CALL DISPOSITION USING COMPREHEND API

A Project Report Submitted to

Jawaharlal Nehru Technological University Anantapur, Ananthapuramu

in partial fulfillment of the requirements for the

Award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND SYSTEMS ENGINEERING

Submitted by

S.SREENIJA	**16121A15A4**
M.DEEPIKA	**16121A1563**
P.JEEVITHA	**16121A1585**
B.MANIKAR	**16121A15B6**

Under the Guidance of

Dr.C.Sushama

Assosiate Professor

Dept. of CSSE, SVEC

Department of Computer Science and Systems Engineering

Sree Vidyanikethan Engineering College (Autonomous)

Sree Sainath Nagar, Tirupati – 517 102

(2016-2020)

April, 2020

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

Sree Sainath Nagar, Tirupati

Department of Science and Systems Engineering

CERTIFICATE

This is to certify that the project report entitled

"CALL DISPOSITION USING COMPREHEND API"

is the Bonafide work done by

S.SREENIJA	**16121A15A4**
M.DEEPIKA	**16121A1563**
P.JEEVITHA	**16121A1585**
B.MANIKAR	**16121A15B6**

in the Department of **Computer Science and Systems Engineering**, and submitted to Jawaharlal Nehru Technological University Anantapur, Ananthapuramu in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Systems Engineering during the academic year 2019-2020. This work has been carried out under my supervision. The results of this project work have not been submitted to any university for the award of any degree or diploma.

Guide:	***Head:***
Dr.C.Sushama	Dr. C Madhusudhana Rao
Assosiate Professor	Professor& Head
Dept. of CSSE	Dept. of CSSE
INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

We hereby declare that the project report titled **Call Disposition Using Comprehend API** is the genuine work carried out by us, in **B.Tech(Computer Science and Systems Engineering)** degree course of **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR** and has not been submitted to any other college or University for the award of any degree or diploma.

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the students

(S.SREENIJA 16121A15A4)

(M.DEEPIKA 16121A1563)

(P.JEEVITHA 16121A1585)

(B.MANIKAR 16121A15B6)

ACKNOWLDGEMENT

We are extremely thankful to our beloved Chairman and Founder **Dr. M. Mohan Babu,**Padmasri awardee of SreeVidyanikethan Educational Institutions who took keen interest and encouraged me in every effort throughout this B. Tech Program.

We are extremely thankful to **Mr. Vishnu Manchu,** CEO of Sree Vidyanikethan Educational Institutions who took keen interest and encouraged me in every effort throughout this B. Tech Program.

We owe our gratitude to **Dr. P. C. Krishnamachary,** Principal, Sree Vidyanikethan Engineering College for permitting me to use the facilities available to accomplish the Project course successfully.

We express our heartfelt thanks to **Dr. C. Madhusudhanrao,** Professor and Head, Department of Computer Science and Systems Engineering, for his kind attention and valuable guidance to me throughout the Project course.

We are extremely thankful to our Project Supervisor **Dr. C.Sushama,** Associate Professor of Computer Science and Systems Engineering , who took keen interest and encouraged me in every effort throughout the Project course.

We are thankful to our Project Coordinator **Dr. C. Sushama,** Associate Professor of CSSE for her valuable support and guidance throughout the Project course.

We express our sincere thanks to all the teaching and non-teaching staff of Computer Science and Systems Engineering Department for their cooperation.

Signatures of the students

1.(S.SREENIJA 16121A15A4)

2.(M.DEEPIKA 16121A1563)

3.(P.JEEVITHA 16121A1585)

4.(B.MANIKAR 16121A15B6)

ABSTRACT

A call disposition is technique that describes the outcome of a call. Call dispositions are actually huge time savers, as they act as a one-click shorthand for call outcomes, and save representatives lots of time in manual note-taking. The benefits of call disposition includes Effortlessly Cascading Information to the Entire Team, Assigning Custom Labels for Instant Customer Overview ,Easily Keeping the Management Informed . There's another benefit to call dispositions as well.Latent Dirichlet Allocation(LDA) is using to get the topic summarization. Like call notes templates, call dispositions standardize input, which means it gives you better visibility into the outcome of calls to monitor activity, track trends, identify patterns, and much more, across your entire organization . The only Limitation in this technique is about the status of the call. The data from phone calls or text is taken as input to services . The call recording is converted into text and punctuated according to the sentences. These text is Summarized Accordingly.

Keywords:

Call disposition; Latent Dirichlet Allocation(LDA); Topic summarization;

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

LDA	Latent dirichlet allocation
NLP	Natural language processing
NLTK	Natural language toolkit
DUC	Document understing conferences
ASR	Automatic speech recognition
CRM	Customer relationship management.

NOTATIONS

α	The per-document topic distributions,
β	The per-topic word distribution,
θ	The topic distribution for document *m*,
φ	The word distribution for topic *k*,
z	The topic for the *n*-th word in document *m*, and
w	The specific word

1.INTRODUCTION

1.1 INTRODUCTION TO THE TOPIC

A call disposition describes the outcome of a call. They include statuses like, “demo scheduled,” “left voicemail”, and even “no longer in service.” In fact, customer service teams have actually logged call dispositions for years, and sales teams have followed to record call outcomes.Call dispositions give sales managers valuable intelligence they can use to optimize their team’s phone performance and sales process. Best of all, properly implemented call dispositions are actually huge time savers, as they act as a one-click shorthand for call outcomes, and save reps lots of time in manual note-taking. Automated Call Categorization is a Fundamental Component of Voice of Customer Engagement Analytics.Automated Call Categorization Has Been Tested And Proven By Customers To Be More Accurate Than Manual Call Categorization, And Can Be Used With Confidence For Decision-making And Across 100% of Your Customer Interactions.

Customer contact centers, or call centers, represent a new domain for integrating advanced speech technologies for improved customer services. Most notably there has been the trend for older complicated touch-tone menus to be converted into voice-enabled directed dialogs. In recent years several commercial systems have been developed to automatically route calls to particular service agents based on a brief spoken problem description .Call routing systems save money and improve customer satisfaction by connecting the customer to the most appropriate and skilled agent. The router reduces queue time and call duration – saving both time and money while simultaneously improving customer satisfaction. To-date most commercial systems have considered deployments in large-scale call center environments where work forces tend to consist of highly specialized agents that perform specific business tasks. Collaborative group interactions between contact agents within large call centers tends to be low. Smaller call centers are more abundant and diversified in terms of group interactions needed to solve tasks. The need for automatic summarization increases as the amount of textual information increases. A lot of information is available on internet but to sort out the

required information is a tedious job. The need for technologies that can do all the sorting and quickly identify the relevant information on its own therefore plays an important role.

Text Summarization looks to covert a large body of text to a few sentences without losing the key themes of the text. Summarization can be broadly classified into two categories – extractive summarization techniques and abstractive summarization techniques. As call centers gets calls as input rather than lots of text , the agent needs to summarize data. A call disposition describes the outcome of a call. They include statuses like, “demo scheduled,” “left voicemail”, and even “no longer in service.” In fact, customer service teams have actually logged call dispositions for years, and sales teams have followed to record call outcomes.

Call dispositions give sales managers valuable intelligence they can use to optimize their team’s phone performance and sales process. Best of all, properly implemented call dispositions are actually huge time savers, as they act as a one-click shorthand for call outcomes, and save reps lots of time in manual note-taking.

Nearly all call center managers understand the importance of equipping their team with the call center software and associated tools that they need to excel. However all too often, when searching for a new call center software solution, call disposition codes are not high on their list of must have call center software features **.**This is a costly oversight. Call disposition codes are a simple feature that have an enormous impact on call center productivity, collaboration and customer satisfaction ratings. They are one of those features that call centers cannot afford. In such conditions, agents are more likely to collaborate to address new or unseen problems and share skill sets.

