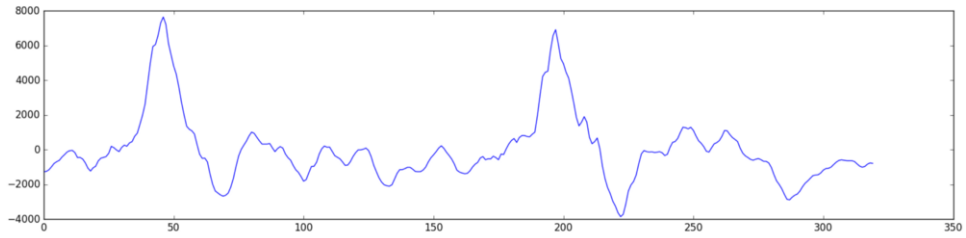


Figure 6: First 100 sample points of the word "Hello"

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):

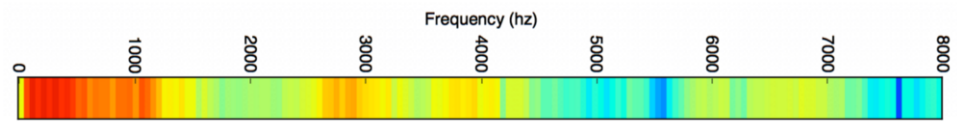


Figure 7: Spectrograph

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.

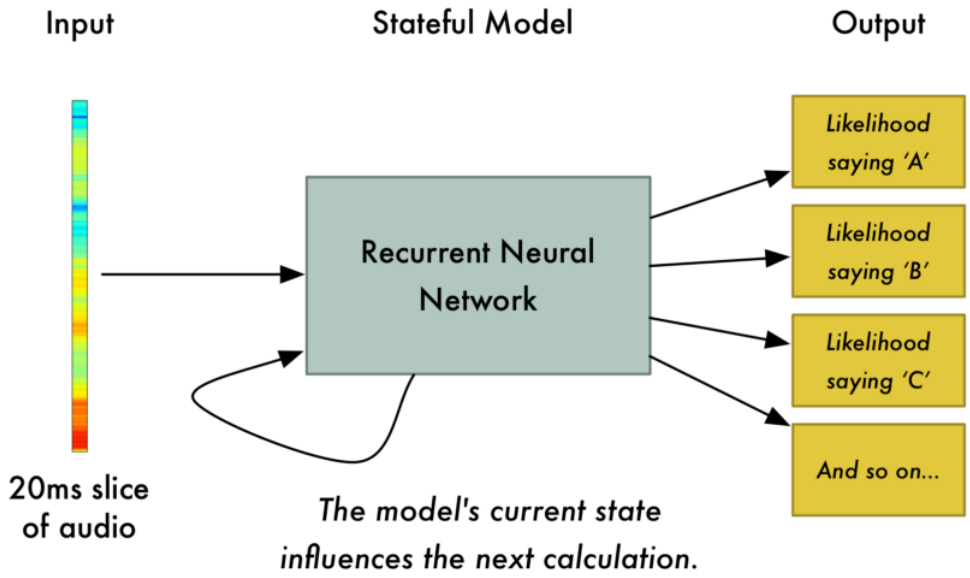


Figure 8: Neural Network Model

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:

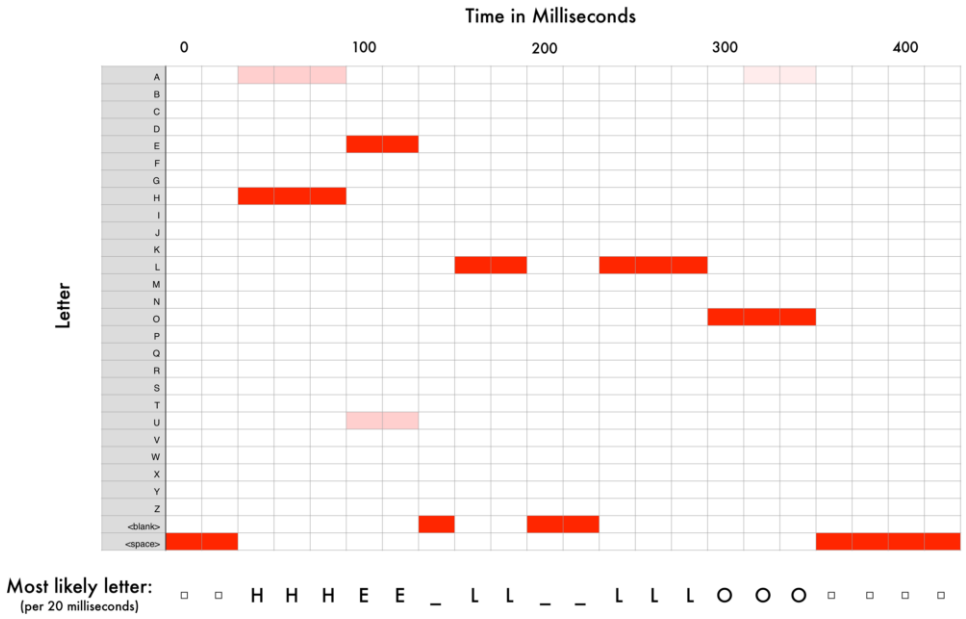


Figure 9: Graph of Time VS Frequently occuring letters

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.

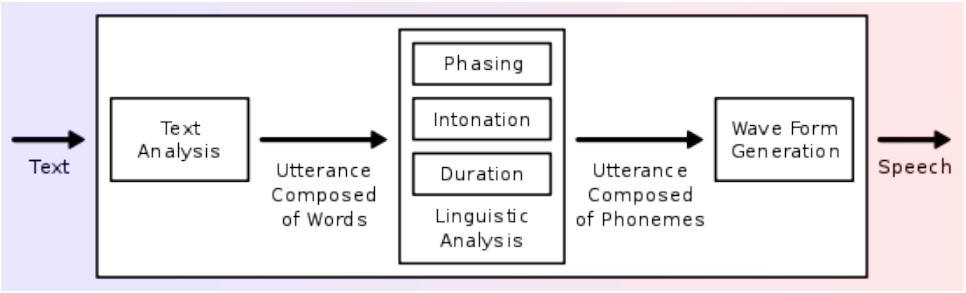


Figure 10: Overview of a typical TTS system

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.

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.

Shantanu Kalamdane T150074264

(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 Proposed Architecture

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. We have modified and added some components in the original structure of general 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. The following is the structure of the Next-Generation of Virtual Personal Assistants:

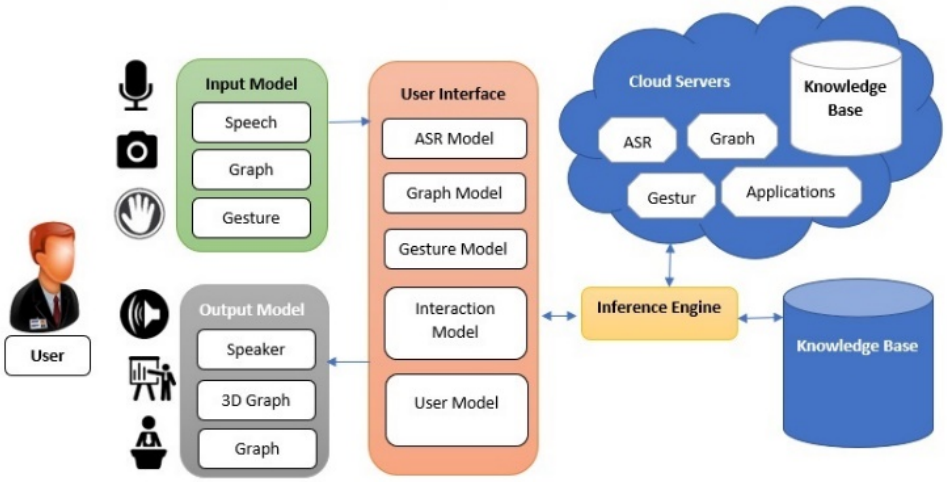


Figure 1: The Structure of The Next-Generation of Virtual Personal Assistants

4.1 Knowledge Base

There are two knowledge bases. The first is the online and the second is local knowledge base which include all data and facts based on each model, such as facial and body data sets for gesture modal, speech recognition knowledge bases, dictionary and spoken dialog knowledge base for ASR modal, video and image body data sets for Graph Model, and some users information and the setting system.

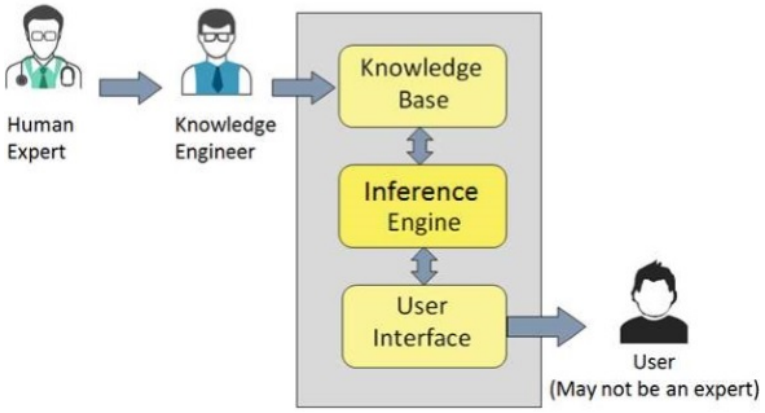


Figure 2: The Knowledge Base

4.1.1 Knowledge System

A Knowledge-Based System (KBS) is a computer program that reasons and uses a knowledge base to solve complex problems. The one common theme that unites all knowledge based systems is an attempt to represent knowledge explicitly and a reasoning system that allows it to derive new knowledge.

4.1.2 Interference Engine

An inference engine is a computer program that applies artificial intelligence to try to obtain answers or responses to queries from a knowledge base. A programs protocol for navigating through the rules and data in a knowledge system in order to solve the problem. The major task of the inference engine is to select and then apply the most appropriate rule at each step as the expert system runs, which is called rule-based reasoning. The part of a decision support system that performs the reasoning function.

4.2 Graph Model

The Graph Model analyzes video and image in real-time by using the Graph Model and extracts frames of the video that collect by the camera and the input model; then it sends those frames and images to the Graph Model and applications in Cloud Servers for analyzing those frames and images and returning the result.

4.3 Gesture Model

The gesture model uses the camera and Kinect in the input model to read the movements of the human body and the facial expressions; then it sends all data to the gesture model and applications in Cloud Servers to analyze those frames and images and returning the result.

