GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

VOICE-BASED SEARCH ASSISTANT

(USING N-GRAM MODEL)

A Report of Capstone Project 2

Submitted by

MRINAL SHUBHAM

(1613112031 / 16SCSE112021)

in partial fulfilment for the award of the degree

of

Bachelor of Technology

IN

Computer Science and Engineering with Specialization of Data Analytics

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Dr. SAMPATH KUMAR K, M.Tech., Ph.D.,

Professor

APRIL / MAY- 2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"VOICE-BASED SEARCH ASSISTANT (USING N-GRAM MODEL)"** is the bonafide work of **"MRINAL SHUBHAM (1613112031)"** who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL

Ph.D (Management), Ph.D (CS)

Professor & Dean,

School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR

Dr. SAMPATH KUMAR K

M.Tech (CS), Ph.D (CS)

Professor,

School of Computing Science & Engineering

Abstract

Voice search majorly permits users to talk into a tool that's equipped with voice technology (Voice Assistants) as against writing keywords into an exploration question to get the desired results. Voice technology uses speech recognition to know what users are saying with high accuracy. It then provides results orally to the user. Though it would look like a brand-new construct however voice search has been around for quite a while. Programs like voice to text and quick voice dialing are nice samples of voice search. In addition, voice assistants like Google Assistant, Siri, Microsoft Cortana, and Amazon Alexa all use voice search capabilities. In fact, nowadays we've reached a stage where more than just devices can be optimized for voice search. Brands, platforms, and websites can also be optimized for voice search. Voice recognition is classed according to what style of utterance they have the ability to recognize.

Voice recognition are classified according to what type of utterance they have ability to recognize. They are classified as:

Continuous Speech: When user speak in a more normal, fluid manner without having to pause between word, which is referred as Continuous Voice.

Discrete Speech: When user speak with taking rest between each word then such Voice is referred as discrete Voice.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER -1

Introduction

Speech is the basic, common and efficient form of communication method for people to interact with each other. Today Voice technologies are commonly available for a limited but interesting range of task. This technology enables machines to respond correctly and reliably to human voices and provide useful and valuable services. As communicating with computer is faster using voice rather than using keyboard, so people will prefer such system. Communication among the human being is dominated by spoken language, therefore it is natural for people to expect voice interfaces with computer.

This can be accomplished by developing voice recognition system: Voice-to-text which allows computer to translate voice request and dictation into text. Voice recognition system: Voice-to text is the process of converting an acoustic signal which is captured using a microphone to a set of words. The recorded data can be used for document preparation.

Voice Recognition (SR) is the ability to translate a dictation or spoken word to text. Voice Recognition known as “automatic Voice recognition “(ASR), or Voice to text (STT)

- Voice recognition is the process of converting an acoustic signal, captured by a microphone or any peripherals, to a set of words.
- To achieve Voice understanding we can use linguistic processing

The recognized words can be an end in themselves, as for applications such as commands & control data entry and document preparation. In the society every one either human or animals wish to interact with each other and tries to convey own message to others. The receiver for messages may get the exact and full idea of the senders, or may get the partial idea or sometimes cannot understand anything out of it. In some cases, may happen when there is some lacking in

- communication (i.e. when a child convey message, the mother can understand easily while others cannot)

A. Types of Voice utterance

Voice recognition are classified according to what type of utterance they have ability to recognize. They are classified as:

1.Isolated word: Isolated word recognizer usually requires each spoken word to have quiet (lack of an audio signal) on bot h side of the sample window. It accepts single word at a time.

2.Connected word: It is similar to isolated word, but it allows separate utterances to „run-together" which contains a minimum pause in between them.

3.Continuous Voice: it allows the users to speak naturally and in parallel the computer will determine the content.

4.Spontaneous Voice: It is the type of Voice which is natural sounding and is not rehearsed.

B. Types of speaker model

Voice recognition system is broadly into two main categories based on speaker models namely speaker dependent and speaker independent.

1) Speaker dependent models: These systems are designed for a specific speaker. They are easier to develop and more accurate but they are not so flexible.

2) Speaker independent models: These systems are designed for variety of speaker. These systems are difficult to develop and less accurate but they are very much flexible.