Pointing out the need for new speech and language technologies for call centers that can facilitate collaborative work environments. Believing that careful integration of speech and language systems can help to foster new and effective forms of collaboration between agents while improving customer service, call efficiency, reducing agent workload and training time. To begin to focus on these new directions for call center speech research, this paper presents initial results towards collection and analysis of the University of Colorado Call Center Corpus. The first describe the corpus collection, transcription, and call-type analysis. Calls are labeled into a hierarchical calltypetaxonomy. Then consider four machine learning methods or automatically classifying incoming calls based on this call-type labeling. Results are shown both for classifiers trained from human-transcribed data as well as for classifiers trained in a lightly-supervised manner.

Deploying new speech recognition technologies into call centers is expensive and time-consuming due to the cost of transcribingand labeling data for each new task domain. In this paper and also consider issues related to utilizing untranscribed data as well as HTML text-documents related to the task-domain for improving acoustic and language modeling for call center transcription. Earlier work by Lamelhave shown that given enough untranscribed data and a good language model, the unsupervised training can achieve similar performance with regards to supervised training. In this work we explore this methodology and consider both the issues related to unsupervised acoustic and language model training from the vast amount of data that can be collected from a call center.

1.2 PROBLEM STATEMENT

Reducing the time delay by converting the input (voice message) into text and then summarizing the converted text . This project describes a system for the summarization of single and multiple documents. The system produces multi as well as single document summaries using lda techniques for identifying common terms across the set of documents. For each term, the system identifies representative passages that are included in the final summary. Results of our evaluation are also presented.

1.3 MOTIVATION

Nearly all call center managers understand the importance of equipping their team with the call center software and associated tools that they need to excel. However all too often, when searching for a new call center software solution, call disposition codes are not high on their list of must-have call center software features. This is a costly oversight. Call disposition codes are a simple feature that have an enormous impact on call center productivity, collaboration and customer satisfaction ratings. They are one of those features that call centers cannot afford to be without.

Call disposition codes, or activity codes, are quick labels that can be applied to call records that describe the call. They allow the agent who handled the call to:

Indicate the type of call (i.e., return, shipping issue, bug, etc.)

Indicate the reason for the call (i.e., purchase new product, complaint, etc.)

Indicate the outcome of the call (i.e., resolved, refund, escalated to manager, transferred to billing, etc.)

Designate a required action (i.e., follow-up required, confirm with management, issue refund at end of month, etc.)

Call disposition codes take seconds for the call center agent to apply and capture relevant information about the call that teams need to thrive. After an agent finishes the call, the disposition code window pops up in their browser. They can easily select the code from a dropdown menu and apply it to the call record with a click of their mouse. Call disposition codes are therefore a quick and easy way to label calls and keep track of customer contact history so that no detail about the call slips through the cracks.

1.4 OBJECTIVES OF THE PROJECT WORK

The main Objective of this Project is to reduce the time for analyzing the problem raised by the customer to the call center agents.Call Summarization is the technique which reduces the time for analysis.

To Gain business insights – capture the customer experience, identify product and purchase trends across your customer base, and identify valuable leads.

To Ensure script compliance – monitor the percent of employees that stay on script, and identify the ones that need redirection.

To Improve workforce training – identify instances of non-compliance, or where no appointment is set, to retrain agents who are not following the new script.

To Improve sales and marketing effectiveness – categorize calls by source of inquiry and collect statistics on the frequency of client inquiries about specific products.

1.5 ORGANIZATION OF THESIS

The report is organized into five chapters .The context of the method, a brief definition of the problem, motivation to propose a new method, the objectives and outline of the thesis is presented in Chapter 1. In Chapter 2, the related prior research work is briefly reviewed. The physical model, a new technique is used to analyze the call conversation using the Algorithm to reduce the analysis time, performance measurement of the problem and performance analysis are discussed in the Chapter 3. The results of the method and the comparison with the observed values and the discussions about it are briefed in the Chapter 4. In Chapter 5, the analysis and the outcome of the method and the future work related to the method are discussed. The lists of references that form part of the report are appended.

1.6 ADVANTAGES

The helpful information about call disposition codes as well as the top 7 benefits of using call disposition codes in the call center:

- **Keep your entire team on the same page**

Making sure all relevant parties are informed about the outcome of calls can be a time consuming process in the call center. Agents typically send out an email to their colleague, walk over to the agent who asked for an update, call a manager or chat with the tech support team. With call disposition codes, this inefficient process can be thrown out the window. When a call has finished, the agent can simply select a call disposition code from a dropdown menu. If they would like to add more call details than the code can capture, they can write a note. As soon as the agent applies the code it will be updated in the CRM, call center software and helpdesk. Thus, anyone on their team can access this information in real-time just by opening the customer's activity history. This makes it simple for the entire team to be informed about the outcome of the call as soon as it finishes.

- **Easily alert teammates to follow up**

If an agent finishes a call that requires follow up from their teammate or manager, they can easily add the call disposition code "Follow Up Required" and a note to the call record. The agent assigned to the contact will then see this code when they open the contact and know immediately that they need to follow up with the customer. The agent can open the note to view the details and also add their own note or change the call disposition code as needed. This makes it simple to alert teammates to follow up with a customer so that everyone can remain on the same page.

- **Assign custom labels for an at-a-glance customer overview**

Create customized call disposition codes to reflect the customer data that is most relevant to your team. This will allow your team to gain a more comprehensive overview of the customer just by opening their account history Bottom of Form

- **Scrub call lists to remain DNC compliant**

Making sure your call center remains DNC compliant can often be a difficult and time consuming process. Using call disposition codes can make this a bit easier. Agents can label each call with the following codes: "disconnected," "fax machine," "busy signal," "incorrect number," "inactive number," "requested no contact," etc. These phone numbers can then be removed from campaign lists, call center software and CRM so that they aren't contacted in the future. Scrubbing call lists and your CRM

using call disposition codes makes it simple to ensure that your team remains DNC compliant.

- **Create more successful calling campaign lists**

Creating optimal calling campaign lists can mean the difference between a successful call campaign and a waste of time. With call disposition codes, it is simple to create campaign lists with only the contacts who are most likely to have a positive response. Simply sort contact lists based on disposition codes and export the list in .csv format. Then email the list to all agents or a subset of agents for them to import as their contact list. This makes it simple to create a calling campaign list that has the potential for big results.

- **Easily keep management informed**

Call center managers don't have a lot of time to dig through a customer's call recordings and contact history searching for the outcome of an important call. With call disposition codes, they don't have to. They can see the disposition codes that were applied to the interaction from the customer's activity history. This makes it easy to gain a comprehensive understanding of call outcomes in seconds.

- **Benefit from comprehensive reporting on call outcomes**

Managers who like analyzing more targeted call center metrics love call disposition codes. They allow them to view exactly how many (and the percentage of) calls resulted in a sale, answering machine, dropped call, busy signal, return, exchange, escalation to manager, transfer to tech support, etc. Essentially any call outcome that is meaningful to the team can be turned into a disposition code and tracked over time. Call disposition codes are therefore an awesome tool to keep managers informed of the metrics that matter most to them.Call disposition codes are a simple, yet powerful tool that keeps the entire team on the same page and allows call center managers to gain a comprehensive understanding of metrics that are most meaningful to them. Next time you're on the market for call center software make sure call disposition codes are on your must-have feature list. Doing so will ensure that your team has the tools they need to acquire relevant data about their callers so that no detail slips through the cracks.

2.REVIEW OF LITERATURE

The author[1] proposed “Speech Enaled Natural Language Call Routing” which states that-The design and performance of the BBN Call Director product for automatic call routing and the methodology for its deployment. The component technologies for the BBN Call Director are a statistical n-gram speech recognizer and a statistical topic identification system that, together, provide the framework for processing natural language responses from callers. To achieve commercial success, however, a superior deployment process is as important as the technology itself. In order to minimize the operational and financial risk for the call center, BBN has developed a deployment process that provides an integrated methodology for building the business case, optimizing performance, and proving the benefit of natural language call routing. This process is based on quantifying IVR benefit in terms of saved agent labor by analyzing end-to-end recordings of live calls. Routing accuracy is of critical importance because of the direct cost impact of a misrouted call to the call center. Experimental results on real traffic coming into a customer call center indicate that BBN Call Director can reduce the number of misrouted calls by 28%, which translates to a savings of 2 to 4 minutes for each of those calls. For large call centers handling several million calls per year, the corresponding cost savings can be in the millions of dollars.