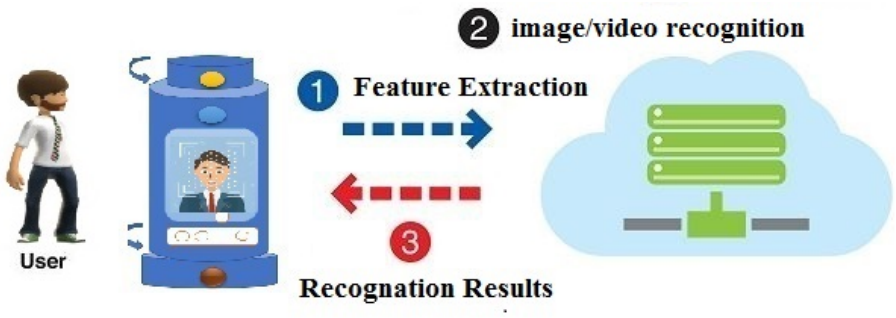


Figure 3: The Graph Model

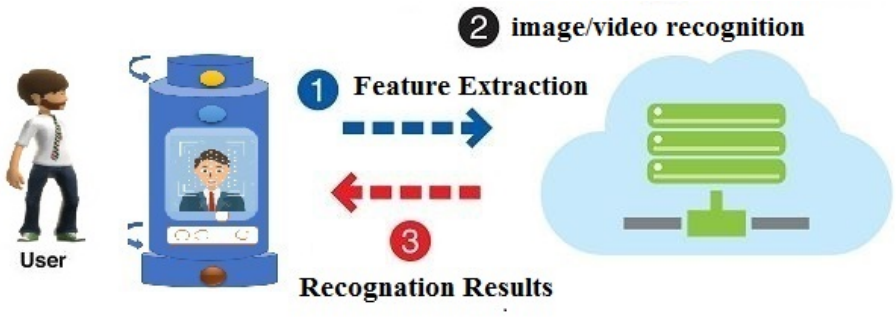


Figure 4: The Graph Model

4.4 ASR Model

The speech recognition model will work in real-time with the microphone in the input model with the ASR model in Cloud Servers to recognize the utterances that a user speaks into a microphone and then convert it to text; then it sends the text to the applications in Cloud Servers to analyze the text and returning the result.

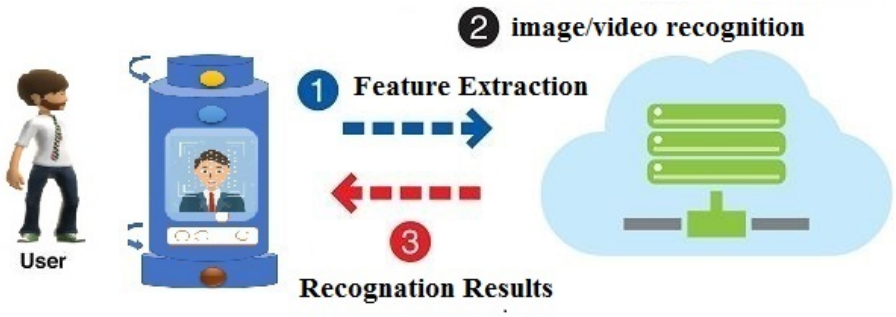


Figure 5: The Graph Model

4.5 Interaction Model

This is the main model that will be used to provide interaction between users of the system and the system models by receiving the data from the input model and analyzing the data to send for each model based on its tasks, then returning result that will be used to make the final decision.

4.6 Interference Engine

The inference engine works together with the Interaction Model in the chain of conditions and derivations and finally deduces the outcome. they analyze all the facts and rules, then sorts them before concluding to a solution.

4.7 User Model

This model has all information about the users that will use the system. It can include personal information such as users' names and ages, their interests, their skills and knowledge, their goals and plans, their preferences and their dislikes or data about their behavior and their interactions with the system. All information will be collected by asking the user some questions then storing all answers in the Knowledge Base.

User models are the subdivision of human computer interaction which describes the process of building up and modifying a conceptual understanding of the user. The main goal of user models is customization and adaptation of systems to the user's specific needs. The system needs to "say the 'right' thing at the 'right' time in the 'right' way".To do so it needs an internal representation of the user. Another common purpose is modeling specific kinds of users, including modeling of their skills and declarative knowledge, for use in automatic software-tests.User-models can thus serve as a cheaper alternative to user testing.

4.8 Input Model

This model has all information about the users that will use the system. It can include personal information such as users' names and ages, their interests, their skills and knowledge, their goals and plans, their preferences and their dislikes or data about their behavior and their interactions with the system. All information will be collected by asking the user some questions then storing all answers in the Knowledge Base.

4.9 Output Model

This model will receive the final decision from the Interaction Model with an explanation, then it will choose the perfect output device to show the result such data show, speakers or screen based on the result.

5 Graph Model

5.1 Computer Vision

Computer vision is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.[1][2][3] "Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding."[9] As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner.[10] As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems.

5.2 Image Recognition

There are a number of different types of artificial intelligence, and one major flavor of AI is called Computer Vision. It refers to the ability of computers to acquire, process, and analyze data coming primarily from visual sourcesthe ability to track or predict movement for instance but could also include data from heat sensors and other similar source.

You might call image recognition a subset of computer vision, in that it refers to the ability of a computer to see, to decipher and understand the information fed to it from an image, be it a still, video, graphic, or even live. This is no small feat. If youve ever scratched your head at a bizarre spelling or grammar correction that Google, Siri or Microsoft Word suggest, then you get an idea of how tough it is for computers to understand the rules of written language, even though they are predictable and consistent. It gets even more complicated when computers tackle the visual.

Consider that a photo, image, or video is infinitely more complex and open-ended than the words that make up a sentence. Think of a newborn that is dazzled by light and color, and you begin to a touch the experience of a computer that has no pre-defined way of understanding what all the various data in an image are. In fact, to a computer, a photo is simply a bunch of tiny colored dots arrayed in pattern (what we call pixels, to be more precise). In order to make sense of what those dots all mean, the computer needs to first understand that patterns make up things called objects, and objects exist in space and have dimensions, and on an on. Thats a pretty steep learning curve. (In fact, as humans we use about half our brain power to process visual information!)

A digital image represents a matrix of numerical values. These values represent the data associated with the pixel of the image. The intensity of the different pixels, averages to a single value, representing itself in a matrix format. The information fed to the recognition systems is the intensities and the location of different pixels in the image. With the help of this information, the systems learn to map out a relationship or pattern in the subsequent images supplied to it as a part of the learning process. After the completion of the training process, the system performance on test data is validated. In order to improve the accuracy of the system to recognize images, intermittent weights to the neural networks are modified to improve the accuracy of the systems. Some of the algorithms used in image recognition (Object Recognition, Face Recognition) are SIFT (Scale-invariant Feature Transform), SURF (Speeded Up Robust Features), PCA (Principal Component Analysis), and LDA (Linear Discriminant Analysis).

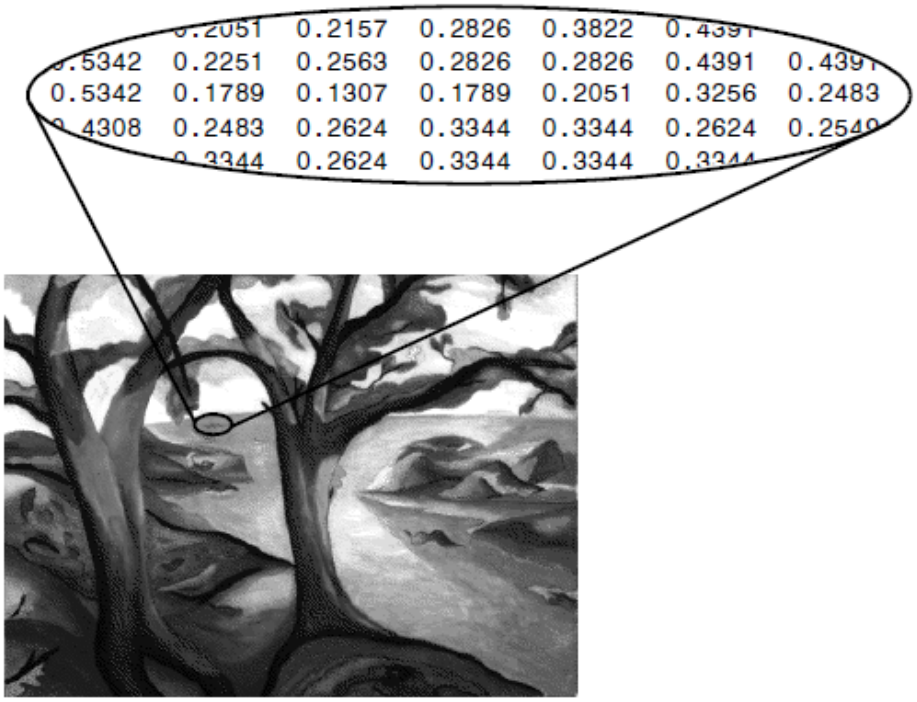


Figure 6: small section of an image represented in Matrix Format

6 Gesture Model

6.1 Gesture Recognition

Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focuses in the field include emotion recognition from face and hand gesture recognition. Users can use simple gestures to control or interact with devices without physically touching them. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. However, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques.[1] Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at this point will move accordingly. This could make conventional input on devices such and even redundant.

6.2 Gesture recognition features:

- More Accurate
- High Stability
- Time saving to unlock a device

6.3 Major application areas of gesture recognition in the current scenario are:

- Automotive Sector
- Consumer Electronics sector
- Transit Sector
- Defense

7 Kinect

KINECT 3D sensing camera (development code name "Project Natal") is used in the hardware part, while the function of real-time capture, microphone input, speech recognition, image recognition, interactive community features and so on are also included in this part to accurately identify human body and real time capture. As shown in figure Kinect has three lenses, left and right sides of the lens respectively are the IR emitter and infrared sensors that can be used for position control and collecting in-depth data (the distance from the camera) by matching.The middle one is RGB color cameras which is used for collecting image location to locate the image positions. Color camera support 1280*960 resolution imaging, infrared camera support max 640*480 imaging. Kinect can also be used to focus, and the electric motor can be adjusted to capture objects and images. Kinects basic structure and function is largely described as above. This article takes advantage these features to realize interactive control.