Purpose

Nearly 20% people of the world ae suffering from various disabilities; many of them are blind or unable to use their hands effectively. The Voice recognition system in those particular cases provide a significant help to them, so that they can share information with people by operating computer through voice input. Consider the Thousands of people in world they are not able to use their hands making typing impossible. our project it for these people who can't type, and see, even for those of us who are lazy and don't feel like it Our project is capable to recognize the Voice and convert the input audio into text; it also enables a user to perform operations such as searching any information from any search bar. Voice Assistant often began its answer by sourcing the website first, followed by one of their accurate and concise snippets. For example, when asked "how do you get wine stains out of a rug," Google Home began its response by sourcing the featured snippet that is the first results, starting with "According to Patch…" and then completed its one-sentence answer. While snippets can provide more traffic and credibility to websites that are sourced in Google's answer, and generally make answering our basic questions much easier, they can also hurt the click-rate of other websites by giving someone a reason to not visit a website they may have otherwise. With how Google Home sources its snippets and given their

history of products and services growing into and altering the majority, it would be reasonable to assume that this will become even more prevalent in the near future.

CHAPTER -2

Existing System

The current challenges of Voice recognition are caused by two major factors – reach and loud environments. This calls for even more precise systems that can tackle the most ambitious

Automatic Voice Recognition use-cases. Think about live interviews, Voice recognition at a loud family dinner or meetings with various people. These are the upcoming challenges to be solved for next-gen voice recognition.

Beyond this, Voice recognition needs to be made available for more languages and cover wide topics. Because as of now, ASR needs a lot of data to work well and some of it just hasn't been collected for certain languages and topics. Without adding these, ASR systems will remain noticeably handicapped.

The field of **Speech recognition** adds another dimension to the **problem** of natural language understanding. The **problems** include the background noise, changes in pronunciation according to whether words are spoken in isolation or in sentence and in variations of accent between individuals.

Automated search assistant has in years has become a practical concept, which is now being implemented in different languages around the world. Voice recognition has been used in real-world human language applications, such as information recovery. Voice in human can be said as the most common means of the communication because the information maintains the basic role in conversation. The conversation or Voice that is captured by a microphone or a telephone is converted from acoustic signal to a set of words in Voice recognition. The literature survey for research was done by referring to the journal papers, conference papers, articles, books, internet and databases. Overall, this chapter describes a review of Voice recognition task, Voice recognition approaches, current Voice recognition system, Tamil Voice recognition system as well as different type of methods applied to Voice recognition system. Based on the review of the advantages and disadvantages, this thesis discusses the most suitable techniques and methods to develop a Voice recognition system.