The author[2] proposed Unsupervised acostic system that describes some recent experiments using unsupervised techniques for acoustic model training in order to reduce the system development cost. The approach uses a speech recognizer to transcribe un-annotated raw broadcast news data. The hypothesized transcription is used to create labels for the training data. Experiments providing supervision only via the language model training materials show that including texts which are contemporaneous with the audio data is not crucial for success of the approach, and that the acoustic models can be initialized with as little as 10 minutes of manually annotated data. These experiments demonstrate that unsupervised training is a viable training scheme and can dramatically reduce the cost of building acoustic models.

The author[3] proposed lightly supervised and unsupervised acostic model training which depends on computer speech language and The last decade has witnessed substantial progress in speech recognition technology, with today's state-of-the-art systems being able to transcribe unrestricted broadcast news audio data with a word error of about 20%. However, acoustic model development for these recognizers relies on the availability of large amounts of manually transcribed training data. Obtaining such data is both time-consuming and expensive, requiring trained human annotators and substantial amounts of supervision. Experiments providing supervision only via the language model training materials show that including texts which are contemporaneous with the audio data is not crucial for success of the approach, and that the acoustic models can be initialized with as little as 10 min of manually annotated data. These experiments demonstrate that light or no supervision can dramatically reduce the cost of building acoustic models.

Despite the rapid progress made in large vocabulary continuous speech recognition ,there remain many outstanding challenges .One of the main challenges is to reduce the cost ,both in terms of human effort and financial needs ,required to adapta recognition system to a new task or another language. One of the most often cited costs is that of obtaining the necessary transcribed acoustic training data ,which is an expensive process in terms of both manpower and time. There are certain audio sources, such as radio and television broadcasts ,that can provide an essentially unlimited supply of acoustic training data.However,forthevastmajorityofaudiodatasourcestherearenocorrespondingaccurate word transcriptions. Some of these sources, in particular ,the main American television channels also broadcast manually derived closed-captions. The closed-captions area close, but not exact transcription of what is being spoken ,and these are only coarsely time-aligned with the audio signal. Manual transcripts are also available forcertain radio

Broadcasts. Some preliminary experiments with This work was partially financed by the European Commission under the Human Language Technologies project Coretex.lightly supervised acoustic model training were described, where the basic idea is to use a speech recognizer to automatically transcribe unannotated data, thus generating

"approximately" labeled training data. By iteratively increasing the amount of training data, more accurate acoustic models are obtained, which can then be used to transcribe another set of unannotated data. A straight forward approach of training on all the automatically annotated data was compared with one in which the closed captions are used to filter the hypothe-sized transcriptions, removing words that are "incorrect". To our surprise, somewhat comparable recognition results were
obtained both with and without filtering, suggesting that inclusion of the closed-captions in the language model training material provided sufficient supervision. Although the idea of using untranscribed data to train acoustic models has been
proposed before, Everyone are not aware of any other large scale experiments with this technique on a publicly available corpora.The effects of using different levels of supervision, as provided by the language model training texts, on the accuracy of the acoustic models constructed using automatically generated word transcriptions. The next section presents the basic idea so flightly supervised training, followed by a description of the corpora used in this work andan overview of the LIMSI broadcast news transcription system.

State-of-the-art speech recognition systems are trained using transcribed utterances, preparation of which is labor intensive and time-consuming. In this paper,author[4] Describing a new method for reducing the transcription effort for training in automatic speech recognition (ASR). Active learning aims at reducing the number of training examples to be labeled by automatically processing the unlabeled examples, and then selecting the most informative ones with respect to a given cost function for a human to label. Automatically estimate a confidence score for each word of the utterance, exploiting the lattice output of a speech recognizer, which was trained on a small set of transcribed data. Compute the utterance confidence scores based on these word confidence scores, then selectively sample the utterances to be transcribed using the utterance confidence scores. In our experiments, to show that the reduce amount of labeled data needed for a given word accuracy by 27%.State-of-the-art speech recognition systems require transcribed utterances for training, and transcription is a labor intensive and time-consuming process. Active learning aims at reducing the number of training examples to be labeled by inspecting the unlabeled examples, and intelligently selecting the most

informative ones with respect to a given cost function for a human to label [1). The goal of the learning algorithm is to select the examples for labeling which will have the largest improvement on the performance.Describing a new method for reducing the transcription effort for training in ASR, by selectively sampling a subset of the data. For this purpose, Automatically label each word of the utterance with a confidence score, exploiting the lattice output of a speech recognizer, which was initially trained on a small set of transcribed data. Compute the utterance confidence scores from the word-based confidence scores, and selectively sample the utterances to be transcribed using these scores. Testing the approach in the framework of AT&T's How May 1 Help you,sM natural spoken dialog system. Transcription is an important procedure both for extending the system to other domains, and for incorporating new calltypes into the existing system. The transcription capability is limited, so selective sampling over the terabytes of speech database is crucial. In the following, First describe the related work in the machine learning domain, as well as review some of the related work in language processing. Describing the algorithm, and describing how we compute confidence scores using the lattice output of ASR. Describing experiments and results.

Describing the results of the Call Centre Conversation Summarization task at Multiling'15. According to author[5] CCCS task consists in generating abstractive synopses from call centre conversations between a caller and an agent. Synopses are summaries of the problem of the caller, and how it is solved by the agent. Generating them is a very challenging task given that deep analysis of the dialogs and text generation are necessary. Three languages were addressed: French, Italian and English translations of conversations from those two languages. The official evaluation metric was ROUGE-2. Two participants submitted a total of four systems which had trouble beating the extractive baselines. The datasets released for the task will allow more research on abstractive dialog summarization. Speech summarization has been of great interest to the community because speech is the principal modality of human communications, and it is not as easy to skim, search or browse speech transcripts as it is for textual messages. Speech recorded from call centres offers a great opportunity to study goal-oriented and focused conversations between an agent and a caller. The Call Centre Conversation Summarization (CCCS) task consists in automatically generating summaries of spoken conversations in the form of textual synopses that shall inform on the content of a conversation and might

be used for browsing a large database of recordings. Compared to news summarization where extractive approaches have been very successful, the CCCS task's objective is to foster work on abstractive summarization in order to depict what happened in a conversation instead of what people actually said. The track leverages conversations from the Decoda and Luna corpora of French and Italian call centre recordings, both with transcripts available in their original language as well as English translation (both manual and automatic). Recordings duration range from a few minutes to 15 minutes, involving two or sometimes more speakers. In the public transportation and help desk domains, the dialogs offer a rich range of situations (with emotions such as anger or frustration) while staying in a coherent and focused domain. Given transcripts, participants to the task shall generate abstractive summaries informing a reader about the main events of the conversations, such as the objective of the caller, whether and how it was solved by the agent, and the attitude of both parties. Evaluation has been performed by comparing submissions to reference synopses written by quality assurance experts from call centres. Both conversations and reference summaries are kindly provided by the SENSEI project. This paper reports on the results of the CCCS task in term ROUGE-2 evaluation metric. Two participants have submitted four systems to the task. In addition, provides three baselines which frame the performance that would be obtained by extractive systems. The results are analysed according to language, human annotator coherence and the impact of automatic translation.

Presenting an automatic abstractive summarization system of meeting conversations. Our system extends a novel multi-sentence fusion algorithm in order to generate abstract templates. Using these templates we can make the algorithm work much more efficient and It also leverages the relationship between summaries and their source meeting transcripts to select the best templates for generating abstractive summaries of meetings. Manual and automatic evaluation results demonstrate the success of our system in achieving higher scores both in readability and informativeness. The main contributions of are: 1) The successful adaptation of a word graph algorithm to generate templates from human authored summaries; 2) The implementation of a novel template selection algorithm that effectively leverages the relationship between human authored summary sentences and their source transcripts; and 3) A comprehensive testing of our approach,

comprising both automatic and manual evaluations. Instantiate the framework on the AMI corpus and compare our summaries with those created from a state-of the-art systems. The evaluation results demonstrate that our system successfully creates informative and readable summaries.

