**Customer Care Email Topic Extraction**

Thesis submitted in partial fulfillment of the

requirements for the

**Post Graduate Diploma in Data Science**

By

**Eshita Gangwar**

18125760021

Under the guidance of

Dr. Subhabaha Pal

Senior Faculty

Manipal Academy of Higher Education

Bangalore

(If two guides are there indicate them side by side)



**MANIPAL ACADEMY OF HIGHER EDUCATION, MANIPAL**

**Customer Care Email Topic Extraction**

Thesis submitted in partial fulfillment of the

requirements for

**Post Graduate Diploma in Data Science**

By

**Eshita Gangwar**

18125760021

Under the guidance of

Dr. Subhabaha Pal

Senior Faculty

Manipal Academy of Higher Education

Bangalore



**MANIPAL ACADEMY OF HIGHER EDUCATION, MANIPAL**

**Customer Care Email Topic Extraction**

Thesis submitted in partial fulfillment of the

requirements for

**Post Graduate Diploma in Data Science**

By

**Eshita Gangwar**

18125760021

**Examiner 1**   **Examiner 2**

Signature: Signature:

Name: Name:



**MANIPAL ACADEMY OF HIGHER EDUCATION, MANIPAL**

**CERTIFICATE**

This is to certify that the project work titled

**Customer Care Email Topic Extraction**

is a bonafide record of the work done by

**Eshita Gangwar**

18125760021

In partial fulfillment of the requirements for the award of **Post Graduate Diploma in Data Science**under Manipal Academy of Higher Education, Manipal, Manipal and the same has not been submitted elsewhere for any kind of certification/recognition.

Dr. Subhabaha Pal

Senior Faculty

Manipal Academy of Higher Education

Bangalore

**Acknowledgments**

I would like to express my deep gratitude to Dr. Subhabaha Pal, my mentor for the project who clarified my understanding towards the data and guided me in data cleaning, data preparation and modeling.

I would also like thank the project mid- term review panelists which includes Mr. Mallikarjuna Doddamane and Mrs. Chandrika Bhargava for their valuable feedback and suggestions that I incorporated in my project.

My thanks are also extended to professors who taught us subjects in Data Science Diploma program specifically Exploratory Data Science, Data Visualization, Unstructured Data, and Machine Learning. The teachings in these subjects helped me to complete the project.

Finally, I wish to thank Manipal Global Academy team who supported with logistics and helped me in completing the project.

**Table of Contents**

1. **Acknowledgments**
2. **Table of contents**
3. **Abstract 2**
4. **List of Figures 3**
5. **Introduction 4**
   1. **Motivation**
   2. **Project scope**
   3. **Project Goal**
   4. **Literature/Market survey**
6. **Project Description 8**
   1. **Business/Domain Understanding**
   2. **Project stakeholders**
   3. **Datasets understanding**
   4. **Data Limitations**
   5. **Benefits of project**
7. **Exploratory Data Analysis 13**
   1. **Data exploration**
   2. **Data cleaning**
   3. **Data Transformation**
8. **Design 16** 
   1. **Analytical methods and Technology**
   2. **Data Visualization**
   3. **Short data snapshots**
   4. **Short code snippets**
9. **Modelling 30**
   1. **Challenges faced**
   2. **Model Interpretation**
   3. **Short data output**
10. **Key Results 32**
    1. **Output of intermediate steps**
    2. **Final outcome/Sample outputs**
    3. **Analysis of the results**
11. **Conclusion 33**
    1. **Summary of the project outcome**
    2. **Future work**
12. **References 34**

**3. Abstract**

Customer service centers are most crucial part of any company. They represent the company and communicate with the customers on its behalf. These canters also impart valuable information through customer feedback. As they form the bridge between the company and its customers, it is important they convey information timely and effectively. Emails are the most popular means of business communication. In order to efficiently and effectively utilize time, the customer service centres need to extract the relevant information from these emails. Then they need organizing it and promptly respond to the customers accordingly. In this project, we propose categorizing emails based on the customer mail. The topic of emails is determined along with its probability and also determines the topic of the email. After preliminary data pre-processing, we use the LDA algorithm based approach to classify the emails and then text summarization is performed. This way, the project helps to classify emails and determine their summary to save valuable time for the employees

**4. List of Figures**

1. Distribution of Document Word Counts
2. Word Cloud of Dataset
3. Number of Document per topic
4. Distribution of Document Word Counts by Dominant Topic
5. Word cloud for each Topic

**5. INTRODUCTION**

**5.1 Motivation**

Emails are convenient and the most popular means of communication for customers. They help in avoiding long waiting hours for telephonic conversation as well as in preserving the record of the communication that is happening between the customer service center and the customer. It is crucial that these emails be properly organized at the customer service so as to spontaneously and appropriately reply. Also, if properly utilized, these emails can yield invaluable data for the companies to access the needs and demands of the customers. As technology advances, the customers have more interactive and dynamic relation with the company. The amount of the emails received by customers at the customer service center is increasing rapidly. Handling this enormous amount of email manually can be a very time-consuming and complex task. The problem can be solved if emails are automatically classified on the basis of their content.