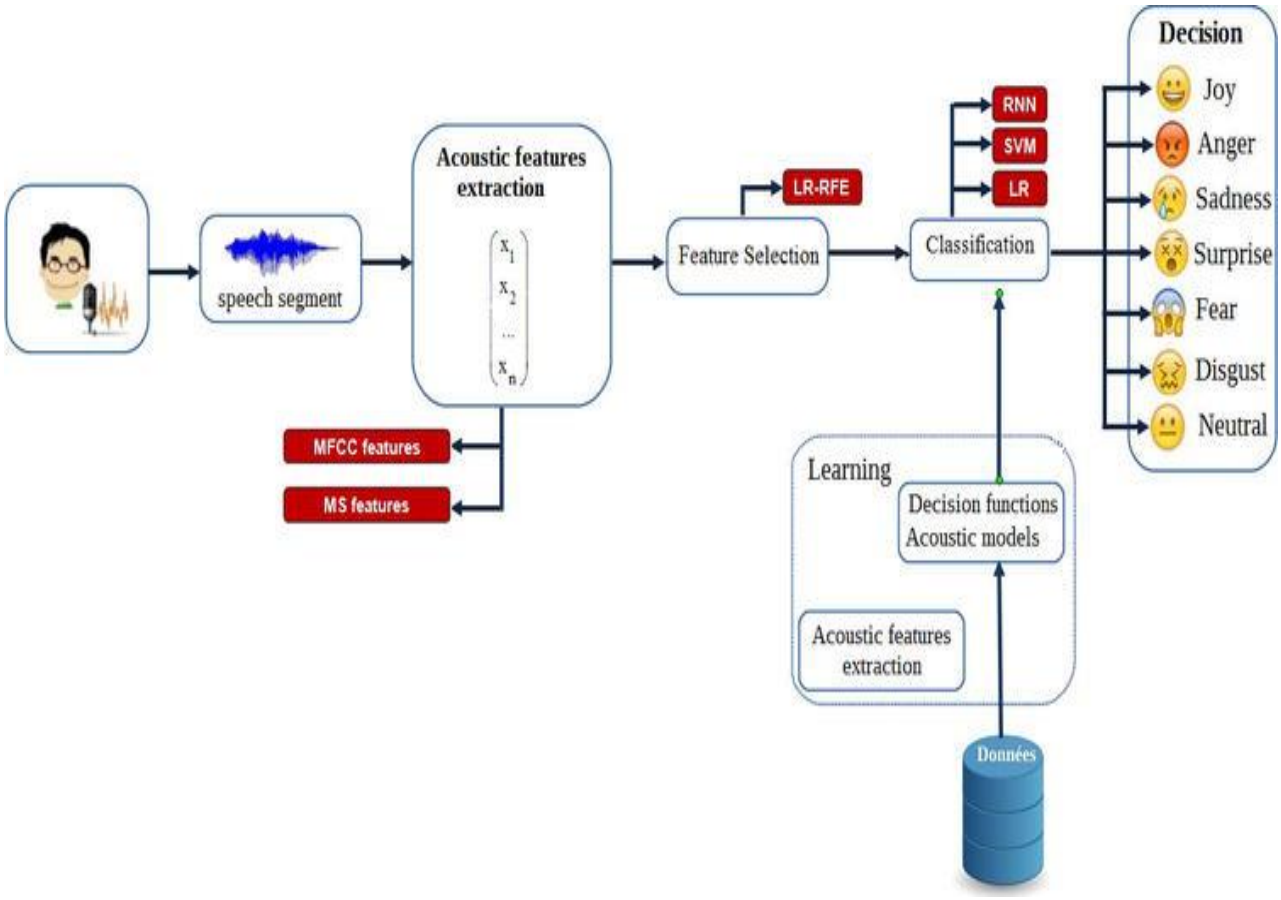


Fig- 1. Voice Recognition model

CHAPTER - 3

Proposed System

A handful of packages for Voice recognition exist on PyPI. A few of them include:

- apiai
- assemblyai
- google-cloud-speech
- pocketsphinx
- Voice_Recognition
- Watson-developer-cloud
- wit

Some of these packages—such as wit and apiai—offer built-in features, like natural language processing for identifying a speaker's intent, which go beyond basic Voice recognition. Others, like google-cloud-Voice, focus solely on Voice-to-text conversion. The Voice Recognition library acts as a wrapper for several popular Voice APIs and is thus extremely flexible. One of these—the Google Web Voice API—supports a default API key that is hard-coded into the Voice Recognition library. That means you can get off your feet without having to sign up for a service.

Voice Recognition makes working with audio files easy thanks to its handy AudioFile class. This class can be initialized with the path to an audio file and provides a context manager interface for reading and working with the file's contents.

Due to the rapid consumer adoption of virtual assistants, the dynamic of search queries is also changing. Voice search's natural language syntax lends itself to longer tail keywords.

Here are some examples that illustrate how voice search incorporates longer queries that more distinctly identify user intent:

- **Traditional Text Search Query:** This search may include someone typing in "Chinese food in Houston" or "Chinese food delivery."
- **Voice Search Query:** This search might involve something more elaborate and in a conversational tone, such as: "Where's the closest Chinese food restaurant?" or "Find Chinese food delivery open now."

Take Alexa, for instance: this virtual assistant will highlight the top search result that best matches your search query. These results can also be repeated so that a user does not have to peer at a screen.

While the voice search method makes it easier for the user, marketers need to realize that if their content happens to be in position number two, number three or lower in search results, the user will likely never see it. In other words, it makes ranking for the most-coveted #1 search result position even more valuable for ultimate success.

Speech recognition software can analyze the sounds you make by filtering what you say, digitizing it to a format it can "read", and then analyzing it for meaning. Then, based on algorithms and previous input, it can make a highly accurate educated guess as to what you are saying. It gets to know the speaker's use of language.

Unsurprisingly, if the speech recognition software is only used by one person, it will be trained specifically for how that person talks. It becomes increasingly more complex when a device or software is geared towards multiple different markets around the world. This is because engineers have to program the ability to understand infinite more variations; language, dialects, accents, phrasing. But the complexities don't stop there.

Even with hundreds of hours of input, other factors can play a huge role in whether or not the software can understand you:

Background noise can easily throw a speech recognition device off track. This is because it does not inherently have the ability to distinguish the ambient

sounds it “hears” of a dog barking or a helicopter flying overhead, from your voice.

Engineers have to program that ability into the device; they conduct data collection of these ambient sounds and “tell” the device to filter them out.