In order for summaries to be readable and informative, they should be grammatically correct and contain important information in meetings. To this end, creating the framework consisting of the following two components: 1) An off-line template generation module, which generalizes collected human-authored summaries and creates templates from them; and 2) An online summary generation module, which segments meeting transcripts based on the topics discussed, extracts the important phrases from these segments, and generate abstractive summaries of them by filling the phrases into the appropriate templates. cluster the templates into similar groups.Utilize root verb information for this process assuming that these verbs such as “discuss” and “suggest” that appear in summaries are the most informative factors in describing meetings. Therefore, after extracting root verbs in summary sentences, creates fully connected graphs where each node represents the root verbs and each edge represents a score denoting how similar the two word senses are. To measure the similarity of two verbs, first identify the verb senses based on their frequency in WordNet and compute the similarity score based on the shortest path that connects the senses in the hypernym taxonomy.Then convert the graph into a similarity matrix and apply a Normalized Cuts method to cluster the root verbs. Finally, all templates are organized into the groups created by their root verbs.

3.METHODOLOGY

Topic Modeling is a technique to extract the hidden topics from large volumes of text. Latent Dirichlet Allocation(LDA) is a popular algorithm for topic modeling with excellent implementations in the Python's Gensim package. The challenge, however, is how to extract good quality of topics that are clear, segregated and meaningful**.** This depends heavily on the quality of text preprocessing and the strategy of finding the optimal number of topics.

3.1. LATENT DIRICHLET ALLOCATION

Latent Dirichlet Allocation (LDA) is a "generative probabilistic model" of a collection of composites made up of parts. Its uses include Natural Language Processing (NLP) and topic modelling, among others.In terms of topic modelling, the composites are documents and the parts are words and/or phrases (phrases *n* words in length are referred to as n-grams).But you could apply LDA to DNA and nucleotides, pizzas and toppings, molecules and atoms, employees and skills, or keyboards and crumbs.The probabilistic topic model estimated by LDA consists of two tables (matrices). The first table describes the probability or chance of selecting a particular part when sampling a particular topic (category).

The LDA algorithm assumes your composites were generated like so:

- Pick your unique set of parts.
- Pick how many composites you want.
- Pick how many parts you want per composite (sample from a Poisson distribution).
- Pick how many topics (categories) you want.
- Pick a number between not-zero and positive infinity and call it *alpha*.
- Pick a number between not-zero and positive infinity and call it *beta*.

- Build the ‘parts-versus-topics’ table. For each column, draw a sample (spin the wheel) from a Dirichlet distribution (which is a distribution of distributions) using *beta* as the input. Each sample will fill out each column in the table, sum to one, and give the probability of each part per topic.

- Build the ‘composites-versus-topics’ table. For each row, draw a sample from a Dirichlet distribution using *alpha* as the input. Each sample will fill out each row in the table, sum to one, and give the probability of each topic per composite.

- Build the actual composites. For each composite, look up its row in the ‘composites-versus-topics’, sample a topic based on the probabilities in the row, go to the ‘parts-versus-topics’ , look up the topic sampled, sample a part based on the probabilities in the column, repeat from step 2 until you’ve reached how many parts this composite was set to have.

3.2. PHYSICAL MODEL

LDA is a generative probabilistic model that assumes each topic is a mixture over an underlying set of words, and each document is a mixture of over a set of topic probabilities.

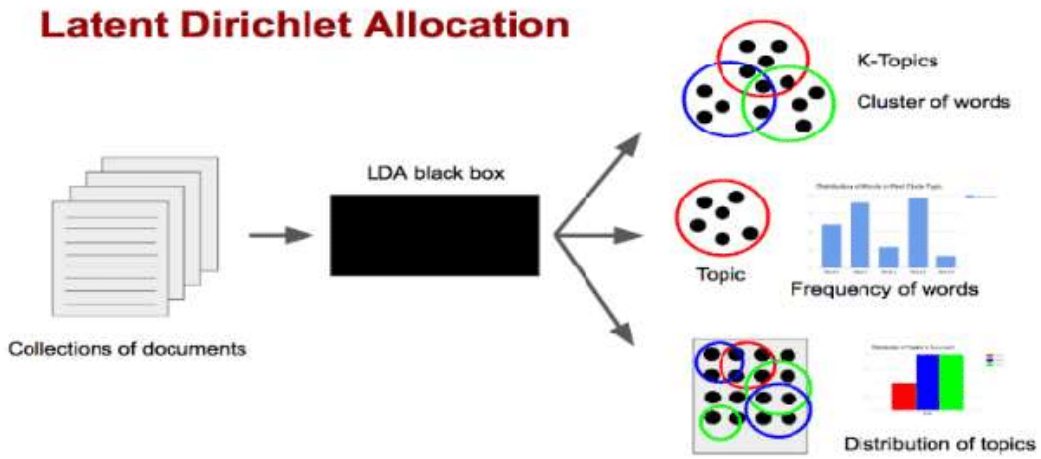

Figure 3.1 Latent Dirichlet Allocation

Latent:

Latent refers to everything that don't know a priori and are hidden in the data. Here, the themes or topics that document consists of are unknown, but they are believed to be present as the text is generated based on those topics.

Dirichlet:

Dirichlet is a 'distribution of distributions'. But what does this mean? Let's think about this with the help of an example. Let's suppose there is a machine that produces dice and can be control whether the machine will always produce a dice with equal weight to all sides, or will there be any bias for some sides. So, the machine producing dice is a distribution as it is producing dice of different types. Also, knows that the dice itself is a distribution as gets multiple values when a dice is roll . This is what it means to be a distribution of distributions and this is what Dirichlet is. Here, in the context of topic modeling, the Dirichlet is the distribution of topics in documents and distribution of words in the topic. It might not be very clear at this point of time, but it's fine as looked at it in more detail in a while.

Allocation:

Allocation means that once having Dirichlet, allocating topics to the documents and words of the document to topics.That's it. This is what LDA is in a nutshell. Now let's understand how this works in topic modeling.Just to recap, what LDA says is that each word in each document comes from a topic and the topic is selected from a per-document distribution over topics. Sothere are two matrices.

- Θtd = P(t|d) which is the probability distribution of topics in documents.
- Φwt = P(w|t) which is the probability distribution of words in topics.

And, says that the probability of a word given document i.e. P(w|d) is equal to:

$$\sum_{t=1}^{T} p(w|t,d)p(t|d)$$

where T is the total number of topics. Also, let's assume that there is W number of words in our vocabulary for all the documents. Assume conditional independence, and can say that

$$P(w|t,d) = P(w|t)$$

The dot product of Θtd and Φwt for each topic t.So, looking at this and can think of LDA similar to that of matrix factorization or SVD, where to decompose the probability distribution matrix of word in document in two matrices consisting of distribution of topic in a document and distribution of words in a topic.And to tie back to our example of dice, and can say that each word in the distribution of words in a topic is similar to a side of the dice, and having Dirichlet parameter to control if all the words have same probability in a topic or will that topic have an extreme bias towards some words. Same intuition is for distribution of topics in a document.Now comes the important part. To start with, let's randomly assign weights to both the matrices and assume that our data is generated as per the following steps:

- Randomly choose a topic from the distribution of topics in a document based on their assigned weights. In the previous example, let's chose pink topic
- Next, based on the distribution of words for the chosen topic, select a word at random and put it in the document
- Repeat this step for the entire document

3.3. ALGORITHM FOR THIS METHOD

Step 1: LDA is a form of unsupervised learning that views documents as bags of words (ie order does not matter). LDA works by first making a key assumption: the way a document was generated was by picking a set of topics and then for each topic picking a set of words.it reverse engineers this process. To do this it does the following for each document m.

Step 2 : Assume there are k topics across all of the documents

Distribute these k topics across document m (this distribution is known as α and can be symmetric or asymmetric, more on this later) by assigning each word a topic.

Step 3 : For each word w in document m, assume its topic is wrong but every other word is assigned the correct topic.

Step 4 : Probabilistically assign word w a topic based on two things:

- what topics are in document m

- how many times word w has been assigned a particular topic across all of thedocuments (this distribution is called β, more on this later)

Step 5 : Repeat this process a number of times for each document and you're done!