Massive amounts of data are collected on daily basis in so many companies. The huge amount of information makes it difficult to understand the data or to find what we are searching. We have to employ different methods to organize the data systematically and extract relevant information from it. Topic modeling is one such powerful technique which allows us to not only organize but also summarize large amount of textual data. It is helpful for determining different topical patterns present in the dataset which otherwise remain invisible.: Topic modelling is a method for finding subjects (i.e. topics) from a data that best describe the content present in the document. There are many methods which can be used to obtain topic models. For our project, we will be using LDA technique to find the topics of the emails at the customer service centers. This will help the employees to summarize the content of the emails.

Text summarization refers to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document. Automatic text summarization is a common problem in machine learning and natural language processing (NLP).

# Project Scope

Machine learning and natural language processing techniques give us the ability to extract hidden topics from large volumes of text. Topic modeling is useful when dealing with collections of documents that are simply too large for an individual to manually read and summaries. Examples of large volumes of text include online reviews, news articles and email conversations within an organization. Knowing the key topics of discussion is very useful for businesses that want to understand their customer’s problems or what they particularly like about the business. In this project we are dealing with the emails, and we want that without reading whole email we can get topic of the email along with the small summary. This will help in saving time. In this project we are using the customer care segment.

# 5.3 Project Goal

The aim of this project is :

* Identify the Topic of the mail.
* To whom the mail should be assigned.
* Email Summary

The project will be useful for the customer care executives in any company based on the text that is received. The project target is to create an application that analyses the text of the email to determine the topic of the mail along with the summary of the mail. These predicted and analyzed data can be observed by individual to know the email without even reading it.

# 5.4 Literature Survey

**5.4.1 Polarity Categorization on Product Reviews**

Mixymol (2017) showed a way to tackle the problem of sentiment polarity categorization. The dataset is collected from amazon.com. The dataset contains 376 instances of reviews of Nokia mobile in the form of a text file. Two classification algorithms namely Naive Bayes and Support Vector Machine Algorithms are taken to classify the reviews as positive, negative or neutral. The paper depicts the results from two different classifiers and a study is done on comparing one with the other.

**5.4.2 Automated Classification of Customer Emails via Association Rule Mining**

(Subramanian and Ramaraj, 2007)illustrated how to automatically route incoming emails to the person in charge using an association rule mining methodology and Apriori algorithm. One major advantage of the association rule-based classifier is that it does not assume that terms are independent which is a great advantage. Also, in this research work, the training of the association rule-based classifier is relatively fast. We will use a similar approach in our work to this methodology for topic identification.

**5.4.3 Emotion Detection in Email Customer Care**

Gupta et al. (2009) shows how to extract salient features and identify emotion in emails at customer service centers. these features show customer anger, dissatisfaction with the business, and warnings like take legal action, report to higher authorities or to leave. They also used supervised methods to predict emotional emails. We will use these approach of extracting salient features and for deciding the polarity of the email.

**5.4.4 Using Text Mining for Automated Customer Inquiry Classification**

In this paper, Jetley et al. (2015) describe an automated technique to evaluate customer inquiries and shows how this technique is helpful to classify customer inquiries. The application is illustrated through an example related to a leading multinational automation company. Here, the data is initially subjected to indexing. Indexing involves tokenization, pre-processing and attribute definition. Thereafter, Information Retrieval is performed in which relevant information from the text is extracted. This is followed by data analysis process. The results provide further insight into the types and trends within the available data. The research illustrated the use of customer inquiry classification to identify which product is being mentioned in the customer inquiry and intention analysis on customer inquiries to classify the customers intention. This paper described an automated method used to analyse customer inquiries which saves manual efforts.

**5.4.5 Finding Topics in Emails: Is LDA Enough?**

In this research, Joty et al. (2009) work helps to find topics in several email conversations at sentence level. They propose that the most commonly used topic models like Latent Dirichlet Allocation cannot be very successful as the characteristics emails and written monologues vary from each other. They demonstrate these important characteristics of emails which make them distinct from other written texts and also show why topic models alone may not be precise and accurate. An important aspect of their research is collaboration of strengths of both, unsupervised and supervised approach while finding features in the emails. They propose a solution where sentences in emails are characterized using three features: topic, conversation and lexical features for accurate analysis.

**5.4.6 A Survey of Topic Modelling in Text Mining**

Alghamdi and Alfalqi (2015) describe several Topic Modelling methods in this paper. They define topic as a cluster of words that have co-occurrence. According to them, topic modelling can associate words with similar meanings and distinguish between usages of words with more than one meanings. The different method explained in the paper is Latent Semantic Analysis, Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation (LDA), Correlated Topic Model (CTM) and Topic Evolution Model. The paper also describes the characteristics typical to each method along with their limitation. The paper also describes the differences among theoretical background and their applications. Also, the paper has discussed the evolution of topic modeling with time. They demonstrate how different papers consider time and use time as an important factor. Several papers have used different methods of model topic evolution. Some use continuous-time model while some use discretizing time. Citation relationship and time discretization have been used as well.

**5.4.7 Integrating Document Clustering and Topic Modeling**

Xie and P.Xing (2013) proposed a multi-grain clustering topic model (MGCTM) in their paper which collaborates document clustering and topic modeling into a single frame which performs the two separate tasks together to attain better performance.They perform the experiments on two most widely used datasets for clustering;Reuters-21578 and 20-Newsgroups datasets. The experimentation result shows that the clustering based topic models perform better and are more accurate. They ultimately conclude that how both of these methods are related and mutually beneficial.