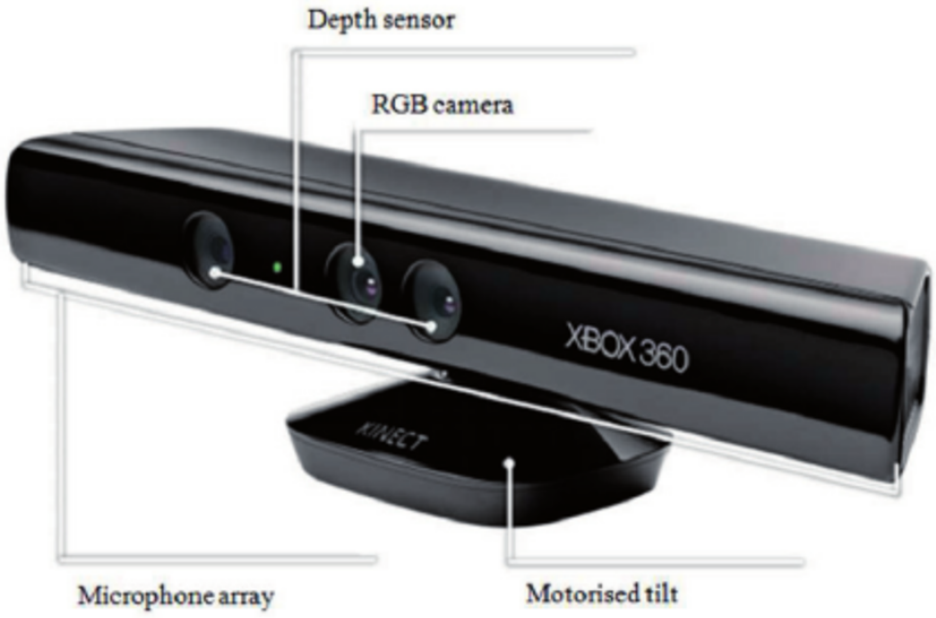


Figure 7: Kinect Devices

8 Object Detection by Voila Jones

Our object detection procedure classifies images based on the value of simple features. There are many motivations for using features rather than the pixels directly. The most common reason is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. For this system there is also a second critical motivation for features: the feature based system operates much faster than a pixel-based system. The simple features used are reminiscent of Haar basis functions which have been used by Papageorgiou et al. More specifically, we use three kinds of features. The value of a two-rectangle feature is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent. A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a four-rectangle feature computes the difference between diagonal pairs of rectangles. Given that the base resolution of the detector is 24x24, the exhaustive set of rectangle features is quite large, over 180,000 . Note that unlike the Haar basis, the set of rectan- gle features is overcomplete.

8.0.1 Integral Image

Rectangle features can be computed very rapidly using an intermediate representation for the image which we call the integral image. The integral image at location contains the sum of the pixels above and to the left of, inclusive:

$ii(x, y) = \sum_{x^1 \leq x, y^1 \leq y} i(x^1, y^1)$

Given example images (x1,y1),...(xn,yn) where yi = 0 for negative and positive examples respectively. Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for yi = 0,1 respectively, where l and m are the numbers of negatives andpositives respectively. For t = 1,...,T

Normalize the weights,

$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n}} w_{t,j}$

so that $w_t.$ is a probability distribution.

For each feature, 6 , train a classifier 7 3 which is restricted to using a single feature. The error is evaluated with respect to $w_t, \epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$. Choose the classifier, h_t , with the lowest error ϵ_t. Update the weights:

$w_t + 1, i = w_{t,i} \beta^{1-e_i}$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

The final strong classifier is:

h(x) = For x = 1 , $\sum_{t=1}^{T}\alpha_t h_t(x) \geq 1/2\sum_{t=1}^{T}\alpha_t$

For x = 0 , Otherwise

where $\alpha_t = log\frac{1}{\beta_t}$

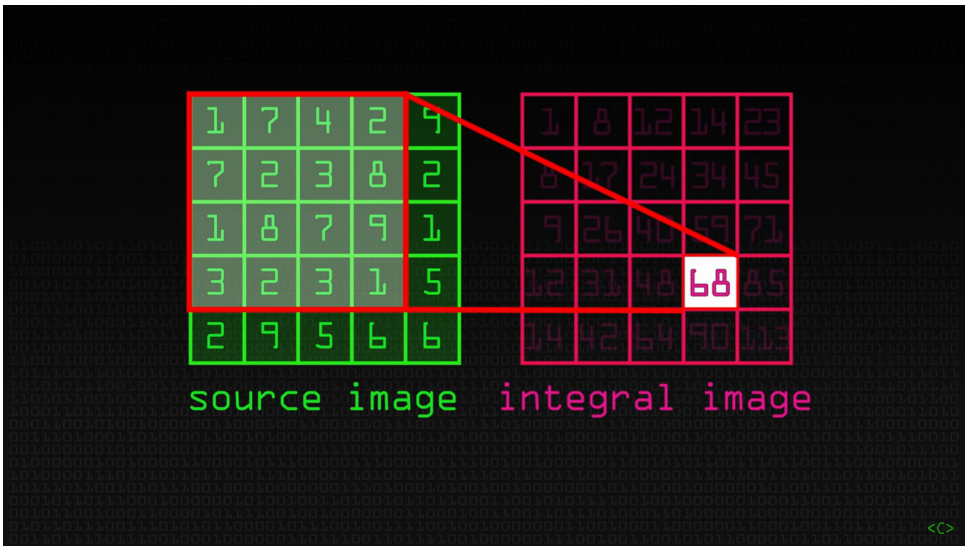


Figure 8: Source and Integral image

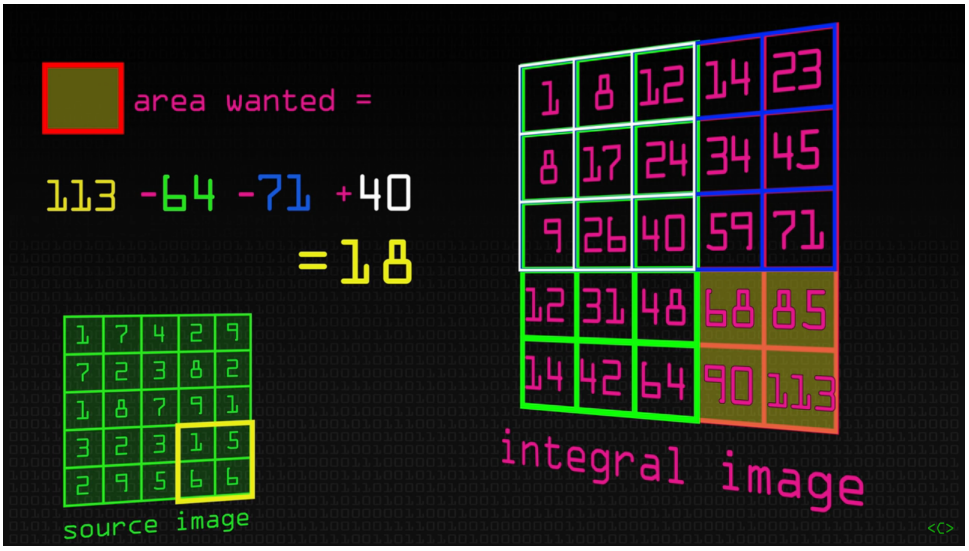


Figure 9: Finding out resultant area in an image

9 Graph and Gesture Recognition using Deep Learning

9.0.1 Deep Learning

Deep learning is a class of machine learning algorithms that :

1. use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.
2. learn in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) manners.
3. learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

Deep learning models are vaguely inspired by information processing and communication patterns in biological nervous systems yet have various differences from the structural and functional properties of biological brains (especially human brains), which make them incompatible with neuroscience evidences.

9.0.2 Convolutional Neural Networks (CNN)

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery.

CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing.They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.

1. Convolutional

   Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli.

   Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical

to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10000 weights for each neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters.[12] For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using backpropagation..

2. Pooling

   Convolutional networks may include local or global pooling layers,[clarification needed] which combine the outputs of neuron clusters at one layer into a single neuron in the next layer.[13][14] For example, max pooling uses the maximum value from each of a cluster of neurons at the prior layer.[15] Another example is average pooling, which uses the average value from each of a cluster of neurons at the prior layer.

3. Fully Connected

   Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP).The flattend matrix goes through a fully connected layer to classify the images

4. Receptive Fields

   In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from every element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically the subarea is of a square shape (e.g., size 5 by 5). The input area of a neuron is called its receptive field. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer.

9.0.3 Working

Let us consider the use of CNN for image classification in more detail. The main task of image classification is acceptance of the input image and the following definition of its class. This is a skill that people learn from their birth and are able to easily determine that the image in the picture is an elephant. But the computer sees the pictures quite differently:

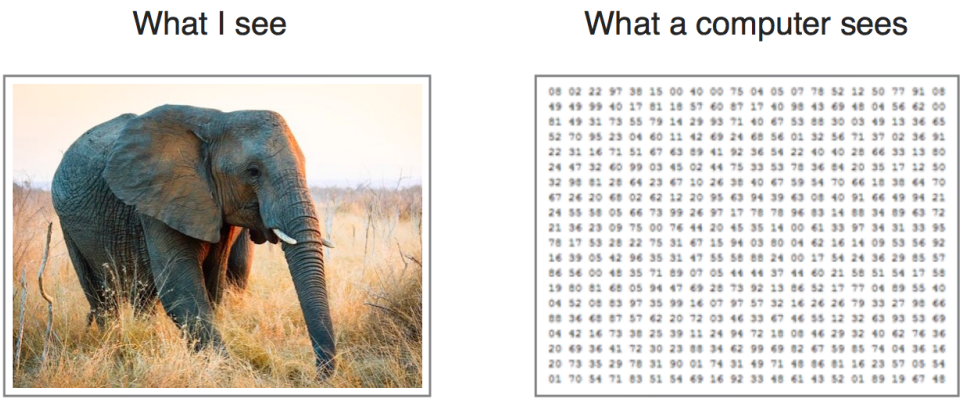


Figure 10: Finding out resultant area in an image

Instead of the image, the computer sees an array of pixels. For example, if image size is 300 x 300. In this case, the size of the array will be 300x300x3. Where 300 is width, next 300 is height and 3 is RGB channel values. The computer is assigned a value from 0 to 255 to each of these numbers. his value describes the intensity of the pixel at each point.

To solve this problem the computer looks for the characteristics of the base level. In human understanding such characteristics are for example the trunk or large ears. For the computer, these characteristics are boundaries or curvatures. And then through the groups of convolutional layers the computer constructs more abstract concepts.

In more detail: the image is passed through a series of convolutional, nonlinear, pooling layers and fully connected layers, and then generates the output.