3.4. GENSIM

Gensim is billed as a Natural Language Processing package that does 'Topic Modeling for Humans'. But its practically much more than that.

Topic modeling, it is a technique to extract the underlying topics from large volumes of text. Gensim provides algorithms like LDA and LSI and the necessary sophistication to build high-quality topic models.Topic models and word embedding are available in other packages like scikit, R etc. But the width and scope of facilities to build and evaluate topic models are unparalleled in gensim, plus many more convenient facilities for text processing.

It is a great package for processing texts, working with word vector models (such as Word2Vec, FastTextetc) and for building topic models.

Also, another significant advantage with gensim is: it handle large text files without having to load the entire file in memory.

Parameters:

- **corpus** (*{iterable of list of (int, float), scipy.sparse.csc}, optional*) – Stream of document vectors or sparse matrix of shape (*num_terms*, *num_documents*). If not given, the model is left untrained (presumably because you want to call **update()** manually).
- **num_topics** (*int, optional*) – The number of requested latent topics to be extracted from the training corpus.
- **id2word** ({dict of (int, str), **gensim.corpora.dictionary.Dictionary**}) – Mapping from word IDs to words. It is used to determine the vocabulary size, as well as for debugging and topic printing.
- **distributed** (*bool, optional*) – Whether distributed computing should be used to accelerate training.
- **chunksize** (*int, optional*) – Number of documents to be used in each training chunk.

- **passes** (*int, optional*) – Number of passes through the corpus during training.
- **update_every** (*int, optional*) – Number of documents to be iterated through for each update. Set to 0 for batch learning, > 1 for online iterative learning.
- **alpha** (*{numpy.ndarray, str}, optional*) –

  Can be set to an 1D array of length equal to the number of expected topics that expresses our a-priori belief for the each topics' probability. Alternatively default prior selecting strategies can be employed by supplying a string:

  - 'asymmetric': Uses a fixed normalized asymmetric prior of *1.0 / topicno*.
  - 'auto': Learns an asymmetric prior from the corpus (not available if *distributed==True*).
- **eta** (*{float, np.array, str}, optional*) –

  A-priori belief on word probability, this can be:

  - scalar for a symmetric prior over topic/word probability,
  - vector of length num_words to denote an asymmetric user defined probability for each word,
  - matrix of shape (num_topics, num_words) to assign a probability for each word-topic combination,
  - the string 'auto' to learn the asymmetric prior from the data.
- **decay** (*float, optional*) – A number between (0.5, 1] to weight what percentage of the previous lambda value is forgotten when each new document is examined. Corresponds to Kappa from **Matthew D. Hoffman, David M. Blei, Francis Bach: "Online Learning for Latent Dirichlet Allocation NIPS'10"**.
- **offset** (*float, optional*) –

  Hyper-parameter that controls how much it slow down the first steps the first few iterations. Corresponds to Tau_0 from Matthew D. Hoffman,

David M. Blei, Francis Bach: "Online Learning for Latent Dirichlet Allocation NIPS'10".

- **eval_every** (*int, optional*) – Log perplexity is estimated every that many updates. Setting this to one slows down training by ~2x.
- **iterations** (*int, optional*) – Maximum number of iterations through the corpus when inferring the topic distribution of a corpus.
- **gamma_threshold** (*float, optional*) – Minimum change in the value of the gamma parameters to continue iterating.
- **minimum_probability** (*float, optional*) – Topics with a probability lower than this threshold will be filtered out.
- **random_state** (*{np.random.RandomState, int}, optional*) – Either a randomState object or a seed to generate one. Useful for reproducibility.
- **ns_conf** (*dict of (str, object), optional*) – Key word parameters propagated to **gensim.utils.getNS()** to get a Pyro4 Nameserved. Only used if *distributed* is set to True.
- **minimum_phi_value** (*float, optional*) – if *per_word_topics* is True, this represents a lower bound on the term probabilities.
- **per_word_topics** (*bool*) – If True, the model also computes a list of topics, sorted in descending order of most likely topics for each word, along with their phi values multiplied by the feature length (i.e. word count).
- **callbacks** (list of **Callback**) – Metric callbacks to log and visualize evaluation metrics of the model during training.
- **dtype** (*{numpy.float16, numpy.float32, numpy.float64}, optional*) – Data-type to useduring calculations inside model. All inputs are also converted.

3.5. **DICTIONARY AND CORPUS**

In order to work on text documents, Gensim requires the words (aka tokens) be converted to unique ids. In order to achieve that, Gensim lets you create a Dictionary object that maps each word to a unique id.

By converting your text/sentences to a [list of words] and pass it to the corpora.Dictionary() object. The dictionary object is typically used to create a 'bag of words' Corpus. It is this Dictionary and the bag-of-words (Corpus) that are used as inputs to topic modeling and other models that Gensim specializes in.

Alright, what sort of text inputs can gensim handle? The input text typically comes in 3 different forms:

- As sentences stored in python's native list object
- As one single text file, small or large.
- In multiple text files.

Now, when the text input is large, it is needed to be able to create the dictionary object without having to load the entire text file.The good news is Gensim lets you read the text and update the dictionary, one line at a time, without loading the entire text file into system memory.But, before getting in, let's understand some NLP jargon.

A 'token' typically means a 'word'. A 'document' can typically refer to a 'sentence' or 'paragraph' and a 'corpus' is typically a 'collection of documents as a bag of words'. That is, for each document, a corpus contains each word's id and its frequency count in that document. As a result, information of the order of words is lost.

3.6 PROPOSED SYSTEM

One of the primary applications of natural language processing is to automatically extract what topics people are discussing from large volumes of text. Some examples of large text could be feeds from social media, customer reviews of hotels, movies, etc, user feedbacks, news stories, e-mails of customer complaints etc.

Knowing what people are talking about and understanding their problems and opinions is highly valuable to businesses, administrators, political campaigns. And it's really hard to manually read through such large volumes and compile the topics.Thus is required an automated algorithm that can read through the text documents and automatically output the topics discussed.The Latent Dirichlet Allocation (LDA) from Gensim package along with the Mallet's implementation (via Gensim). Mallet has an efficient implementation of the LDA. It is known to run faster and gives better topics segregation.

3.7. PREREQUISITES – DOWNLOAD NLTKSTOPWORDS

The stopwords from NLTK and spacy's en model for text pre-processing. Later, the spacy model is used for lemmatization.

Lemmatization is nothing but converting a word to its root word. For example: the lemma of the word 'machines' is 'machine'. Likewise, 'walking' –> 'walk', 'mice' –> 'mouse' and so on.

An existingnltk_data directory is searched to install NLTK data. If one does not exist it will attempt to create one in a central location (when using an administrator account) or otherwise in the user's filespace. If necessary, run the download command from an administrator account, or using sudo. The recommended system location isC:\nltk_data (Windows); /usr/local/share/nltk_data (Mac); and /usr/share/nltk_data (Unix). You can use theflag to specify a different location (be sure to set thenltk_data environment variable accordingly).

Run the command python –m nltk.downloader all. To ensure central installation, run the command .

Windows: Use the "Run…" option on the Start menu. Windows Vista users need to first turn on this option, using start->properties->customize to check the box to activate the "Run…" option.

Test the installation: Check that the user environment and privileges are set correctly by logging in to a user account, starting the Python interpreter, and accessing the Brown Corpus

The core packages used are gensim, spacy and pyLDAvis . Besides this and also using numpy,matplotlib and pandas for data handling and visualization.Python is open source object oriented interpreted language. Of the many features, one of the important features that makes python a strong programming language is Python packages. A lot of external packages are written in python which you can be installed and used depending upon your requirement.

Python packages are nothing but directory of python scripts. Each script is a module which can be a function, methods or new python type created for particular functionality. numpy is one such important package created to ease array computation in python.

The process of downloading and installing numpy packages and how to use them in python environment on mac, windows, ubuntu and fedora operating systems. The basics of python programming language are not covered in this blog. For beginners, the basics of python programming language are covered in this Edureka blog.