**6. Project Description**

**6.1 Business/Domain Understanding**

Emails are convenient and the most popular means of communication for customers. They help in avoiding long waiting hours for telephonic conversation as well as in preserving the record of the communication that is happening between the customer service center and the customer. It is crucial that these emails be properly organized at the customer service so as to spontaneously and appropriately reply. Also, if properly utilized, these emails can yield invaluable data for the companies to access the needs and demands of the customers. As technology advances, the customers have more interactive and dynamic relation with the company. The amount of the emails received by customers at the customer service center is increasing rapidly. Handling this enormous amount of email manually can be a very time-consuming and complex task. The problem can be solved if emails are automatically classified on the basis of their content.

Massive amounts of data are collected on daily basis in so many companies. The huge amount of information makes it difficult to understand the data or to find what we are searching. We have to employ different methods to organize the data systematically and extract relevant information from it.

* 1. **Project stakeholders**

The project’s main stakeholders are the multi-national companies with customer care services that receive emails. These companies have various products. They will receive tons of email regarding different products. To read each and every mail will be tiresome and time consuming. Our model is generalized and can adapt to the situation and if we change the dataset it will work fine.

* 1. **Datasets understanding**

We’re going to use the 20-Newsgroups dataset. The raw data is a 22.2 MB JSON file with discussions scraped from 20 different Newsgroups discussion boards. While we don’t specify desired labels for the topics, it’s useful in our case to use a data set that we know has an underlying structure with distinct topics of discussion. We would expect that if LDA works as intended, it would be able to separate out the topics with a similar structure to each discussion group

The topics of the discussion include the following:

* comp.graphics
* comp.os.ms-windows.misc
* comp.sys.ibm.pc.hardware
* comp.sys.mac.hardware
* comp.windows.xrec.autos
* rec.motorcycles
* rec.sport.baseball
* rec.sport.hockeysci.crypt
* sci.electronics
* sci.med
* sci.spacemisc.forsaletalk.politics.misc
* talk.politics.guns
* talk.politics.mideasttalk.religion.misc
* alt.atheism
* soc.religion.christian

**6.4 Data Limitation**

We often want to quickly get an overview of what different texts are about—for example, to decide what to read in-depth or to simply classify the texts. [Topic modelling](https://www.sciencedirect.com/topics/computer-science/topic-modeling) is an [automatic approach](https://www.sciencedirect.com/topics/computer-science/automatic-approach) that attempts to extract the most important topics per text document. The basic assumption of [topic modelling](https://www.sciencedirect.com/topics/computer-science/topic-modeling) is that documents are created using a set of topics the authors want to describe and discuss in the documents. The topics might, however, not be explicitly specified in the documents, and might remain only implicitly in the heads of the authors. Nevertheless, for each topic, the authors still use certain words in the documents. Therefore, for this analysis, we say that a topic is formed by a set of related words. Hence, there are probabilities with which certain words appear in the context of several topics. Topic modelling makes use of this by aiming to extract these probabilities and thereby recreating the topics. Mathematically, we need an [algorithm](https://www.sciencedirect.com/topics/computer-science/algorithms) which is able to group the words extracted from documents into probable topics. The most common one used is LDA, but there are others to choose from. The concrete algorithm is mostly uninteresting for the user of the topic modelling method, because all algorithms are not exact. Topic modelling can give only a rough idea of what the main topics consisting of important words are. The further analysis and interpretation needs manual effort. Yet, especially for larger text corpora, topic modelling can be an interesting pre-analysis before manual coding. The topics found can form initial ideas for coding, and we can mark context in which they were found to be checked in detail by the coder.

The other limitation in using this algorithm is that, not a well-defined generative model: by this we mean if our model is given the new document whose topic distribution is different. In other word we can say if we are given with the unseen document with different matter it is subject to misclassify them.