Another factor is the way humans naturally shift the pitch of their voice to accommodate for noisy environments; speech recognition systems can be sensitive to these pitch changes.

Shazam, an app that is used to instantly identify music, is another great example of how speech recognition technology works.

When you hit the Shazam button, you are effectively starting an audio recording of your surroundings.

The app differentiates the ambient noise, identifies the song’s pattern, and compares the audio recording to its database.

Eventually, tracking down the song that was playing and supplying the information to its curious end-user.

In much the same way, your voice is recognized as the input.

The device or software then separates the noise (individualistic vocal patterns, accents, ambient sounds, and so on) from the keywords and turns it into text that the software can understand.

This is why speech recognition technology developed in North America for the North American accent does not work well when foreigners attempt to use it; native speakers pronounce things more or less consistently – save for individual variety.

Whereas, foreigners speaking English with an accent introduce irregular intonations and phrasing.

SDSs are often chronologically categorized into three generations– informational, transactional and problem solving. The first generation SDSs focus on providing users with the information they request, such as flight status and weather information. The second generation SDSs conduct transactions automatically with users, e.g.,

to book air flight tickets or perform bank balance transfer. The third generation SDSs are often used in customer support by interacting with callers to diagnose the problems they are experiencing with a device or a service.

The chronological (functional) categorization of SDSs does not necessarily imply the level of technological difficulties. Some of the problems in informational SDSs remain the most challenging topics in spoken dialog research. To better understand different technological challenges, SDSs can be categorized technologically into three categories – form filling, call

routing and voice search. Form-filling is the most commonly used technology deployed in the first and the second generation SDSs, where directed or mixed-initiative dialog systems are used to gather the attribute values of an entity that users are interested in (e.g., the originating and destination cities of a flight). In such systems, users often have to use canned expressions within a small domain. In a directed dialog system, user's utterances may contain only what the system has prompted for, which is often a single piece of semantic information; while in a mixed-initiative system, users may volunteer

more semantic information in a single utterance – we call this type of semantic understanding high-resolution in the sense that multiple semantic constituents (commonly called "slots') need to be identified. The call-routing applications remove the constraints on what users can say, so users can speak naturally. This is accomplished at the expense of limiting the target semantic space: the understanding of natural language inputs is often achieved with statistical classifiers, which map users' inputs to a list of possible destination classes (intents). The classifiers can hardly perform high resolution understanding with many slots, or scale up with a huge number (e.g., thousands to millions) of destination classes. Voice search applications differ from the form-filling applications in their lack of detailed, high resolution semantic analysis.

They are similar to call-routing applications with respect to the naturalness of user inputs and the huge input space. However, they differ from call-routing applications in the sense that their semantic space, or in the terminology of call-routing systems, the inventory of the "destination classes" is enormous – sometimes in the range of millions of entries. Data are seldom sufficient to train a statistical classifier. The table below compares the three types of technologies.

	User input utterances		Target semantic representation	
	Naturalness	Input space	Resolution	Semantic space
Form filling/directed dialog	low	small	low	small
Form filling/mixed-initiative	Low-medium	small	high	small
Call routing	high	large	low	small
Voice search	Medium-high	large	low	medium-large

Table- 1. Types of technology

Natural Language Processing or NLP is a field of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages.

Use Cases of NLP

In simple terms, NLP represents the automatic handling of natural human language like speech or text, and although the concept itself is fascinating, the real value behind this technology comes from the use cases.

NLP can help you with lots of tasks and the fields of application just seem to increase on a daily basis. Let's mention some examples:

- NLP enables the recognition and **prediction of diseases** based on electronic health records and patient's own speech. This capability is being explored in health conditions that go from cardiovascular diseases to depression and even schizophrenia. For example, Amazon Comprehend Medical is a service that uses NLP to extract disease conditions, medications and treatment outcomes from patient notes, clinical trial reports and other electronic health records.

- Organizations can determine what customers are saying about a service or product by identifying and extracting information in sources like social media. This **sentiment analysis** can provide a lot of information about customers choices and their decision drivers.

- An inventor at IBM developed a **cognitive assistant** that works like a personalized search engine by learning all about you and then remind you of a name, a song, or anything you can't remember the moment you need it to.

- Companies like Yahoo and Google filter and classify your emails with NLP by analyzing text in emails that flow through their servers and **stopping spam** before they even enter your inbox.

- To help **identifying fake news**, the NLP Group at MIT developed a new system to determine if a source is accurate or politically biased, detecting if a news source can be trusted or not.