All python packages are installed using pip – Package Installer for Python. You can view the details of all python packages and download them from Python Package Index (PyPI). However, pip is automatically installed when you download and install python from python.org or any other python integrated environment. Please read the blog for the best python integrated platforms which also provides loads of other functionalities. pip is the simplest way to download packages directly from PyPI from your command line.

3.8. LDA IMPLEMENTATION

LDA's approach to topic modeling is it considers each document as a collection of topics in a certain proportion. And each topic as a collection of keywords, again, in a certain proportion.Once you provide the algorithm with the number of topics, all it does it to rearrange the topics distribution within the documents and keywords distribution within the topics to obtain a good composition of topic-keywords distribution.

A topic is nothing but a collection of dominant keywords that are typical representatives. Just by looking at the keywords, you can identify what the topic is all about.

The following are key factors to obtaining good segregation topics:

The quality of text processing.

The variety of topics the text talks about.

The choice of topic modeling algorithm.

The number of topics fed to the algorithm.

The algorithms tuning parameters.

3.8.1.PREPARE STOPWORDS

Natural Language Processing with PythonNatural language processing (nlp) is a research field that presents many challenges such as natural language understanding.

What are Stop words?A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

Stop words can be filtered from the text to be processed. There is no universal list of stop words in nlp research, however the nltk module contains a list of stop words.The process of converting data to something a computer can understand is referred to as pre-processing. One of the major forms of pre-processing is to filter out useless data. In natural language processing, useless words (data), are referred to as stop words.

3.8.2. REMOVE EMAILS AND NEWLINE CHARACTERS

There are many emails, newline and extra spaces that is quite distracting. Let’s get rid of them using regular expressions.After removing the emails and extra spaces, the text still looks messy. It is not ready for the LDA to consume. The sentences are need to break down each sentence into a list of words through tokenization, while clearing up all the messy text in the process.Gensim’s simple_preprocess is great for this.

3.8.3.TOKENIZE WORDS AND CLEAN-UP TEXT

Tokenization is the process by which big quantity of text is divided into smaller parts called tokens**.**Natural language processing is used for building applications such as Text classification, intelligent chatbot, sentimental analysis, language translation, etc. It becomes vital to understand the pattern in the text to achieve the above-stated purpose.

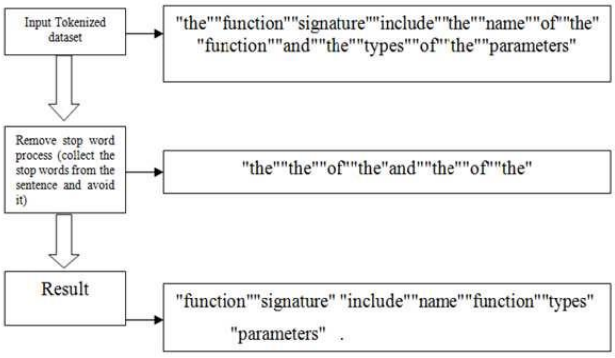

Fig 4.1 Tokenization process

These tokens are very useful for finding such patterns as well as is considered as a base step for stemming and lemmatization**.**Word tokenization is the process of splitting a large sample of text into words. This is a requirement in natural language processing tasks where each word needs to be captured and subjected to further analysis like classifying and counting them for a particular sentiment etc. The Natural Language Tool kit(NLTK) is a library used to achieve this. Install NLTK before proceeding with the python program for word tokenization .Let's tokenize each sentence into a list of words, removing punctuations and unnecessary characters altogether.Gensim's simple_preprocess() is great for this. Additionally I have set deacc=True to remove the punctuations.

- word_tokenize module is imported from the NLTK library.
- A variable "text" is initialized with two sentences.
- Text variable is passed in word_tokenize module and printed the result. This module breaks each word with punctuation which you can see in the output.

3.8.4. REMOVE STOPWORDS

With the Python programming language, A myriad of options to use in order to remove stop words from strings. You can either use one of the several natural language processing libraries such as NLTK, SpaCy, Gensim, TextBlob, etc., or if you need full control on the stop words that you want to remove, you can write your own custom script.
A number of different the approaches, depending on the NLP library you're using.

- Stop Words with NLTK
- Stop Words with Gensim
- Stop Words with SpaCy

Using Python's NLTK Library:

The NLTK library is one of the oldest and most commonly used Python libraries for Natural Language Processing. NLTK supports stop word removal, and you can find the list of stop words in the corpus module. To remove stop words from a sentence, you can divide your text into words and then remove the word if it exits in the list of stop words provided by NLTK.

3.8.5.CREATE THE DICTIONARY AND CORPUS

Gensim creates a unique id for each word in the document. The produced corpus shown above is a mapping of (word_id, word_frequency).For example, (0, 1) above implies, word id 0 occurs once in the first document. Likewise, word id 1 occurs twice and so on.This is used as the input by the LDA model. A dictionary can be made to a certain level of satisfaction if it is made with data and information acquired from widely representative and properly balanced language corpus. A language corpus provides an empirical basis in the selection of words and other lexical items as well as in supplying the most authentic information relating to pronunciation, usage, grammar, meaning, illustration, and other information with which all the words and lexical items in a general reference dictionary are furnished with. In the same manner, a language corpus supplies the most authentic information relating to compounds, idioms, phrases, and proverbs, etc., which is also included within a general reference dictionary with equal attention and importance. To explain how linguistic data and information collected from a corpus can contribute toward compilation a more useful dictionary. Although a corpus has better functional utilities in development of electronic dictionary, To concentrate here on the use of a corpus in the compilation of printed dictionary. Occasionally refer to the TDIL corpora developed in the Indian languages and use linguistic data and information from these to substantiate our arguments and observations.

3.8.6. BUILDING THE TOPIC MODEL

Everything that is required to train the LDA model is present . In addition to the corpus and dictionary, you need to provide the number of topics as well.Apart from that, alpha and eta are hyperparameters that affect sparsity of the topics. According to the Gensim docs, both defaults to 1.0/num_topics prior.chunksize is the number of documents to be used in each training chunk. update_every determines how often the model parameters should be updated and passes is the total number of training passes.

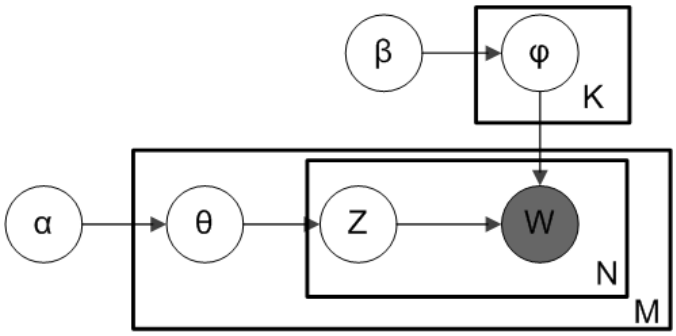

Fig. 4.2 Latent Dirichlet Model

α is the per-document topic distributions,

β is the per-topic word distribution,

θ is the topic distribution for document *m*,

φ is the word distribution for topic *k*,

z is the topic for the *n*-th word in document *m*, and

w is the specific word

3.8.7. FINDING THE DOMINANT TOPIC

Now that the LDA model is built, the next step is to examine the produced topics and the associated keywords.One of the practical application of topic modeling is to determine what topic a given document is about.To find that, finding the topic number that has the highest percentage contribution in that document.

LDA is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

- Each document is modeled as a multinomial distribution of topics and each topic is modeled as a multinomial distribution of words.
- LDA assumes that the every chunk of text feeding into it will contain words that are somehow related. Therefore choosing the right corpus of data is crucial.
- It also assumes documents are produced from a mixture of topics. Those topics then generate words based on their probability distribution.

3.9. PERFORMANCE EVALUATION

Let us analyze the working model with the simple example. This performance measurement of this method will be as follows.

The sentence scoring has been done as follows:

Si = w1*Ci+w2*Ki+w3*Ti+w4*Li

Where, Si – score of sentence I

Ci – score of sentence i based on cue words

Ki – score of sentence i based on keywords

Ti – score of sentence i based on title words

Li – location of sentence in document

w1, w2, w3, w4– are the weights assigned

In short, for document summary, score of a sentence is dependent on the frequency of the words in that sentence, their related weightage as per the details given above and the sum of it.