**7. Exploratory Data Analysis**

**7.1 Data exploration**

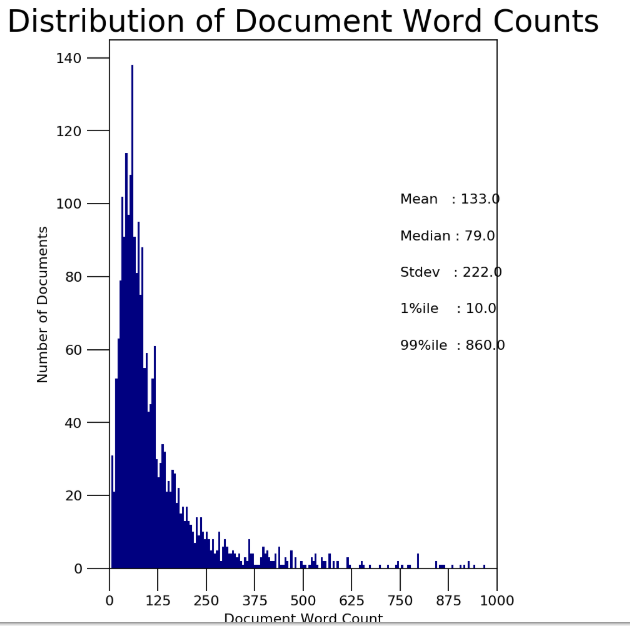
****

Fig 1: Distribution of Document Word Counts

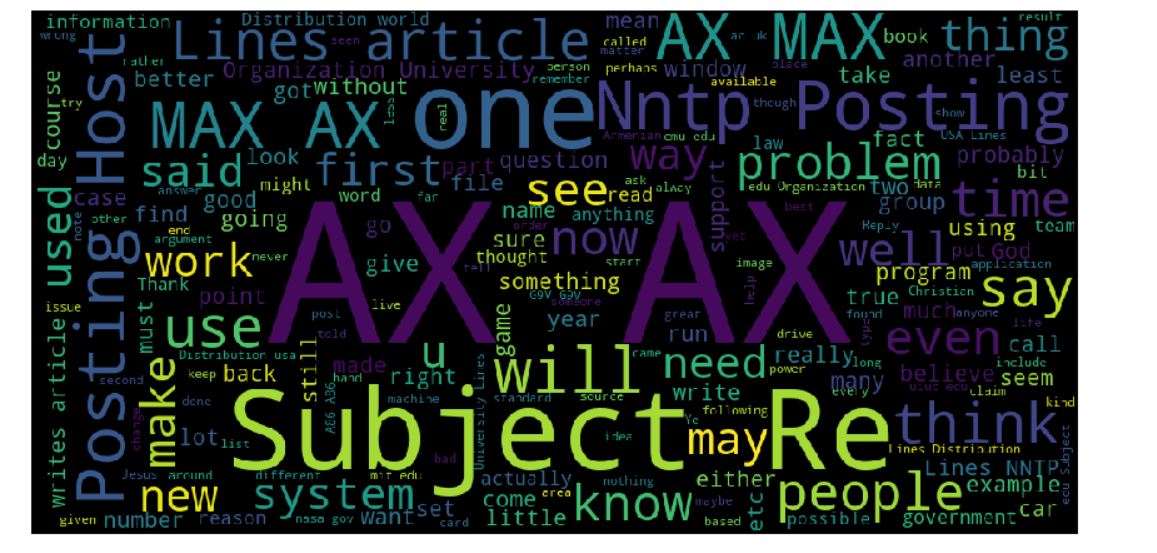


Fig 2: Word Cloud of Dataset

**7.2 Data cleaning**

In our project, the email can be viewed as a set of sentences. The purpose of our project is to help find the topic of the email and also determine the summary of the email. The first procedure is of topic determination. In this module, the email dataset is converted into a form which is easy to understand for finding the probable topic. Thus data pre-processing is performed. Firstly, we will employ the technique known as the tokenization. All the tokens that have been extracted may not be useful for our classification. Therefore the unnecessary words, known as the stop words, are eliminated using the technique known as stop words removal. The remaining tokens are subjected to the process of stemming. Stemming attempts to reduce a word to its root form. Therefore, the document is then represented in the form of root words rather than the original words. Other method called Part of speech tagging is then applied. POS tagging associates a word in the test to the corresponding part of speech.

**7.2.1 Tokenize Sentences.**

**def** sent\_to\_words(sentences):

print(sentences)

**for** sent **in** sentences:

sent = re.sub('\S\*@\S\*\s?', '', sent)

sent = re.sub('\s+', ' ', sent)

sent = re.sub("**\'**", "", sent)

sent = gensim.utils.simple\_preprocess(str(sent), deacc=**True**)

**yield**(sent)

*#Convert to list*

data = df.content.values.tolist()

data\_words = list(sent\_to\_words(data))

**7.2.2 Lemmatize and Stopword Removal.**

**def** process\_words(texts, stop\_words=stop\_words, allowed\_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):

texts = [[word **for** word **in** simple\_preprocess(str(doc)) **if** word **not** **in** stop\_words] **for** doc **in** texts]

texts\_out = []

nlp = spacy.load('en', disable=['parser', 'ner'])

**for** sent **in** texts:

doc = nlp(" ".join(sent))

texts\_out.append([token.lemma\_ **for** token **in** doc **if** token.pos\_ **in** allowed\_postags])

texts\_out = [[word **for** word **in** simple\_preprocess(str(doc)) **if** word **not** **in** stop\_words] **for** doc **in** texts\_out]

**return** texts\_out

data\_ready = process\_words(data\_words)

**7.3 Data Transformation**

# Create Dictionary

id2word = corpora.Dictionary(data\_ready)

# Create Corpus: Term Document Frequency

corpus = [id2word.doc2bow(text) for text in data\_ready]

1. **Design**
   1. **Analytical methods and Technology used**

# 8.1.2 Topic Modelling

Analytics Industry is all about obtaining the “Information” from the data. With the growing amount of data in recent years, that too mostly unstructured, it’s difficult to obtain the relevant and desired information. But, technology has developed some powerful methods which can be used to mine through the data and fetch the information that we are looking for.

One such technique in the field of text mining is Topic Modelling. As the name suggests, it is a process to automatically identify topics present in a text object and to derive hidden patterns exhibited by a text corpus. Thus, assisting better decision making.

Topic Modelling is different from rule-based text mining approaches that use regular expressions or dictionary based keyword searching techniques. It is an unsupervised approach used for finding and observing the bunch of words (called “topics”) in large clusters of texts.

Topics can be defined as “a repeating pattern of co-occurring terms in a corpus”. A good topic model should result in – “health”, “doctor”, “patient”, “hospital” for a topic – Healthcare, and “farm”, “crops”, “wheat” for a topic – “Farming”.