The Convolution layer is always the first. he image (matrix with pixel values) is entered into it. Imagine that the reading of the input matrix begins at the top left of image. Next the software selects a smaller matrix there, which is called a filter (or neuron, or core). Then the filter produces convolution, i.e. moves along the input image. The filters task is to multiply its values by the original pixel values. All these multiplications are summed up. One number is obtained in the end. Since the filter has read the image only in the upper left corner, it moves further and further right by 1 unit performing a similar operation. After passing the filter across all positions, a matrix is obtained, but smaller then a input matrix.

This operation, from a human perspective, is analogous to identifying boundaries and simple colours on the image. But in order to recognize the properties of a higher level such as the trunk or large ears the whole network is needed.

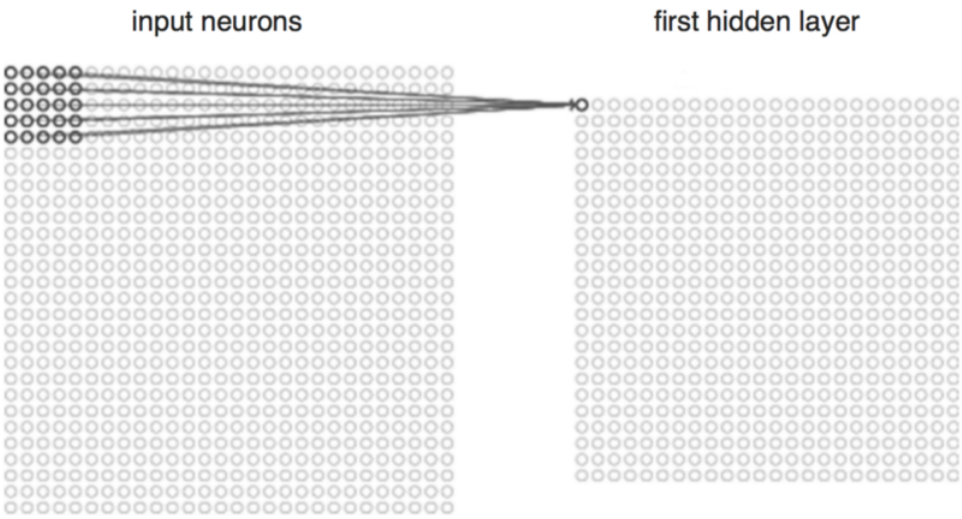


Figure 11: Finding out resultant area in an image

The network will consist of several convolutional networks mixed with nonlinear and pooling layers. When the image passes through one convolution layer, the output of the first layer becomes the input for the second layer. And this happens with every further convolutional layer.

The nonlinear layer is added after each convolution operation. It has an activation function, which brings nonlinear property. Without this property a network would not be sufficiently intense and will not be able to model the response variable (as a class label).

The pooling layer follows the nonlinear layer. It works with width and height of the image and performs a downsampling operation on them. As a result the image volume is reduced. This means that if some features (as for example boundaries) have already been identified in the previous convolution operation, than a detailed image is no longer needed for further processing, and it is compressed to less detailed pictures.

After completion of series of convolutional, nonlinear and pooling layers, it is necessary to attach a fully connected layer. This layer takes the output information from convolutional networks. Attaching a fully connected layer to the end of the network results in an N dimensional vector, where N is the amount of classes from which the model selects the desired class.

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. https://www.youtube.com/watch?v=uEJ71VlUmMQ
3. Paul Voila and Micheal Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. 2001

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.

Akash Rasal T150074255

(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 Holobot

Holobot is a combination of hologram and virtual personal assistant devices. Holograms on integration with virtual personal assistant will make an exceptionally well duo. It will combine a complex physics and a complex computer related technology to form a new device. Holobots will be enabled with cameras, projectors, kinnect, etc. Gesture recognition, emotion recognition, holographic videochats will become possible with holobots.

3.1 Gesture recognition

Gesture recognition is a type of perceptual computing user interface that allows computers to capture and interpret human gestures as commands. The general definition of gesture recognition is the ability of a computer to understand gestures and execute commands based on those gestures. How is a gesture defined? In order to understand how gesture recognition works, it is important to understand how the word gesture is defined. In its most general sense, the word gesture can refer to any non-verbal communication that is intended to communicate a specific message. In the world of gesture recognition, a gesture is defined as any physical movement, large or small, that can be interpreted by a motion sensor. It may include anything from the pointing of a finger to a roundhouse kick or a nod of the head to a pinch or wave of the hand. Gestures can be broad and sweeping or small and contained. In some cases, the definition of gesture may also include voice or verbal commands. For instance, Kinect looks at a range of human characteristics to provide the best command recognition based on natural human inputs. It provides both skeletal and facial tracking in addition to gesture recognition, voice recognition and in some cases the depth and color of the background scene. Kinect reconstructs all of this data into printable three-dimensional (3D) models. The latest Kinect developments include an adaptive user interface that can detect a users height.

Figure 1: Holobot demo by Microsoft

3.2 Emotion Recognition

Emotion recognition is a technique used in software that allows a program to "read" the emotions on a human face using advanced image processing. Companies have been experimenting with combining sophisticated algorithms with image processing techniques that have emerged in the past ten years to understand more about what an image or a video of a person's face tells us about how he/she is feeling. With advances in technology, emotion recognition software has become very capable. Besides its ability to track basic facial expressions for emotion such as sadness, happiness, anger, surprise, etc.,

emotion recognition software can also capture what experts call "micro-expressions" or subtle body language cues that may betray an individuals feelings without his/her knowledge. In many senses, emotion recognition goes along with other kinds of facial profiling such as biometric image recognition. And both of these types of technology can be used for the same types of security purposes. For instance, law enforcers can use emotion recognition software to try to get more information about someone during interrogation. Emotion recognition continues to evolve in much the same way as other new technologies like natural language processing are advancing, and these advances are largely made possible by the combination of ever more powerful processors, the scientific development of sophisticated algorithms, and other related innovations.

3.3 Holographic video chats

Whether you need to inform Obi-Wan he is your only hope, or just want to hang out with loved ones who are far away, who hasnt dreamed of a future involving holograms? Microsofts holoportation demo shows that its possible- if youve got a couple of Hololenses and a couple of rooms surrounded by 3D cameras anyway. Its not a perfect re-creation: you can tell that youre looking at a hologram because of occasional glitches and an overall low polygon count. But its not hard to imagine this being more intimate than a Skype chat. The setup isnt simple. A HoloLens is required to see the holograms in realtime, and a room surrounded by 3D cameras is necessary to create them. And however good the technology becomes, theres one barrier to this feeling deeply immersive: the HoloLens itself. You cant see other people using this tech unless youre wearing a HoloLens, and you cant make eye contact with people who are wearing the augmented reality device. To see another persons holographic face, they need to take off their headset which means they cant see you.

In one of the demonstrations by Microsoft, Izadi Interacts with his daughter Lilly, who is not wearing a HoloLens. His daughter cannot see him, and at one point almost walks straight through him because of this. So theres a design problem here that headsets cant readily solve. But its still a remarkable example of what sort of tech well have access to in the future, and its particularly cool to see the recorded conversation played back, and even shrunk.

The recent hologram call was apparently limited to a 3D image on a monitor, rather than being a Star-Wars style projected hologram. But its an exciting development that suggests holographic calling isnt too far away though, well at least have to wait until 5G networks arrive in full. Most estimates suggest 2020 as the year for an initial rollout of 5G network technology, but there has to be an agreed standard before that happens. In other words, itll be a while before we see holographic calls arriving, but work has already started on making them a reality. KT is working on the holographic call as part of a set of 5G-based immersive media which also includes 360-degree Live VR.

4 VPAs for commercial offices

Limiting virtual personal assistants to homes will be restricting its abilities. VPAs can be used in office to read out emails, setup meeting, and see attachments. Commercial VPAs will be enabled with cameras, projectors, etc. Cameras can be used for gesture recognition while projectors can be used to view attachments of the emails. Also, text summarisation can be done on the emails to hear exact summary instead of hearing the whole email.

4.1 Text summarisation

What Is Automatic Text Summarization? Summarizer is a micro service that uses the Classifier4J framework and its summarization module to scan through large documents and returns the sentences that are most likely useful for generating a summary. Automatic summarization of text works by first calculating the word frequencies for the entire text document. Then, the 100 most common words are stored and sorted. Each sentence is then scored based on how many high frequency words it contains, with higher frequency words being worth more. Finally, the top X sentences are then taken, and sorted based on their position in the original text. By keeping things simple and general purpose, the automatic text summarization algorithm is able to function in a variety of situations that other implementations might struggle with, such as documents containing foreign languages or unique word associations that arent found in Standard English language corpuses.

Why You Need Text Summarization? Business leaders, analysts, paralegals, and academic researchers need to comb through huge numbers of documents every day to keep ahead, and a large portion of their time is spent just figuring out what document is relevant and what isnt. By extracting important sentences and creating comprehensive summaries, its possible to quickly assess whether or not a document is worth reading. Automatic text summarization is also useful for students and authors. Imagine being able to automatically generate an absjustract based for your research paper or chapter in a book in a clear and concise way that is faithful to the original source material!

Abstraction-based summarization

The abstraction technique entails paraphrasing and shortening parts of the source document. When abstraction is applied for text summarization in deep learning problems, it can overcome the grammar inconsistencies of the extractive method. The abstractive text summarization algorithms create new phrases and sentences that relay the most useful information from the original text just like humans do. Therefore, abstraction performs better than extraction. However, the text summarization algorithms required to do abstraction are more difficult to develop; thats why the use of extraction is still popular. Here is an example: Abstractive summary: Joseph and Mary came to Jerusalem where Jesus was born. How does a text summarization algorithm work? Usually, text summarization in NLP is treated as a supervised machine learning problem (where future outcomes are predicted based on provided data). Typically, here is how using the extraction-based approach to summarize texts can work: 1. Introduce a method to extract the merited keyphrases from the source document. For example, you can use part-of-speech tagging, words sequences, or other linguistic patterns to identify the keyphrases. 2. Gather text documents with positively-labeled keyphrases. The keyphrases should be compatible to the stipulated extraction technique. To increase accuracy, you can also create negatively-labeled keyphrases. 3. Train a binary machine learning classifier to make the text summarization. Some of the features you can use include: Length of the keyphrase Frequency of the keyphrase The most recurring word in the keyphrase Number of characters in the keyphrase 4. Finally, in the test phrase, create all the keyphrase words and sentences and carry out classification for them.

5 Smart displays

What is a smart display?

Smart displays add a touchscreen to the mix, which lets you watch videos or look at pictures. The smart displays we've tested can also walk you through a recipe step by step, show you detailed weather forecasts or give info about restaurants if you're searching for something to eat.