Example:

Doc:

I am adam(S1).*Result* of that my *computer* is okay(S2) . Its alright.(S3).I am okay now(S4).

S2:

Cue Words	: Result, okay	Ci=2
Key Words	: Result , Computer ,	Ki=2
Title Words	: computer	Ti=1
Location	:2	Li=2

Weights of each word is no of times its repeated in a document

Weight of cue words = no of times ‘result’ repeated + no of times ‘okay’

Repated =1+2=3

w4 is no of times a sentence is repeated

Si = w1 * Ci + w2 * Ki + w3 * Ti + w4 * Li

Score of s2 = 3 * 2 + 3 * 2 + 1 * 1 + 1 * 2= 15

Every sentence in the conversation is scored and the highest scored sentences among all the sentences are listed out as the call summarization.

4.RESULTS AND DISCUSSION

A call recording is taken as input for the call summarization. Many methods are used to get the summarized data. Numerous benchmarking datasets are used for experimental evaluation of extractive summarization. Document Understanding Conferences (DUC) is the most common benchmarking datasets used for text summarization. There area number of datasets like DUC, CNN. It contains documents along with their summaries that are created automatically, manually and submitted summaries.

Human Evaluation

Human judgement usually has wide variance on what's thought-about a "good" outline, which implies that creating the analysis method automatic is especially tough. Manual analysis is used, however, this can be each time and labor intensive because it needs humans to browse not solely the summaries however conjointly the supply documents. Other issues are those regarding coherence and coverage.

- Tokenizer result: dividing into single word

  Result:

Table 5.1. Tokenizing the words

S.NO	INPUT	TOKENIZER RESULT
1	My laptop is not working since past few days. I, don't know the cause of error. If you can help me with that, it would be nice. My BIOS is giving me a message stating primary battery. Replace I, don't understand. What is that? Please help me.	['My', 'laptop', 'is', 'not', 'working', 'since', 'past', 'few', 'days', 'I', 'don', 't', 'know', 'the', 'cause', 'of', 'error', 'If', 'you', 'can', 'help', 'me', 'with', 'that', 'it', 'would', 'be', 'nice', 'My', 'BIOS', 'is', 'giving', 'me', 'a', 'message', 'stating', 'primary', 'battery', 'Replace', 'I', 'don', 't', 'understand', 'What', 'is', 'that', 'Please', 'help', 'me']
2	I recharged my phone two days ago .But , it is not updated.	[‘I’,’recharged’,’my’,’phone’,’two’,’days’,’ago’, ‘But’,’it’,’is’,’not’,’updated’]

- Removing stopwords:

  A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

  en_stop = stopwords.words('english')

  Tokens after removing en_stopwords from the existing sentences.

  Result:

  ['My', 'laptop', 'working', 'since', 'past', 'days', 'I', 'know', 'cause', 'error', 'If', 'help', 'would', 'nice', 'My', 'BIOS', 'giving', 'message', 'stating', 'primary', 'battery', 'Replace', 'I', 'understand', 'What', 'Please', 'help']

- Extracted good quality of topics that are clear, segregated and meaningful using LDA Algorithm.

  Result:

  ['help', 'I', 'My', 'past', 'message', 'nice', 'laptop', 'primary', 'working', 'understand']

- Number of sentences to return for a "top n" summary

  The top scored sentences are segregated from the existing sentences.

  Result:

  ['My laptop is not working since past few days.', 'My BIOS is giving me a message stating primary battery.']

Call center managers don’t have a lot of time to dig through a customer’s call recordings and contact history searching for the outcome of an important call. With call disposition codes, they don’t have to. They can see the disposition codes that were applied to the interaction from the customer’s activity history. This makes it easy to gain a comprehensive understanding of call outcomes in seconds.

Call summarization Result:

Keywords are essential in identifying the importance of the sentence. Thesentence that consists of main keywords is most likely included in the final summary. Techniques involveassigning a score to sentences based on a countenance that are predefined based on the methodology applied. Both wordlevel and sentence level features are employed in text summarization literature.

Result:

```
My laptop is not working since past few days. I, don't know the cause of error.
If you can help me with that, it would be nice. My BIOS is giving me a message s
tating primary battery. Replace I, don't understand. What is that? Please help m
e.
{'top_n_summary': ['My laptop is not working since past few days.', 'My BIOS is
giving me a message stating primary battery.']}
>>>
```

5.CONCLUSION AND FUTURE WORK

CONCLUSION: Extractive summarization process is highly coherent, less redundant and cohesive (summary and information rich). The aim is togive a comprehensive review and comparison of distinctive approaches and techniques of extractive text summarization process. Although research on summarization started way long back, there is still a long way to go. Over thetime, focused has drifted from summarizing scientific articles to advertisements, blogs, electronic mail messages and news articles. Simple eradication ofsentences has composed satisfactory results in massive applications. Some trends in automatic evaluation ofsummary system have been focused. However, the work has not focused the different challengesof extractive text summarization process to its full intensity in premises of time and space complication.

Evaluating summaries (either automatically or manually) is a difficult task. The mainproblem in evaluation comes from the impossibility of building a standard against which the results of the systems that have to be compared. Further, itis very hard to find out what a correct summary is because there is a chance of the system to generate a better summary that is different from any human summary which is used as an approximation to correct output. Content choice is not a settled problem.Peopleare completely different and subjective authors would possibly select completely different sentences. Two distinct sentencesexpressed in disparate words will specific a similar can explicit the same meaning also known as paraphrasing. There existsan approach to automatically evaluate summaries using paraphrases (paraEval). Most text summarization systems perform extractive summarizationapproach (selecting and photocopying extensive sentences from the professional documents). Though humans can cut and paste relevant data from a text, most of the times they rephrase sentences whenever necessary, orthey may join different related data into one sentence. The low inter-annotator agreement observed during manual evaluations suggest that the future of this research area massively depends on the capacity to find efficient ways of automatically evaluating the systems.

FUTURE WORK: Currently the system only deals with nouns while sentence scoring .Adjectives too play a major role in defining the important sentences. The future work

includes adding adjectives also to the along with nouns and then observe the effect on the summary generated. Also, graph based algorithms for sentence scoring are much more efficient but researches are going on this method .

6.REFERENCES

[1] A.L. Gorin, G. Riccardi and J.H. Wright, "How May I Help You?", *Speech Communication*, vol. 23, pp. 113–127.

[2] R. Iyer, H. Gish, D. McCarthy, "Unsupervised Training for Natural Language Call Routing", *ICASSP*, Orlando, Florida, vol. IV, pp. 3900–3903.

[3] P. Natarajan, R. Prasad, B. Suhm and D McCarthy, "Speech- Enabled Natural Language Call Routing: BBN Call Director", *ICSLP*, Denver, Colorado, pp. 1161–1164.

[4] Lori Lamel, Jean-Luc Gauvain, Gilles Adda, "Unsupervised Acoustic Model Training", *ICASSP*, Orlando, Florida, vol. I, pp. 877–880.

[5] Lori Lamel, Jean-Luc Gauvain, Gilles Adda, "Lightly supervised and unsupervised acoustic model training", *ComputerSpeech and Language*, vol. 16(1), pp. 115–129.

[6] DilekHakkani-T″ur, Giuseppe Riccardi and Allen Gorin, "Active Learning for Automatic Speech Recognition", *ICASSP*, Orlando, Florida, vol. IV, pp. 3904–3907.

[7] Jason D. M. Rennie and Ryan Rifkin, "Improving Multiclass Text Classification with the Support Vector Machine", Massachusetts Institute of Technolgy, *AI Memo AIM-2001-026*.

[8] McCallum, Andrew Kachites, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering", *http://www.cs.cmu.edu/ mccallum/bow*, 19/96

[9] Kadri Hacioglu,WayneWard, "A Concept Graph based Confidence Measure" , *ICASSP*, Orlando Florida, vol. I, pp. 225– 228.

[10] Bryan Pellom, Kadri Hacioglu, "Recent Improvements in the CU Sonic ASR System for Noisy Speech: The SPINE Task", *ICASSP*, Hong Kong, vol. I, pp. 4-7.