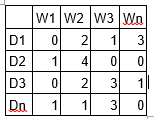
Topic Models are very useful for the purpose for document clustering, organizing large blocks of textual data, information retrieval from unstructured text and feature selection. Various professionals are using topic models for recruitment industries where they aim to extract latent features of job descriptions and map them to right candidates. They are being used to organize large datasets of emails, customer reviews, and user social media profiles.

## Latent Dirichlet Allocation for Topic Modeling

There are many approaches for obtaining topics from a text such as – Term Frequency and Inverse Document Frequency. Latent Dirichlet Allocation is the most popular topic modeling technique and in this article, we will discuss the same.

LDA assumes documents are produced from a mixture of topics. Those topics then generate words based on their probability distribution. Given a dataset of documents, LDA backtracks and tries to figure out what topics would create those documents in the first place.

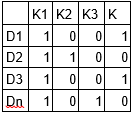
LDA is a matrix factorization technique. In vector space, any corpus (collection of documents) can be represented as a document-term matrix. The following matrix shows a corpus of N documents D1, D2, D3 … Dn and vocabulary size of M words W1,W2 .. Wn. The value of i,j cell gives the frequency count of word Wj in Document Di.

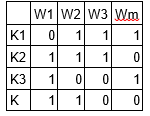
[](https://www.analyticsvidhya.com/wp-content/uploads/2016/08/Modeling2.png)

LDA converts this Document-Term Matrix into two lower dimensional matrices

M1

M2  
M1 is a document-topics matrix and M2 is a topic – terms matrix with dimensions (N,  K) and (K, M) respectively, where N is the number of documents, K is the number of topics and M is the vocabulary size.

[](https://www.analyticsvidhya.com/wp-content/uploads/2016/08/modeling3.png)

[](https://www.analyticsvidhya.com/wp-content/uploads/2016/08/Modeling4.png)

These two matrices already provides topic word and document topic distributions, However, these distribution needs to be improved, which is the main aim of LDA. LDA makes use of sampling techniques in order to improve these matrices.

It Iterates through each word “w” for each document “d” and tries to adjust the current topic – word assignment with a new assignment. A new topic “k” is assigned to word “w” with a probability P which is a product of two probabilities p1 and p2.

For every topic, two probabilities p1 and p2 are calculated. P1 – p(topic t / document d) = the proportion of words in document d that are currently assigned to topic t. P2 – p(word w / topic t) = the proportion of assignments to topic t over all documents that come from this word w.

The current topic – word assignment is updated with a new topic with the probability, product of p1 and p2 . In this step, the model assumes that all the existing word – topic assignments except the current word are correct. This is essentially the probability that topic t generated word w, so it makes sense to adjust the current word’s topic with new probability.

After a number of iterations, a steady state is achieved where the document topic and topic term distributions are fairly good. This is the convergence point of LDA.

# 8.1.2 Text Summarization

Automatic Text Summarization is one of the most challenging and interesting problems in the field of Natural Language Processing (NLP). It is a process of generating a concise and meaningful summary of text from multiple text resources such as books, news articles, blog posts, research papers, emails, and tweets.The demand for automatic text summarization systems is spiking these days thanks to the availability of large amounts of textual data.We will be focusing on the **extractive summarization** technique. These methods rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary. Therefore, identifying the right sentences for summarization is of utmost importance in an extractive method.

### TextRank Algorithm

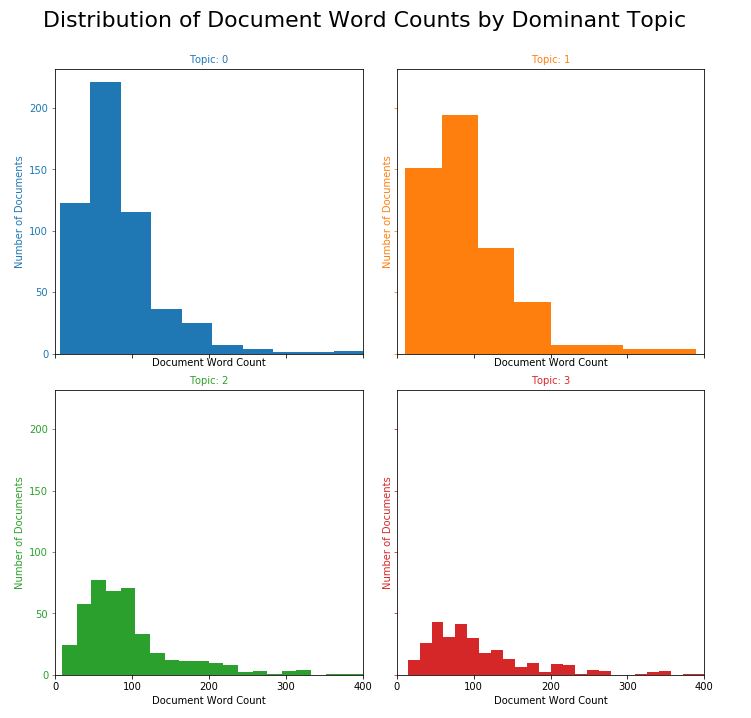
* The first step would be to concatenate all the text contained in the articles
* Then split the text into individual sentences
* In the next step, we will find vector representation (word embeddings) for each and every sentence
* Similarities between sentence vectors are then calculated and stored in a matrix
* The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation
* Finally, a certain number of top-ranked sentences form the final summary
  1. **Data Visualization**