- Amazon's Alexa and Apple's Siri are examples of intelligent **voice driven interfaces** that use NLP to respond to vocal prompts and do everything like find a particular shop, tell us the weather forecast, suggest the best route to the office or turn on the lights at home.
- Having an insight into what is happening and what people are talking about can be very valuable to **financial traders**. NLP is being used to track news, reports, comments about possible mergers between companies, everything can be then incorporated into a trading algorithm to generate massive profits. Remember: buy the rumor, sell the news.
- NLP is also being used in both the search and selection phases of **talent recruitment**, identifying the skills of potential hires and also spotting prospects before they become active on the job market.
- Powered by IBM Watson NLP technology, LegalMation developed a platform to automate routine **litigation tasks** and help legal teams save time, drive down costs and shift strategic focus.

CHAPTER - 4

Software Requirement

The software used for the development of the project is:

Operating system: Windows

Programming Language: Python (3.6)

IDE: Anaconda Navigator (Jupiter)

Package: Voice Recognition Library

Hardware components: Network communication Modem for connecting to internet, Connecting Wires.

Component	Minimum	Recommended
CPU	1.6 GHz	2.53GHz
RAM	2GB	4GB
Microphone	Mic	High quality
Sound card	Sound card	Sound card with very clear signal

Table-2. Hardware Components

Implementation

Models that assign probabilities to sequences of words are called language models or LMs. In this post is show you the simplest model that assigns probabilities to sequences of words, the n-gram. An n-gram is a sequence of N words: a 2-gram (or bigram) is a two-word sequence of words like “This is”, “is a”, “a great”, or “great song” and a 3-gram (or trigram) is a three-word sequence of words like “is a great”, or “a great song”. We’ll see how to use n-gram models to predict the last word of an n-gram given the previous words and thus to create new sequences of words. In a bit of terminological ambiguity, we usually drop the word “model”, and thus the term n-gram is used to mean either the word sequence itself or the predictive model that assigns it a probability.

Types of n-gram:

1. Uni-gram
2. Bi-gram
3. Tri-gram
4. N-gram (N denotes the word that needs to be predicted).

Python Libraries used:

1. Numpy
2. Pandas
3. Spacy (To tokenize the words)
4. Itertools (‘tee’: function is used to iterate the words)

Each Recognizer instance has seven methods for recognizing Voice from an audio source using various APIs. These are:

- Recognize_bing(): Microsoft Bing Speech
- recognize_google(): Google Web Voice API
- recognize_google_cloud(): Google Cloud Speech - requires installation of the googlecloudVoice package
- recognize_houndify(): Humidify by SoundHound
- recognize_ibm(): IBM Voice to Text
- recognize_sphinx(): CMU Sphinx - requires installing PocketSphinx
- recognize_wit(): Wit.ai

Of the seven, only recognize_sphinx() works offline with the CMU Sphinx engine. The other six all require an internet connection.A full discussion of