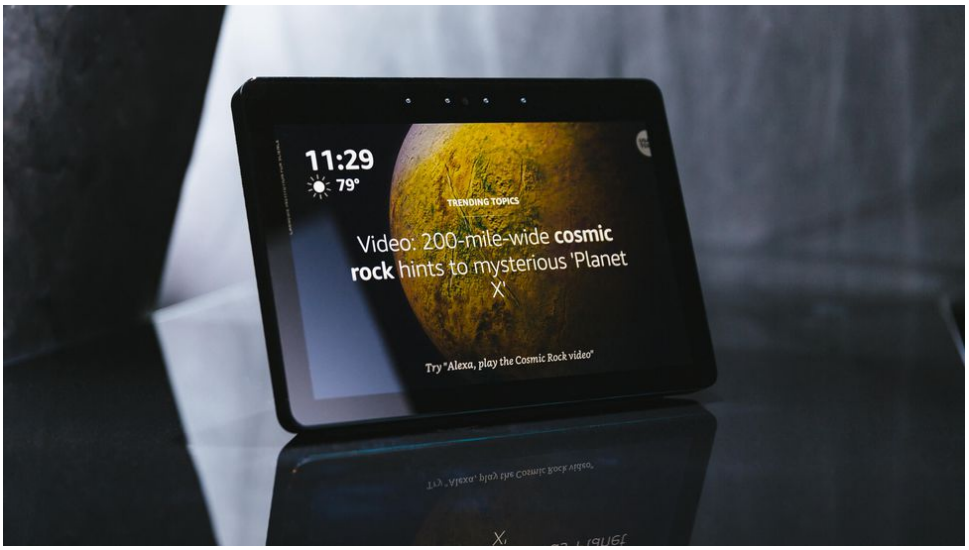


Figure 2: Concept Smart Screen

The second-gen Amazon Echo Show ushered in an updated wave of smart displays. Tyler Lizenby/CNET Currently, smart displays either use Amazon's digital assistant Alexa or Google's competitive Google Assistant. Your options with Alexa are the newly released second-generation Amazon Echo Show and the upcoming Facebook Portal. Google Assistant is built in to the Lenovo Smart Display, the JBL Link View and the upcoming smart displays from LG and Sony. The upcoming Google Home Hub also features Google Assistant.

Screen sizes run from 7 to 10 inches, so smart displays won't replace your television. Without traditional apps, they're also not as functional as tablets, though you will find the same access to Google's Actions and Alexa's Skills, which are voice-centric apps that teach your assistant new tricks. Smart displays aren't really designed for surfing the web, either. Instead, they offer a simple interface you can see from across the room, and are meant more for simple interactions designed to enhance your initial voice query.

All smart displays we've seen also allow you to make video calls, but the upcoming Google Home Hub is an outlier in that it doesn't have a camera. You can still make video calls with it, but the person you're calling won't be able to see you (and yes, some folks might prefer that approach). The rest of the current crop of smart displays all have cameras for making two-way video calls.

6 Hound

HOUND Voice Search and Mobile Assistant content rating is everyone. This app is listed in Productivity category of app store. You could visit SoundHound Inc.'s website to know more about the company/developer who developed this. HOUND Voice Search and Mobile Assistant can be downloaded and installed on android devices supporting 16 api and above. Download the app using your favorite browser and click on install to install the app. Please note that we provide original and pure apk file and provide faster download speed than HOUND Voice Search and Mobile Assistant apk mirrors. You could also download apk of HOUND Voice Search and Mobile Assistant and run it using popular android emulators.

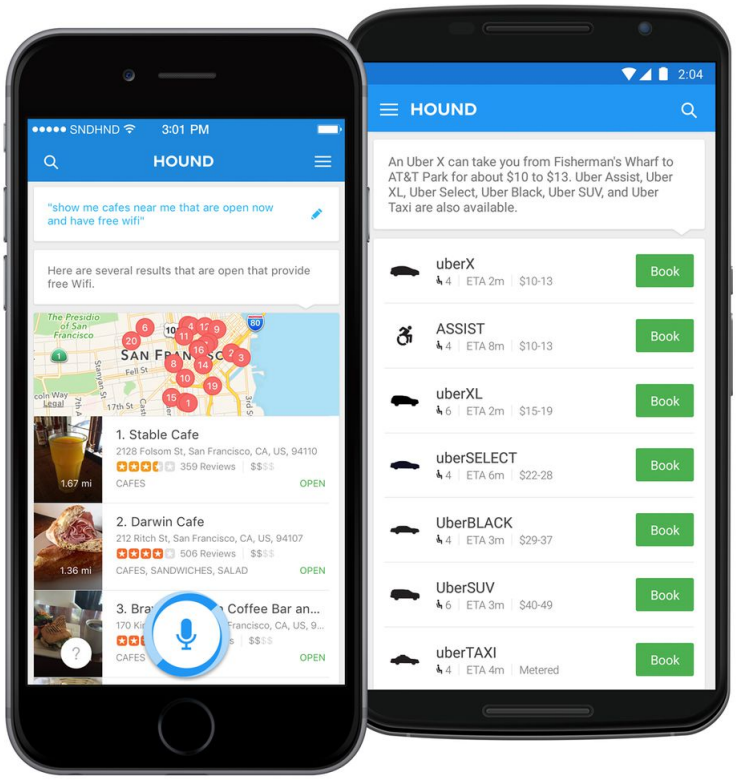


Figure 3: Hound app display

Nearly every one of the world's largest technology companies is trying to figure out how to let computers understand human speech, but a Santa Clara-based startup may have just cut its way to the top of the field. Hound, as its app is called, is a voice-powered digital assistant. You can talk to it, ask it questions, and have it perform tasks for you. What sets Hound apart is that it's faster and more capable than anything you've ever tried before. It's available now on iOS and Android. SoundHound, the company behind Hound, launched the app in beta for Android users last summer, and it's spent the last eight months improving the service with the help of about 150,000 testers. The company is also launching partnerships with Yelp and Uber today to let Hound users get restaurant information and hail a ride from within the app. Those integrations are nice, but Hound has a tall order: it's trying to usurp the likes of Apple and Google as the go-to voice interface for smartphones. Talking to our technology is considered one of the next big leaps in computing. When software has the ability to understand what

we're saying and how we're saying it, it'll be able to parse questions and supply answers, perform tasks on our behalf, and transform how we interact with devices. So far that vision hasn't quite arrived. Apple's Siri often stumbles on simple requests, while Google Now is a personality-devoid arm of the company's search engine. Microsoft's Cortana is trying to be both clever and useful, but it's virtually nonexistent on mobile phones where we need it most. Amazon's Alexa is gaining steam in the smart home, but you can't ask it anything complex.

HOUND IS THE FIRST DIGITAL ASSISTANT THAT FEELS LIKE A REAL STEP FORWARD

Hound is the first digital assistant that feels like a real step toward the future, albeit a handicapped one at the moment. It's not that Hound feels more like you're talking to a human it's quite robotic in fact but it is without a doubt the smartest and fastest voice-based assistant I've ever seen. The app is so fast that it can produce near real-time translations of whole sentences in other languages, and it can spit back mounds of requested data faster than you could ever possibly glean it from Google with a keyboard. You have to open the app to ask it questions, which is a drag. Although the company has added 3D Touch support to its iOS version so you can jump right into a query and a "Ok Hound" voice command to make hands-free requests. The software's true appeal is understanding questions within questions and sussing out human context. You can give it sprawling, absurd requests nested inside other requests like, "What is the population and capitals of Japan and China, their area in square miles, and the population of India, and the area codes of France, Germany, and Spain?" and Hound will give you the information just seconds later. It remembers too, allowing you to try follow-up questions. Ask Hound to find you a coffee shop within walking distance that has free Wi-Fi, and you can then tell it to exclude Starbucks to narrow the search. You can ask it for hotels costing between $200 and 400 a night in Seattle near the Space Needle and when the sun will rise two days before Christmas four years from now. All of it feels hyper-specific, and Hound's default screen is a help section guiding users on what the service is even capable of. But its underlying power is undeniably impressive.

SOUNDHOUND IS KEEPING ITS SECRET SAUCE UNDER WRAPS

SoundHound CEO Keyvan Mohajer won't disclose exactly how the company's software is able to do this when Apple and Google cannot. He says the underlying technology behind Hound is built around a unique approach to natural language processing. When combined with advances in machine learning and other artificial technology techniques, Hound is able to do what Mohajer calls "speech-to-meaning." While other digital assistant software translates what you speak into text and tries to figure out what you said, Hound supposedly skips that step and deciphers your speech as it hears it. "Weve been working on it for nine years. Its not a new direction," Mohajer tells me. "It was an original ambition of the company and we knew it was going to take that long." SoundHound has been spent the better part of the last decade as a lesser-known Shazam. The company's main mobile app identifies music and differentiated itself early on by letting you hum a tune into your phone to hear the song and artist name. SoundHound's proficiency at audio recognition has helped it license out its technology to businesses. Yet all the while Mohajer says the startup has been preparing for the moment when a digital assistant could make use of modern software advances and surpass well-known competitors.

HOUND DEFAULTS TO MICROSOFT BING WHEN IT DOESN'T UNDERSTAND YOU

It's not perfect. When Hound does stumble, it does so in weird, inconsistent ways. SoundHound built its own so-called knowledge graph it doesn't use Google from which the app pulls information. Sometimes that information simply isn't there. It can tell us when President Obama's grandmother was born, but can't tell me who won the Oscar for best supporting actor in 2003. It also defaults to Microsoft's Bing for anything it doesn't understand. More often than not if it's kicking you to search it's because it either misunderstood what you were asking or it just didn't know how to answer. The goal will be for Hound to avoid doing that so often that you ditch using the app altogether. Some of those inconsistencies will undoubtedly be smoothed over with enough time. But there will always be things Hound can't do because it doesn't have direct access to your phone's software, like Apple, or intimate knowledge of your search history and email, like Google. So right now, Hound is limited to what it's best at. For the most part, that includes finding local businesses, asking oddball queries on the fly, and doing language translation. It can also tell you how much an Uber ride will cost without having to input

your pickup location and destination in the Uber app, which is a neat benefit I can see myself using all the time.

Mohajer says the company has created an easy way to develop new "domains," which is industry jargon for a specific skill set aided by a third-party API. For instance, there's a weather domain, a hotels domain provided by Expedia, and a phone domain that each let Hound take your question or request and execute it. SoundHound says it started out in beta with around 50 domains and has grown to more than 100. Mohajer says most of the company's competitors don't offer much more than two dozen simple domains.