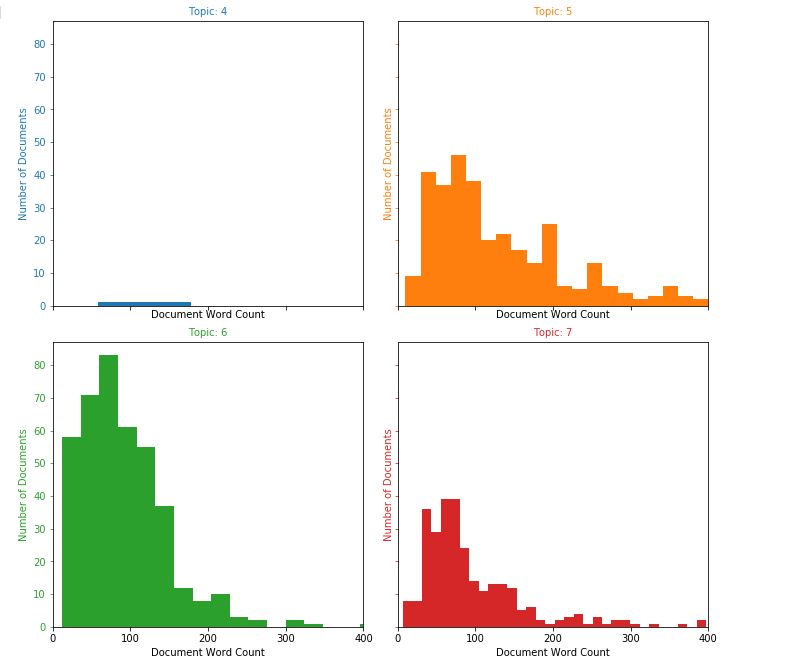
# Number of Document per topic

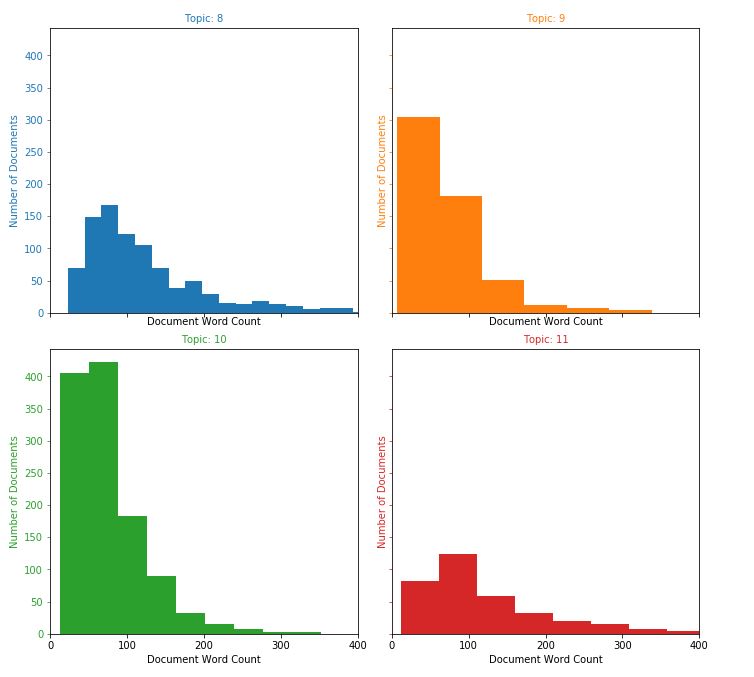
# download.png

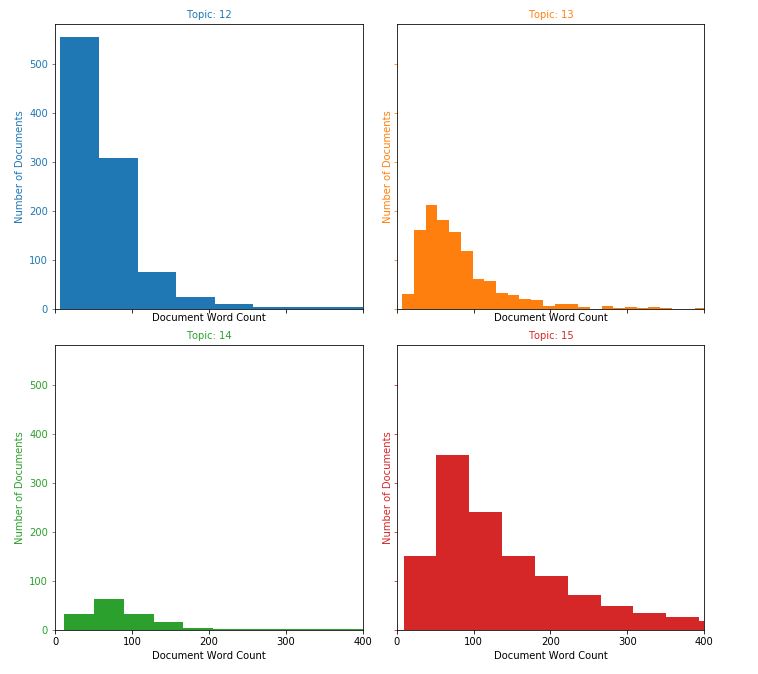
Fig 3 : Number of Document per topic

From the above graph we can see the Topic 15 has the highest number of document or email. In other words we can say it is most discussed among the whole dataset. The Topic 4 is least discussed as the number of document it contains is less than 50.