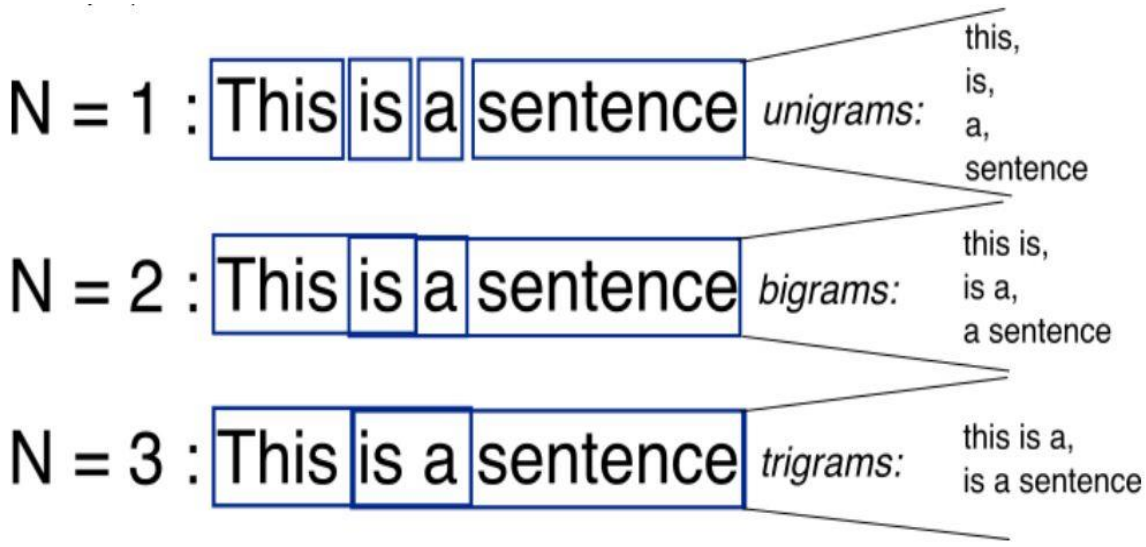


Fig-2. N-gram model

the features and benefits of each API is beyond the scope of this tutorial. Since VoiceRecognition ships with a default API key for the Google Web Voice API, you can get started with it right away. For this reason, we'll use the Web Voice API in this guide. The other six APIs all require authentication with either an API key or a username password combination. For more information, consult the Voice Recognition docs.

Before you continue, we'll need to download an audio file. The one I used to get started, "harvard.wav," can be found here. Make sure you save it to the same directory in which your Python interpreter session is running.

Voice Recognition makes working with audio files easy thanks to its handy AudioFile class. This class can be initialized with the path to an audio file and

provides a context manager interface for reading and working with the file’s contents.

Before we go and actually implement the N-Grams model, let us first discuss the drawback of the bag of words and TF-IDF approaches.

In the bag of words and TF-IDF approach, words are treated individually and every single word is converted into its numeric counterpart. The context information of the word is not retained. Consider two sentences "big red machine and carpet" and "big red carpet and machine". If you use a bag of words approach, you will get the same vectors for these two sentences. However, we can clearly see that in the first sentence we are talking about a "big red machine", while the second sentence contains information about the "big red carpet". Hence, context information is very important. The N-Grams model basically helps us capture the context information.

CHAPTER-5

Code & Result

N-Grams in Python:

```
import nltk
import numpy as np
import random
import string

import bs4 as bs
import urllib.request
import re
```

```
raw_html = urllib.request.urlopen('https://en.wikipedia.org/wiki/Tennis')
raw_html = raw_html.read()

article_html = bs.BeautifulSoup(raw_html, 'lxml')
article_paragraphs = article_html.find_all('p')
article_text = ''

for para in article_paragraphs:
    article_text += para.text

article_text = article_text.lower()
```

```
article_text = re.sub(r'[^A-Za-z. ]', '', article_text)
```

```
ngrams = {}
chars = 3

for i in range(len(article_text)-chars):
    seq = article_text[i:i+chars]
    print(seq)
    if seq not in ngrams.keys():
        ngrams[seq] = []
    ngrams[seq].append(article_text[i+chars])
```

```
curr_sequence = article_text[0:chars]
output = curr_sequence
for i in range(200):
    if curr_sequence not in ngrams.keys():
        break
    possible_chars = ngrams[curr_sequence]
    next_char = possible_chars[random.randrange(len(possible_chars))]
    output += next_char
    curr_sequence = output[len(output)-chars:len(output)]

print(output)
```

Output:

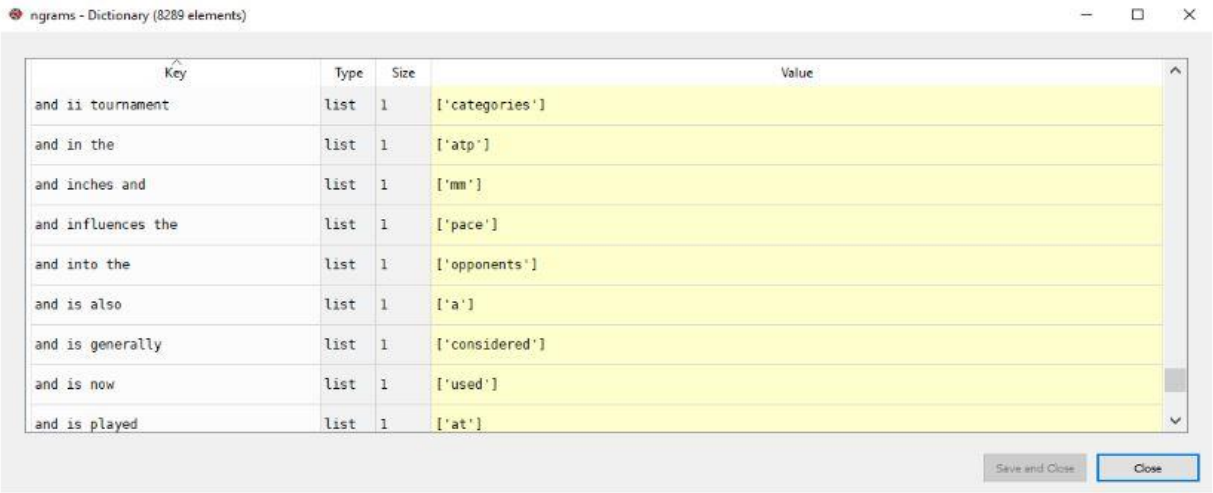
```
tent pointo somensiver tournamedal pare the greak in the next peak sweder most begal tennis sport. the be has si
ders with sidernaments as was that adming up is coach rackhanced ball of ment. a game and
```

```
tennis ahead with the club players under.most coaching motion us . the especific at the hit and events first pre
domination but of ends on the u.s. cyclops have achieved the end or net inches call over age
```