Still, it's hard not to think SoundHound's technology could be so much more useful and widespread if it were adopted by Apple or Google, which would require an acquisition that may never happen. For now, SoundHound is intent on making its product the best alternative out there, and it hopes other companies will rally behind the product if it remains one step ahead of Siri and others. "I use Google Maps on iOS instead of Apple Maps, even though Apple Maps is more integrated," Mohajer told The Verge when Hound first launched in beta. "I think if you deliver something that is substantially better, people will use it." Hound isn't quite there yet, but it is most certainly on its way.

7 SMART-ER homes

SMART home technology use devices connected to the Internet of things (IoT) to automate and monitor in-home systems. It stands for Self-Monitoring Analysis and Reporting Technology. The technology was originally developed by IBM and was referred to as Predictive failure analysis. The first contemporary SMART home technology products became available to consumers between 1998 and the early 2000s. SMART home technology allows users to control and monitor their connected home devices from SMART home apps, smartphones, or other networked devices. Users can remotely control connected home systems whether they are home or away. This allows for more efficient energy and electric use as well as ensuring your home is secure. SMART home technology contributes to health and well-being enhancement by accommodating people with special needs, especially older people. SMART home technology is now being used to create SMART cities. A Smart city functions similar to a SMART home, where systems are monitored to more efficiently run the cities and save money.

As of 2015, the most common piece of SMART home technology in the United States were wireless speaker systems with 17 percent of people having one or more. SMART thermostats were the second most prevalent piece of SMART home technology with 11 percent of people using the device. A 2012 consumer report that pulled data from the National Association of Home Builders looked for what SMART home devices homeowners wanted most and found that top five were wireless security systems (50 percent), programmable thermostats (47 percent), security cameras (40 percent), lighting control systems and wireless home audio systems (39 percent), and home theater and multi-zone HVAC systems (37 percent).

The small actions like pointing towards a television remote can detected by the smarter home using gesture recognition. It may bring us the remote by using some new method. Also, machine learning can be used to learn about the time required to get ready for breakfast from the time of waking up. The home can on its own make breakfast matching the predicted time of the users breakfast. Many devices need to be incorporated to perform all these futuristic actions like cameras, microphones, etc.

8 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.

9 References

1. https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf1
2. https://devhub.io/repos/ActiveNick-HoloBot
3. https://www.theverge.com/2016/3/1/11136298/hound-app-ios-android-siri-google-now-cortana

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.

Parth Parikh T150074247

(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 Dialogflow

3.1 Introduction to Dialogflow

Dialogflow (formerly Api.ai, Speaktoit) is a Google-owned developer of humancomputer interaction technologies based on natural language conversations. Dialogflow has also created a natural language processing engine that incorporates conversation context like dialogue history, location and user preferences. It works on natural language processing and backed by Machine Learning .

Voice and conversational interfaces created with Dialogflow works with a wide range of devices including phones, wearables, cars, speakers and other smart devices. It supports 14+ languages including Brazilian Portuguese, Chinese, English, Dutch, French, German, Italian, Japanese, Korean, Portuguese, Russian, Spanish and Ukrainian. Dialogflow is backed by Google, incorporates Google's machine learning expertise and products such as Google Cloud Speech-to-Text and runs on Google infrastructure, which makes Dialogflow apps easily scalable to millions of users.

Dialogflow supports an array of services that are relevant to entertainment and hospitality industries. Dialogflow also includes an analytics tool that can measure the engagement or session metrics like usage patterns, latency issues, etc.

Dialogflow uses new ways to interact with your product by building engaging voice and text-based conversational interfaces, such as voice apps and chatbots, powered by AI. Dialogflow integrations can be found on websites, mobile apps, Google Assistant, Amazon Alexa, Facebook Messenger, and other popular platforms and devices. The process a Dialogflow agent follows from invocation to fulfilment is similar to someone answering a question, with some liberties taken of course.

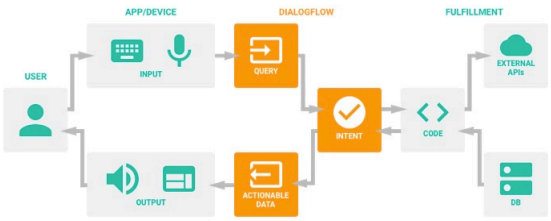
3.2 Why use Dialogflow ?

On any platform - Dialogflow supports more than 20 platforms ranging from Google Home to Twitter.

Across devices - Dialogflow supports all devices ranging from wearables to smart speakers and even cars.

Around the world - Dialogflow supports more than 14 languages worldwide and more support keeps getting added.

3.3 Glossary of Dialogflow



1. agent [Name of your app] -

   Agents are best described as NLU (Natural Language Understanding) modules. These can be included in your app, product, or service and transform natural user requests into actionable data. Agent is the name of your app you are creating. The name of the agent is very important. Here is few guidelines while picking the name: You need to invoke your agent by saying: Okay Google, talk to ¡app name¿

2. intent [conversation starter] -

   Whenever the user ask a question, it will try to match in corresponding Intent. Intent plays vital role in the assistant app. In Dialogflow, an intent houses elements and logic to parse information from the user and answer their requests. To understand the question better by intent we (developer) need to feed as much as data we can. The more variations added to the intent, the better the agent will comprehend the user. Developer need to think of different variations of same question.

3. entity [variables] -

   The Dialogflow agent needs to know what information is useful for answering the users request. These pieces of data are called entities. Assume them to be dynamic variables.

4. fulfilment [Custom code]

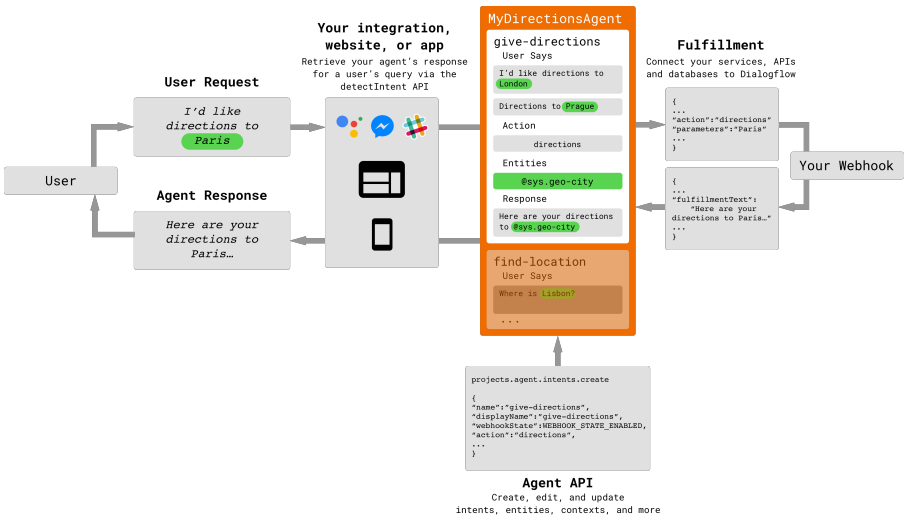
   Dialogflow backed by Google hence it works on cloud functions. When you need to add some custom code you can do it under the fulfilment tab. Fulfilment is where your custom code goes and bind your intent to cloud functions.

5. context

   Context plays vital role in the success of assistant. How? Context helps the assistant to talk more like human by maintaing the context and replying in the context to end users.

6. Platform Integration

   Dialogflow is an open platform as it not only support the integration to Assistant app it support integration to more than 20+ platform such as twitter, Facebook, Slack etc.



4 Tensorflow

4.1 Introduction to Tensorflow

TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud.TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. TensorFlow provides stable Python and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift. Third party packages are available for C#, Haskell, Julia, R, Scala, Rust, OCaml, and Crystal.

4.2 Use cases of Tensorflow

1. Voice/Sound Recognition :

   One of the most well-known uses of TensorFlow are Sound based applications. With the proper data feed, neural networks are capable of understanding audio signals. These can be :

   Voice recognition mostly used in IoT, Automotive, Security and UX/UI.

   Voice search mostly used in Telecoms, Handset Manufacturer.

   Sentiment Analysis mostly used in CRM.

   Flaw Detection (engine noise) mostly used in Automotive and Aviation.

   Regarding common use cases, we are all familiar with voice-search and voice-activated assistants with the new wide spreading smartphones such as Apples Siri, Google Now for Android and Microsoft Cortana for Windows Phone. Language understanding is another common use case for Voice Recognition. Speech-to-text applications can be used to determine snippets of sound in greater audio files, and transcribe the spoken word as text. Sound based applications also can be used in CRM. A use case scenario might be: TensorFlow algorithms standing in for customer service agents, and route customers to the relevant information they need, and faster than the agents.

2. Text Based Applications :

   Further popular uses of TensorFlow are, text based applications such as sentimental analysis (CRM, Social Media), Threat Detection (Social Media, Government) and Fraud Detection (Insurance, Finance).

   Language Detection is one of the most popular uses of text based applications. We all know Google Translate, which supports over 100 languages translating from one to another. The evolved versions can be used for many cases like translating jargon legalese in contracts into plain language.

   Google also found out that for shorter texts, summarisation can be learned with a technique called sequence-to-sequence learning. This can be used to produce headlines for news articles. Below, you can see an example where the model reads the article text and writes a suitable headline.

   Another Google use case is SmartReply. It automatically generates e-mail responses (wishing for the evolved version of this one doing our business on behalf of us).

3. Image Recognition :

   Mostly used by Social Media, Telecom and Handset Manufacturers; Face Recognition, Image Search, Motion Detection, Machine Vision and Photo Clustering can be used also in Automotive, Aviation and Healthcare Industries. Image Recognition aims to recognise and identify people and objects in images as well as understanding the content and context.

TensorFlow object recognition algorithms classify and identify arbitrary objects within larger images. This is usually used in engineering applications to identify shapes for modelling purposes (3D space construction from 2D images) and by social networks for photo tagging (Facebooks Deep Face). By analysing thousands of photos of trees for example, the technology can learn to identify a tree it has never seen before.

Image Recognition is starting to expand in the Healthcare Industry, too where TensorFlow algorithms can process more information and spot more patterns than their human counterparts. Computers are now able to review scans and spot more illnesses than humans.

4. Time Series :

   TensorFlow Time Series algorithms are used for analysing time series data in order to extract meaningful statistics. They allow forecasting non-specific time periods in addition to generate alternative versions of the time series.

   The most common use case for Time Series is Recommendation. Youve probably heard of this use from Amazon, Google, Facebook and Netflix where they analyse customer activity and compare it to the millions of other users to determine what the customer might like to purchase or watch. These recommendations are getting even smarter, for example, they offer you certain things as gifts (not for yourself) or TV shows that your family members might like.