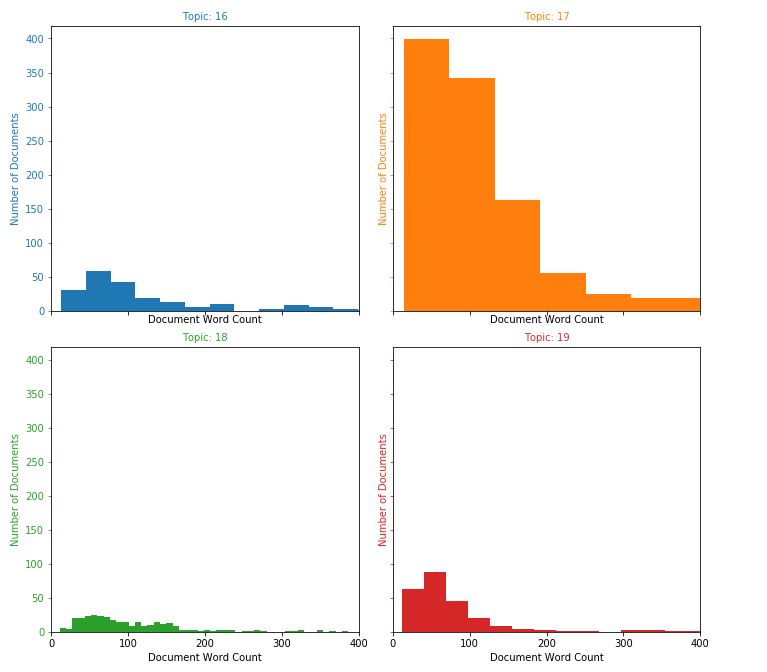


Fig 4: Distribution of Document Word Counts by Dominant Topic

From the above figure we can depicts that the topic 0 has the document which are lengthiest and topic 4 are short. As we have seen that the topic 4 has the least number of the document one reason can be that they are short.

Word cloud for the better interpretation



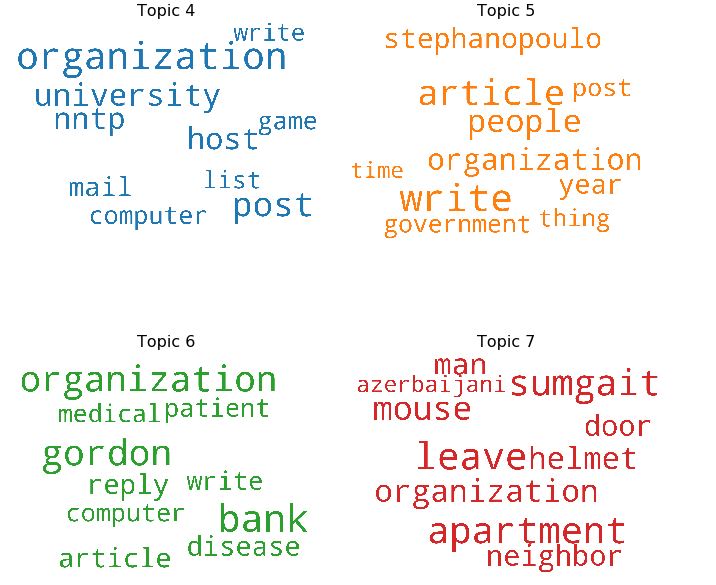
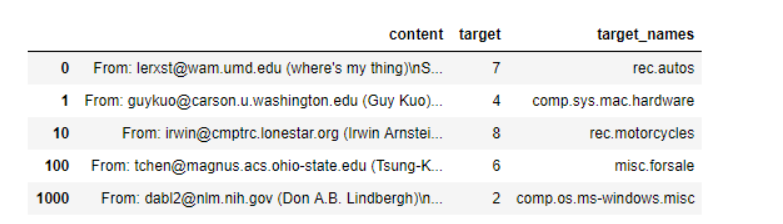


Fig 5: Word cloud for each Topic.

These can be used to depict the topic to the relatable topics which can be easily interpretable. For example Topic zero is related to university, car, and problem. Similarly we can draw conclusion for other topic also.

* 1. **Short data snapshots**



* 1. **Short code snippets**

# Create Dictionary

id2word = corpora.Dictionary(data\_ready)

# Create Corpus: Term Document Frequency

corpus = [id2word.doc2bow(text) for text in data\_ready]

# Build LDA model

lda\_model = gensim.models.ldamodel.LdaModel(corpus=corpus,

id2word=id2word,

num\_topics= 20)

pprint(lda\_model.print\_topics())

## Calculating cosine similarity

**def** similar(sent1, sent2, stopwords=**None**):

**if** stopwords **is** **None**:

stopwords = []

sent1 = [w.lower() **for** w **in** sent1]

sent2 = [w.lower() **for** w **in** sent2]

all\_words = list(set(sent1 + sent2))

vector1 = [0] \* len(all\_words)

vector2 = [0] \* len(all\_words)