Words N-Grams Model:

```
ngrams = {}
words = 3

words_tokens = nltk.word_tokenize(article_text)
for i in range(len(words_tokens)-words):
    seq = ' '.join(words_tokens[i:i+words])
    print(seq)
    if  seq not in ngrams.keys():
        ngrams[seq] = []
    ngrams[seq].append(words_tokens[i+words])
```




```
curr_sequence = ' '.join(words_tokens[0:words])
output = curr_sequence
for i in range(50):
    if curr_sequence not in ngrams.keys():
        break
    possible_words = ngrams[curr_sequence]
    next_word = possible_words[random.randrange(len(possible_words))]
    output += ' ' + next_word
    seq_words = nltk.word_tokenize(output)
    curr_sequence = ' '.join(seq_words[len(seq_words)-words:len(seq_words)])

print(output)
```

Output:

```
tennis is a racket sport that can be played individually against a single opponent singles or between two teams of two players each doubles. each player uses a tennis racket include a handle known as the grip connected to a neck which joins a roughly elliptical frame that holds a matrix of
```

```
tennis is a racket sport that can be played individually against a single opponent singles or between two teams of two players each doubles . each player uses a tennis racket that is strung with cord to strike a hollow rubbe r ball covered with felt over or around a net and into the opponents
```

CHAPTER-6

Conclusion & Discussion

Voice search poses new challenges to the spoken dialog technology in the following areas:

Speech Recognition: The state-of-the-art ASR systems have high error rates on voice search tasks. The vocabulary size of a voice search system can be much larger than a typical form-filling or a call routing application – sometimes reaching millions of lexical entries. Many lexical entries in international individual/ business names are out of vocabulary and lack reliable pronunciation information. Calls are often made from

different noisy environments. In addition, the constraints from language models are often weaker than other ASR tasks – the perplexity of a language model is often high (e.g., 400~500 bits for business DA) for voice search.

Spoken language Understanding (SLU)/Search: One big problem in SLU is the enormous semantic space – a DA system can easily contain hundreds of thousands (if not millions) of listings in a city. There is also a high level of linguistic variance in the input space. For example, users may not use the official name of a business in a DA or business rating system. They would

typically say, for instance, “Sears” instead of the listed official name, “Sears Noise Related Normal ASR Pronunciation Spelling/Chopped Speech Roebuck & Co.” In addition, the SLU/search component must be robust to ASR errors.

Dialog Management: The difficulties in ASR and SLU cause much confusability and uncertainty. Dialog manager has to effectively narrow down the scope of what a user may say to reduce the confusability and uncertainty. Search results often contain multiple entries. Disambiguation strategy is crucial in obtaining sufficient information for the correct understanding of users’ intents with as few dialog turns as possible. Confidence measures are important for the dialog manager to take an appropriate action with each of the hypothesized interpretations, such that the dialog can recover gracefully from ASR and SLU errors.

Feedback loop: No systems can be perfectly built at the initial deployment. Dialog system tuning is often performed painstakingly by spoken dialog experts, starting from error analysis from the logged interaction data to find the flaws in dialog and prompt design, grammar coverage, system implementation, etc. An interesting research topic is the automatic or semi-automatic discovery and remedy of design/implementation flaws.

The grand challenge in voice search application is robustness. The CSELT's study on Telecom Italia's DA system showed that even though the automation rate was 92% in a laboratory study, the actual field trial automation rate was only 30% due to unexpected behaviour of novice SDS users and environment noise.