   The other uses of TensorFlow Time Series algorithms are mainly the field of interest to Finance, Accounting, Government, Security and IoT with Risk Detections, Predictive Analysis and Enterprise/Resource Planning.

5. Video Detection :

   TensorFlow neural networks also work on video data. This is mainly used in Motion Detection, Real-Time Thread Detection in Gaming, Security, Airports and UX/UI fields. Recently, Universities are working on Large scale Video Classification datasets like YouTube-8M aiming to accelerate research on large-scale video understanding, representation learning, noisy data modelling, transfer learning, and domain adaptation approaches for video.

6. As TensorFlow is an open source library, we will see many more innovative use cases soon, which will influence one another and contribute to Machine Learning technology.

5 Apple Inc. provided APIs / Libraries

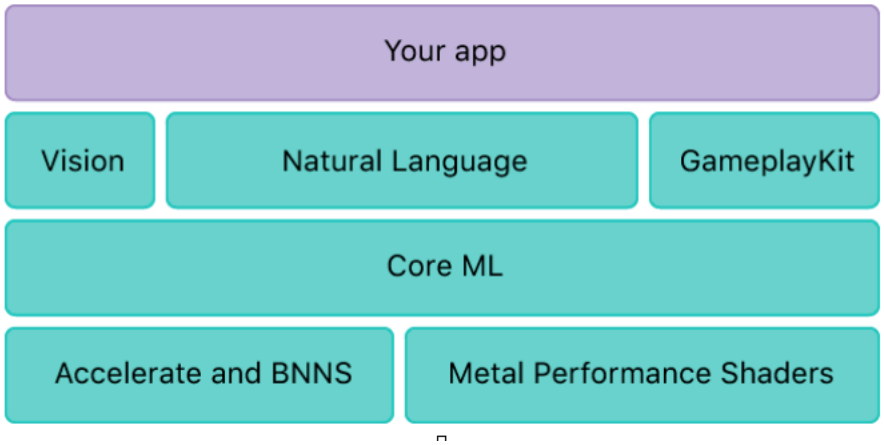
5.1 Core ML

With Core ML, you can integrate trained machine learning models into your app.

A trained model is the result of applying a machine learning algorithm to a set of training data. The model makes predictions based on new input data. For example, a model that's been trained on a region's historical house prices may be able to predict a house's price when given the number of bedrooms and bathrooms.

Core ML is the foundation for domain-specific frameworks and functionality. Core ML supports Vision for image analysis, Natural Language for natural language processing, and GameplayKit for evaluating learned decision trees. Core ML itself builds on top of low-level primitives like Accelerate and BNNS, as well as Metal Performance Shaders.

Core ML is optimised for on-device performance, which minimises memory footprint and power consumption. Running strictly on the device ensures the privacy of user data and guarantees that your app remains functional and responsive when a network connection is unavailable.



5.2 SiriKit

SiriKit enables your iOS apps and watchOS apps to work with Siri, so users can get things done using just their voice. Your content and services can be used in new scenarios, including access from the lock screen and hands-free use.

Siri Suggestions. Using signals like location, time of day and type of motion (such as walking, running or driving), Siri Suggestions learns when to suggest relevant shortcuts. Notifications. Shortcuts

appear as notifications on the lock screen, and users tap the notification to run the task.

On-device. All learning is done locally on the device, so Siri creates an intelligent, personalised experience without compromising your privacy.

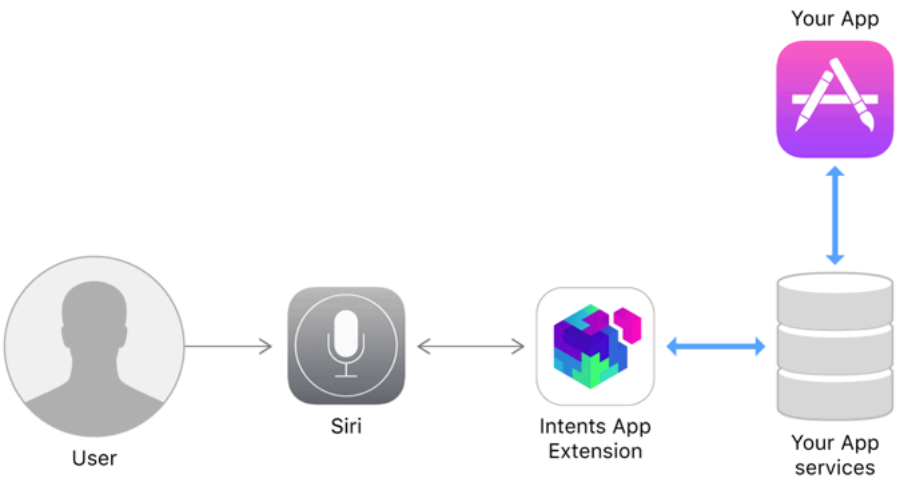
SiriKit encompasses the Intents and Intents UI frameworks, which you use to implement app extensions that integrate your services with Siri. SiriKit supports two types of app extensions :

An Intents app extension receives user requests from SiriKit and turns them into app-specific actions. For example, the user might ask Siri to send a message, book a ride, or start a workout using your app.

An Intents UI app extension displays branding or other customised content in the Siri or Maps interface after your Intents app extension fulfils a user request. Creation of this extension is optional.

SiriKit defines the types of requestsknown as intentsthat users can make. Related intents are grouped into domains to make it clear which intents you might support in your app. For example, the messages domain has intents for sending messages, searching for messages, and marking messages as read or unread.

Your app extensions rarely communicate with the user directly. Siri and Maps typically handle all communication with the user and call out to your extensions only when they need you to provide information. You can provide an Intents UI app extension to customise the information that Siri and Maps display, but doing so is optional.



6 Amazon Inc. provided APIs / Libraries

6.1 Alexa Voice Service

Amazon allows device manufacturers to integrate Alexa voice capabilities into their own connected products by using the Alexa Voice Service (AVS), a cloud-based service that provides APIs to interface with Alexa. Products built using AVS have access to Alexa's growing list of capabilities including all of the Alexa Skills. AVS provides cloud-based automatic speech recognition (ASR) and natural language understanding (NLU). There are no fees for companies looking to integrate Alexa into their products by using AVS.

he voice of Amazon Alexa is generated by a long short-term memory artificial neural network.

6.2 Alexa Skills Kit

Amazon allows developers to build and publish skills for Alexa using the Alexa Skills Kit known as Alexa Skills. These third-party developed skills, once published, are available across Alexa-enabled devices. Users can enable these skills using the Alexa app.

A "Smart Home Skill API" is available, meant to be used by hardware manufacturers to allow users to control smart home devices.

Most skills run code almost entirely in the cloud, using Amazon's AWS Lambda service. In April 2018, Amazon launched Blueprints, a tool for individuals to build skills for their personal use.

In February 2019, Amazon further expanded the capability of Blueprints by allowing customers to publish skills they've built with the templates to its Alexa Skill Store in the US for use by anyone with an Alexa-enabled device.

6.3 Amazon Lex

Amazon Lex is a service for building conversational interfaces into any application using voice and text. It powers the Amazon Alexa virtual assistant. In April 2017, the platform was released to the developer community, and suggested that it could be used for conversational interfaces (chatbots or otherwise) including Web, mobile apps, robots, toys, drones, and more. Amazon already had launched Alexa Voice Services, which developers can use to integrate Alexa into their own devices, like smart speakers, alarm clocks, etc., however Lex will not require that end users interact with the Alexa assistant per se, but rather any type of assistant or interface.

7 Introduction to AI using Python

7.1 Computer Vision using OpenCV

OpenCV (Open Source Computer Vision Library) is an open source library of programming functions for computer vision and machine learning software. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

The library has more than 2500 optimised algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognise faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognise scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.The library is cross-platform and free for use under the open-source BSD license.

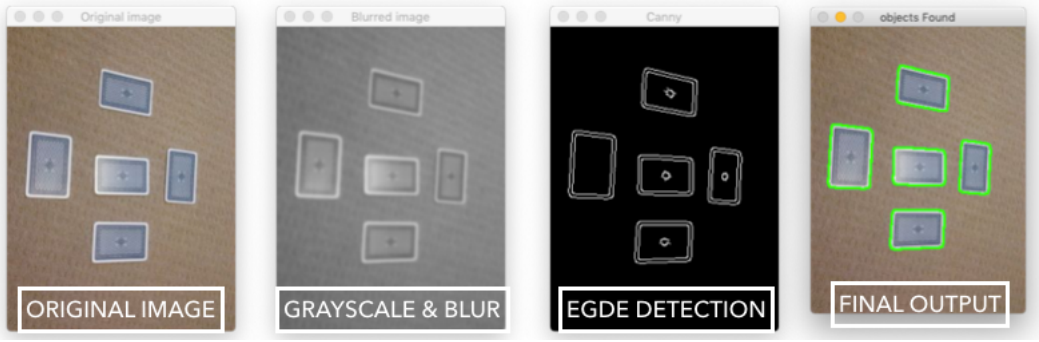
OpenCV supports the deep learning frameworks TensorFlow.

7.1.1 Python program for Edge Detection

CODE

```
import cv2
import sys

# To Read the image
image = cv2.imread("cards.jpg")

#To convert to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

#To blur the image
blurred_image = cv2.GaussianBlur(gray_image, (7,7), 0)

# Show both our images
cv2.imshow("Original image", image)
cv2.imshow("Blurred image", blurred_image)

# Run the Canny edge detector
canny = cv2.Canny(blurred_image, 30, 100)
cv2.imshow("Canny", canny)

#Finding contours
im, contours, hierarchy= cv2.findContours(canny, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

print("Number of objects found = ", len(contours))

#Highlighing the contours
cv2.drawContours(image, contours, -1, (0,255,0), 2)
cv2.imshow("objects Found", image)
cv2.waitKey(0)
```

RESULT



Two important functions in image processing are blurring and grayscale. Many image processing operations take place on grayscale images, as they are simpler to process as they have just two colours. We then detect the contours on the image and highlight to distinguish between objects.

The following functions are used to achieve the above output :

Greyscale -

The function that converts an image to greyscale is cvtColor(). The first argument is the image to be converted, the second is the colour mode. COLOR_BGR2GRAY stands for Blue Green Red to Grey. OpenCV uses the reverse of RGB colour scheme, that is it uses the BGR colour scheme in its parameters.

Blur -

The Gaussian blur is the most popular function to blur images, as it offers good blurring at fairly fast speed. The function is GaussinBlur().

The first argument is the image itself.