[11] Timothy Kuhn, Michele Jackson, "Accomplishing Knowledge: A Communicative Model of Knowledge Applied to a Call Center", *submitted to Communication Theory*

APPENDICES

```
import speech_recognition as sr
from os import path
from punctuator import Punctuator
import summarization
AUDIO_FILE = path.join(path.dirname(path.realpath(__file__)), "english.wav")
r = sr.Recognizer()#google offline speech recognization API
with sr.Audiofile(AUDIO_FILE) as source:
    audio = r.record(source)  # read the entire audio file
    try:
        text = r.recognize_google(audio) #gives text is in the format of string
        p = Punctuator('model.pcl')

        response = summarization.summarize(p.punctuate(text),2)
        print(response)
    except sr.UnknownValueError:
print("Google Speech Recognition could not understand audio")

    except sr.RequestError as e:
print("Could not request results from Google")
```

SUMMARIZATION

```
import sys
import nltk
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from gensim import corpora, models
import gensim
import numpy
def LDA(data) :#Latent Dirichlet Allocation
   texts = []
```

```
en_stop = stopwords.words('english')
  tokenizer = RegexpTokenizer(r'\w+')
en_stop = stopwords.words('english')
  tokens = tokenizer.tokenize(data)
stopped_tokens = [i for i in tokens if not i in en_stop]
texts.append(stopped_tokens)
  dictionary = corpora.Dictionary(texts)
  corpus = [dictionary.doc2bow(text) for text in texts]
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics=1, id2word =
dictionary, passes=100)
  list = ldamodel.show_topics(num_topics=20, formatted=False)
  topics = []
  for i in list[0][1]:
topics.append(i[0])
  return topics
def summarize(data, top_sentences, n=400, cluster_threshold=8):
   print(data)
   def score_sentences(sentences, important_words):
      scores = []
sentence_idx = -1
 for s in [nltk.tokenize.word_tokenize(s) for s in sentences]
sentence_idx += 1
word_idx = []
          for w in important_words:
             try:
word_idx.append(s.index(w))
             except ValueError as exc:
                pass
word_idx.sort()
          if len(word_idx)== 0: continue
          clusters = []
          cluster = [word_idx[0]]
i = 1
          while i<len(word_idx):
```

```
            if word_idx[i] - word_idx[i - 1] <cluster_threshold:
cluster.append(word_idx[i])
            else:
clusters.append(cluster[:])
                cluster = [word_idx[i]]
i += 1
clusters.append(cluster)
max_cluster_score = 0
          for c in clusters:
significant_words_in_cluster = len(c)
total_words_in_cluster = c[-1] - c[0] + 1
            score = 1.0 * significant_words_in_cluster \
               * significant_words_in_cluster / total_words_in_cluster

            if score >max_cluster_score:
max_cluster_score = score

scores.append((sentence_idx, score))

       return scores
   sentences = [s for s in nltk.tokenize.sent_tokenize(data)]
normalized_sentences = [s.lower() for s in sentences]

   words = [w.lower() for sentence in normalized_sentences for w in
nltk.tokenize.word_tokenize(sentence)]
top_n_words = LDA(data)
scored_sentences = score_sentences(normalized_sentences, top_n_words)
top_n_scored = sorted(scored_sentences, key=lambda s: s[1])[-top_sentences:]
top_n_scored = sorted(top_n_scored, key=lambda s: s[0])
   return dict(top_n_summary=[sentences[idx] for (idx, score) in top_n_scored])
```

Result:

```
My laptop is not working since past few days. I, don't know the cause of error.
If you can help me with that, it would be nice. My BIOS is giving me a message s
tating primary battery. Replace I, don't understand. What is that? Please help m
e.
{'top_n_summary': ['My laptop is not working since past few days.', 'My BIOS is
giving me a message stating primary battery.']}
>>>
```

Department of Computer Science and Systems Engineering

COLLEGE VISION & MISSION

VISION

To be one of the Nation's premier Engineering Colleges by achieving the highest order of excellence in Teaching and Research.

MISSION

- To foster intellectual curiosity, pursuit and dissemination of knowledge.
- To explore students' potential through academic freedom and integrity.
- To promote technical mastery and nurture skilled professionals to face competition in ever increasing complex world.

Department of Computer Science and Systems Engineering

DEPARTMENT VISION & MISSION

VISION

To become a Centre of excellence in Computer Sciences and Systems Engineering through teaching, training, research and innovation to produce high quality engineering professionals who can solve the growing complex problems of the society and Industry.

MISSION

- Established with the cause of development of technical education in advanced computer sciences and engineering with applications to systems there by serving the society and nation.
- Transfer of Knowledge through contemporary curriculum and fostering faculty and student development.
- Create keen interest for research and innovation among students and faculty by understanding the needs of the society and industry.
- Skill development among diversity of students in technical domains and profession for development of systems and processes to meet the demands of the industry and research.
- Imbibing values and ethics in students for prospective and promising engineering profession and develop a sense of respect for all.

Department of Computer Science and Systems Engineering

PROGRAM OUTCOMES

PO1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communicate effectively on complex engineering activities with the engineering

community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Department of Computer Science and Systems Engineering

PROGRAM SPECIFIC OUTCOMES

PSO1. Acquire knowledge of mathematics, Computer Science and Systems Engineering to solve complex engineering problems. (PO1)

PSO2. Identify, Analyze, Design among alternatives and Develop software for applications and systems in the domain of Computers and its based Systems to meet the societal needs. (PO2& PO3)

PSO3. Use the research-based knowledge and methods to solve real-world problems in the fields of Computer Science and Systems Engineering. (PO4)

PSO4. Apply appropriate techniques, use modern programing languages and packages to simulate and develop software by thoroughly understanding the requirements of the system and its constraints in Computer Science and Engineering. (PO5)

PROGRAM EDUCATIONAL OBJECTIVES

PEO1. Graduate will pursue advanced studies in expanses of Computer Science domain and Management.

PEO2. Graduates will evolve as entrepreneurs or be employed in reputed Software Industries and develop Quality Software Systems.

PEO3. Graduates will have career progression through professional skill development and continuing education with ethical attitude.

Department of Computer Science and Systems Engineering

COURSE OUTCOMES

CO1. Knowledge on the project topic.

CO2. Analytical ability exercised in the project work.

CO3. Design skills applied on the project topic.

CO4. Ability to investigate and solve complex engineering problems faced during the project work.

CO5. Ability to apply tools and techniques to complex engineering activities with an understanding of limitations in the project work.

CO6. Ability to provide solutions as per societal needs with consideration to health, safety, legal and cultural issues considered in the project work.

CO7. Understanding of the impact of the professional engineering solutions in environmental context and need for sustainable development experienced during the project work.

CO8. Ability to apply ethics and norms of the engineering practice as applied in the project work.

CO9. Ability to function effectively as an individual as experienced during the project work.

CO10. Ability to present views cogently and precisely on the project work.

CO11. Project management skills as applied in the project work.

CO12. Ability to engage in life-long leaning as experience during the project work.

Department of Computer Science and Systems Engineering

MAPPING OF COURSE OUTCOMES WITH PROJECT

CO4, CO8	Algorithmic approach	√
CO3, CO8	Simulation Model/Experiments Design	√
CO2	Identification of metrics related to the work	√
CO2,CO5	Comparison of results with rival methods	√
CO6	Society related Problem	√
CO9	Self Dependency in Performing Tasks/team work	√
CO12	Is the abstract appropriate to the project?	√
CO1	Introduction & literature Review	√
CO3, CO8	Modeling and Implementation	√
CO7	Results for sustainable development	√
CO11	Feasibility study ,Cost Model	√
CO10,CO12	Conclusions and Future work	√

SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)
SREE SAINATH NAGAR, TIRUPATI – 517 102

Department of Computer Science and Systems Engineering

Mapping of Course Outcomes with POs and PSOs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	√												√			
CO2		√												√		
CO3			√												√	
CO4				√												√
CO5					√											
CO6						√										
CO7							√									
CO8								√								
CO9									√							
CO10										√						
CO11											√					
CO12												√				