This section reviews the technology that addresses the challenges to voice search applications. Not surprisingly, much of the technology is developed with DA systems because they are the most popular voice search applications so far. However, the technology is often applicable to other applications as well. For example, the product/business rating systems directly used the technology developed in a DA application.

Speech Recognition

A detailed error analysis for proper name recognition was reported in an auto-attendant system [14]. Figure 2 shows the distribution of different causes of errors – besides 35% Figure 2. ASR error analysis for a voice search application. Of normal recognition errors, 31% were noise related and 22% were pronunciation related. Many of the calls were made in a noisy environment over different noise channels. Therefore, noise-robustness is crucial to improve the ASR accuracy. On the other hand, there were many foreign names that are difficult to pronounce in an auto-attendant/DA system. In fact, pronunciation is

a pervasive problem that poses challenge in many other voice search applications too. For example, users may specify “Petit Bonheur by Salvatore Adamo” in music search. Hence pronunciation modelling is another important topic in ASR for voice search. In addition, better acoustic and language models are always important to reduce the ASR error rate. Acoustic Modelling: IBM’s auto-attendant system applied speaker clustering in its acoustic model [14]. Simple HMMs that have one Gaussian per context independent phone state were trained first for each speaker. Then the vectors of the means of these models were clustered with the k-means algorithm. For each test utterance, the cluster model that yielded the highest likelihood was selected. In doing so, different channel and noise

conditions can be more precisely modelled by different cluster models, so noise related problems are alleviated. In addition to speaker clustering, speaker adaptation is effective to bring the performance of a speaker- independent system closer to that of a speaker-dependent system. Unlike normal speaker adaptation, the adaptation in [14] was massive in the sense that the adaptation data was obtained from a pool of recent callers rather than a single speaker. The massive adaptation is helpful due to the fact that a caller often calls the same set of individuals, and that a caller may try a name repeatedly when a recognition error occurs. While massive adaptation is helpful to bring down the

error rate for frequent callers, unsupervised utterance adaptation aims at improving the accuracy from an unknown speaker. In this adaptation scheme, the test utterance itself was used for adaptation with a two pass decoding. In the first pass, a speaker independent system or the system after massive adaptation was used to obtain the automatic transcript. Then a forward backward algorithm was applied to obtain the adaptation statistics. After adapting the acoustic models using the collected statistics, the caller's utterance was decoded in a second pass with the adapted model – this second pass may adversely increase the latency of a voice search system. Overall, with all these acoustic model enhancements and an unsupervised derivation of pronunciations (to be described below), a 28% error reduction was observed.

Pronunciation Modelling: One approach to improved pronunciation model is via augmenting the dictionary with pronunciation variants. Data-driven algorithms are commonly applied, which typically include four steps: generating phonetic transcriptions with a recognizer; aligning the auto transcriptions with manually created canonical pronunciations; deriving rules mapping from canonical pronunciations to the variants; and pruning the rules. One limitation of this approach is that the canonical reference pronunciations must be available.

The IBM auto-attendant system adopted an acoustics-only based pronunciation generation approach. The advantage of this approach is that no canonical pronunciation is required. This makes it more practical in voice search applications since many words do not exist in a pronunciation dictionary. With this approach, a trellis of sub-phone units was constructed from an utterance. The transition probabilities in the trellis were derived by weighting the transition probabilities of all the context-dependent realizations of the sub-phone units in a HMM acoustic model. A Viterbi search was performed to obtain the best sub-phone sequences from the trellis and a pronunciation was subsequently derived from the sequence. Experiments showed a 17% relative error reduction when the test set and training set had overlapping unseen words.

References

[1] “Voice recognition- The next revolution” 5th edition.

[2] Ksenia Shalonova, “Automatic Voice Recognition” 07 DEC 2007

[3]Source:http://www.cs.bris.ac.uk/Teaching/Resources/COMS12303/lectures/Ksenia_Shalonoa- Voice_Recognition.pdf

[4] "Fundamentals of Voice Recognition". L. Rabiner & B. Juang. 1993. ISBN: 0130151572.

[5] http://www.abilityhub.com/Voice/speech-description.htm

[6] Charu Joshi “Voice Recognition” Source:

http://www.scribd.com/doc/2586608/Voicerecognition.pdf

[7] John Kirriemuir “Voice recognition technologies”

[8] http://electronics.howstuffworks.com/gadgets/high-techgadgets/Voicerecognition.htm

[9] https://realpython.com/python-speech-recognition/

[10]https://www.researchgate.net/publication/334285447_Project_Report_On_AI_Voice_Recognition_System