The second argument is the window size. Gaussian Blur works over a small window, and blurs all the pixels in that window (by averaging their values). The larger the window, the more blurring will be done, but the code will also be slower. So in the above example we chose a window of (7,7) pixels, which is a box 7 pixels long and 7 pixels wide.

Edge Detection -

The function for Canny edge detection is Canny(). It takes three arguments.

The first is the image.

The second and third are the lower and upper thresholds respectively.

The Canny edge detector detects edges by looking in the difference of pixel intensities.

So for the example above, Im using low thresholds of 30, 100, which means a lot of thresholds will be detected.

Find Contours -

The findContours() finds the contours in the given image. The first option is the output of the canny edge detector. RETR_EXTERNAL tells OpenCv to only find the outermost edges (as you can find contours within contours). The second arguments tells OpenCv to use the simple approximation.

The function returns three values: The image, a list of contours found, and the hierarchy (which can be ignored as it is used if you have many contours embedded within others).

The contours return value is a simple list that contains the number of contours found. Taking the length of it will give us number of objects found.

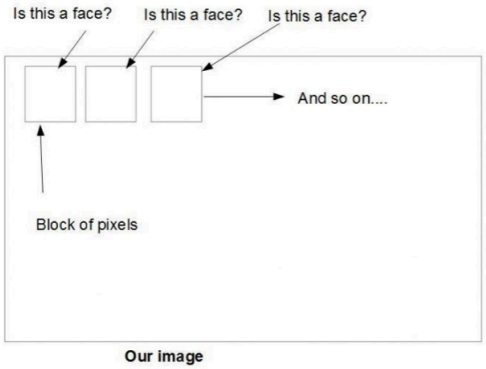
Draw Contours -

Finally, we use the drawContours() function. The first argument is the image we want to draw on. The second is the contours we found in the last function. The 3rd is -1, to say that we want all contours to be drawn (we can choose to only draw certain contours). The fourth is the colour, green in this case, and the last is the thickness.

7.1.2 Python program for Face Detection

OpenCV uses machine learning algorithms to search for faces within a picture. For something as complicated as a face, there isnt one simple test that will tell us if it found a face or not. Instead, there are thousands of small patterns/features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve. These tasks are also called classifiers.

For something like a face, we may have 6,000 or more classifiers, all of which must match for a face to be detected (within error limits, of course). But therein lies the problem: For face detection, the algorithm starts at the top left of a picture and moves down across small blocks of data, looking at each block, constantly asking, Is this a face? Is this a face? Is this a face? Since there are 6,000 or more tests per block, we might have millions of calculations to do, which will grind our computer to a halt.



To get around this, OpenCV uses cascades. Whats a cascade? The best answer can be found from the dictionary: A waterfall or series of waterfalls.

Like a series of waterfalls, the OpenCV cascade breaks the problem of detecting faces into multiple stages. For each block, it does a very rough and quick test. If that passes, it does a slightly more detailed test, and so on.

Since face detection is such a common case, OpenCV comes with a number of built-in cascades for detecting everything from faces to eyes to hands and legs.

CODE

```
import sys
import cv2


imgpath = sys.argv[1]
cascasdepath = "haarcascade_frontalface_default.xml"

image = cv2.imread(imgpath)
```

```
#image = cv2.imread('abba.png')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

face_cascade = cv2.CascadeClassifier(cascasdepath)

faces = face_cascade.detectMultiScale(
    gray,
    scaleFactor = 1.2,
    minNeighbors = 5,
    minSize = (30,30)

    )

print("The number of faces found = ", len(faces))

for (x,y,w,h) in faces:
    cv2.rectangle(image, (x,y), (x+h, y+h), (0, 255, 0), 2)

cv2.imshow("Faces found", image)
cv2.waitKey(0)
```

RESULT

The detectMultiScale() function is a general function that detects objects. Since it is called on the face cascade, thats what it detects. Its parameters are :

The first parameter is the grayscale image.

The second is the scaleFactor. Since some faces may be closer to the camera, they would appear bigger than those faces in the back. The scale factor compensates for this.

The detection algorithm uses a moving window to detect objects. minNeighbors defines how many objects are detected near the current one before it declares the face found. minSize, meanwhile, gives the size of each window.

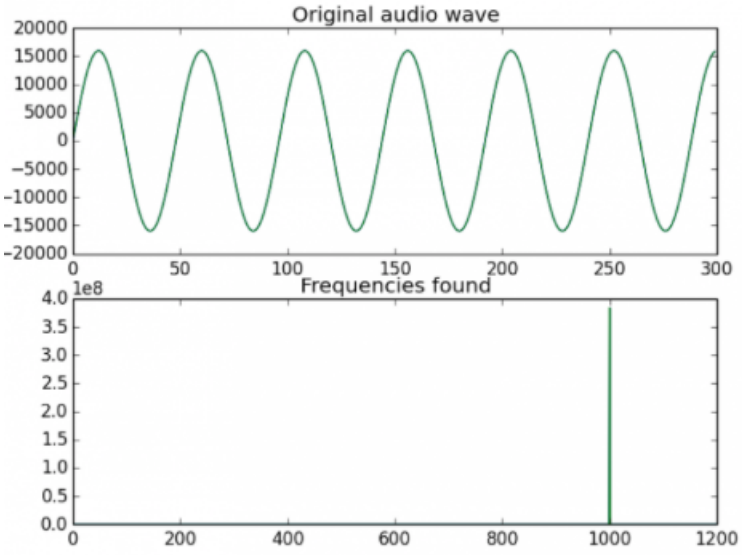
The function returns a list of rectangles where it believes it found a face.

7.2 Audio and Digital Signal Processing

CODE

```
import numpy as np
import wave
import struct
import matplotlib.pyplot as plt

frame_rate = 48000.0
infile = "test.wav"
num_samples = 48000

wav_file = wave.open(infile, 'r')
data = wav_file.readframes(num_samples)
wav_file.close()

data = struct.unpack('{n}h'.format(n=num_samples), data)
data = np.array(data)

data_fft = np.fft.fft(data)

# This will give us the frequency we want
frequencies = np.abs(data_fft)
print("The frequency is {} Hz".format(np.argmax(frequencies)))

plt.subplot(2,1,1)
plt.plot(data[:300])
plt.title("Original audio wave")
plt.subplot(2,1,2)
plt.plot(frequencies)
plt.title("Frequencies found")

plt.xlim(0,1200)
plt.savefig('wave.png').

plt.show()
```

RESULT



To get the frequency of a sine wave, we need to get its Discrete Fourier Transform(DFT). In its simplest terms, the DFT takes a signal and calculates which frequencies are present in it. In more technical terms, the DFT converts a time domain signal to a frequency domain. For example, lets look at our sine wave. The wave is changing with time. If this was an audio file, we could imagine the player moving right as the file plays. In the frequency domain, we see the frequency part of the signal.The signal will change if we add or remove frequencies, but will not change in time. For example, if we take a 1000 Hz audio tone and take its frequency, the frequency will remain the same no matter how long you look at it. But if you look at it in the time domain, we will see the signal moving.

The DFT was really slow to run on computers, so the Fast Fourier Transform (FFT) was invented. The FFT is what is normally used nowadays.

Here, we are reading a sample wave file. The wave readframes() function reads all the audio frames from a wave file.

We take the fft of the data. This will create an array with all the frequencies present in the signal.

np.argmax will return the highest frequency in our signal, which it will then print. As we have seen manually, this is at a $1000Hz$ (or the value stored at data_fft[1000]). And now we can plot the data too.

The function subplot(2,1,1) means that we are plotting a 2x1 grid. The 3rd number is the plot number, and the only one that will change.

7.3 Natural Language Processing using NLTK

7.3.1 Python program using NLTK

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning with wrappers for industrial-strength NLP libraries.

PunktSentenceTokenizer is an sentence boundary detection algorithm that must be trained to be used. NLTK already includes a pre-trained version of the PunktSentenceTokenizer.

This tokenizer divides a text into a list of sentences by using an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences. It must be trained on a large collection of plaintext in the target language before it can be used.

This approach has been shown to work well for many European languages.

CODE

```
#Using the nltk library for natural language processing in python
import nltk

from nltk.tokenize import PunktSentenceTokenizer

#Sample sentence
document = 'Whether you\'re new to programming or an experienced developer, it\'s easy to
    learn and use Python.'

sentences = nltk.sent_tokenize(document)
for sent in sentences:
    print(nltk.pos_tag(nltk.word_tokenize(sent)))
```

RESULT

```
Terminal File Edit View Search Terminal Help
$ python speechTagging.py
[('Whether', 'IN'), ('you', 'PRP'), ("'re", 'VBP'), ('new', 'JJ'), ('to', 'TO'), ('programming'
, 'VBG'), ('or', 'CC'), ('an', 'DT'), ('experienced', 'JJ'), ('developer', 'NN'), (',', ','), (
'it', 'PRP'), ("'s", 'VBZ'), ('easy', 'JJ'), ('to', 'TO'), ('learn', 'VB'), ('and', 'CC'), ('us
e', 'VB'), ('Python', 'NNP'), ('.', '.')]
$
```

KEY :

CC	Coordinating conjunction	NNS	Noun, plural	UH	Interjection
CD	Cardinal number	NNP	Proper noun, singular	VB	Verb, base form
DT	Determiner	NNPS	Proper noun, plural	VBD	Verb, past tense
EX	Existential there	PDT	Predeterminer	VBG	Verb, gerund or present participle
FW	Foreign word	POS	Possessive ending	VBN	Verb, past participle
IN	Preposition or subordinating conjunction	PRP	Personal pronoun	VBP	Verb, non-3rd person singular present
JJ	Adjective	PRP$	Possessive pronoun	VBZ	Verb, 3rd person singular present
JJR	Adjective, comparative	RB	Adverb	WDT	Wh-determiner
JJS	Adjective, superlative	RBR	Adverb, comparative	WP	Wh-pronoun
LS	List item marker	RBS	Adverb, superlative	WP$	Possessive wh-pronoun
MD	Modal	RP	Particle	WRB	Wh-adverb
NN	Noun, singular or mass	SYM	Symbol		
		TO	to		

8 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.

9 References

1. https://dialogflow.com/docs/
2. https://www.tensorflow.org/tutorials/
3. https://developer.apple.com/documentation/coreml/
4. https://developer.apple.com/documentation/sirikit
5. https://developer.amazon.com/alexa-skills-kit/
6. https://developer.amazon.com/alexa-voice-service/
7. https://docs.aws.amazon.com/lex/index.html/
8. https://www.makeuseof.com/tag/diy-google-home-assistant-raspberry-pi/
9. https://www.nltk.org
10. https://www.pythonforengineers.com/articles/