**for** w **in** sent1:

**if** w **in** stopwords:

**continue**

vector1[all\_words.index(w)] += 1

**for** w **in** sent2:

**if** w **in** stopwords:

**continue**

vector2[all\_words.index(w)] += 1

**return** 1 - cosine\_distance(vector1, vector2)

**Smilarity\_matrix**

**def** matrix(sentences, stop\_words):

similarity\_matrix = np.zeros((len(sentences), len(sentences)))

**for** idx1 **in** range(len(sentences)):

**for** idx2 **in** range(len(sentences)):

**if** idx1 == idx2:

**continue**

similarity\_matrix[idx1][idx2] = similar(sentences[idx1], sentences[idx2], stop\_words)

**return** similarity\_matrix

**9. Modelling**

**9.1 Challenges Faced**

The data we are using is for the customer care email classification, the dataset for the particular companies is unavailable hence we are using the newsgroup data that is pre-labeled data, with the 20 discussion board i.e related to the particular topic. We are assuming in the similar manner our companies dataset will also work. The other challenge that we are facing that dataset is very noisy. It needs lot of data cleaning as unwanted characters are available. The number of topics for the data is not well defined we need to select as per requirement.

* 1. **Model interpretation**

Our model will give the topic which will have the highest probability for that particular email or article to which it is related to. If needed it will also show the terms related to this topic. In the second part it will also provide the short summary for the same.

* 1. **Short Data output**

[(0,

'0.009\*"organization" + 0.008\*"write" + 0.006\*"article" + 0.006\*"post" + '

'0.005\*"time" + 0.004\*"university" + 0.004\*"host" + 0.004\*"nntp" + '

'0.004\*"day" + 0.003\*"world"'),

(1,

'0.017\*"file" + 0.010\*"program" + 0.009\*"window" + 0.007\*"server" + '

'0.006\*"include" + 0.006\*"write" + 0.005\*"font" + 0.005\*"set" + '

'0.005\*"version" + 0.005\*"available"'),

(2,

'0.010\*"write" + 0.008\*"organization" + 0.008\*"time" + 0.006\*"article" + '

'0.006\*"post" + 0.004\*"read" + 0.004\*"university" + 0.004\*"look" + '

'0.003\*"way" + 0.003\*"host"'),

(3,

'0.034\*"key" + 0.012\*"chip" + 0.009\*"clipper" + 0.008\*"encryption" + '

'0.008\*"escrow" + 0.006\*"organization" + 0.006\*"bit" + 0.005\*"post" + '

'0.005\*"algorithm" + 0.005\*"encrypt"'),

(4,

'0.789\*"ax" + 0.057\*"max" + 0.004\*"tm" + 0.004\*"bhj" + 0.003\*"giz" + '

'0.003\*"qax" + 0.001\*"gq" + 0.001\*"nrhj" + 0.001\*"biz" + 0.001\*"mg"'),

(5,

'0.017\*"armenian" + 0.008\*"turkish" + 0.006\*"write" + 0.006\*"people" + '

'0.005\*"article" + 0.005\*"government" + 0.005\*"turk" + 0.005\*"organization" '

'+ 0.004\*"armenia" + 0.004\*"turkey"'),

(6,

'0.022\*"car" + 0.008\*"organization" + 0.007\*"write" + 0.006\*"article" + '

'0.005\*"look" + 0.005\*"drive" + 0.004\*"problem" + 0.004\*"university" + '

'0.004\*"tire" + 0.004\*"engine"'),

(7,

'0.009\*"pt" + 0.008\*"play" + 0.008\*"write" + 0.007\*"organization" + '

'0.006\*"game" + 0.006\*"period" + 0.006\*"year" + 0.005\*"article" + '

'0.005\*"point" + 0.004\*"team"'),

(8,

'0.010\*"write" + 0.009\*"article" + 0.007\*"people" + 0.006\*"organization" + '

'0.006\*"stephanopoulo" + 0.005\*"year" + 0.004\*"post" + 0.004\*"government" + '

'0.004\*"thing" + 0.004\*"time"'),

(9,

'0.011\*"organization" + 0.009\*"wire" + 0.007\*"image" + 0.007\*"write" + '

'0.006\*"post" + 0.006\*"scsi" + 0.005\*"nntp" + 0.005\*"wiring" + '

'0.005\*"system" + 0.005\*"host"'),

(10,

'0.020\*"window" + 0.014\*"drive" + 0.013\*"organization" + 0.009\*"problem" + '

'0.009\*"disk" + 0.008\*"post" + 0.008\*"host" + 0.007\*"write" + 0.007\*"nntp" + '

'0.006\*"university"'),

(11,

'0.010\*"president" + 0.009\*"state" + 0.007\*"government" + 0.006\*"mr" + '

'0.006\*"law" + 0.006\*"american" + 0.005\*"people" + 0.004\*"write" + '

'0.004\*"policy" + 0.004\*"organization"'),

(12,

'0.014\*"organization" + 0.011\*"post" + 0.009\*"university" + 0.009\*"host" + '

'0.009\*"nntp" + 0.007\*"mail" + 0.006\*"game" + 0.006\*"computer" + '

'0.006\*"list" + 0.005\*"write"'),

(13,

'0.012\*"organization" + 0.008\*"card" + 0.007\*"write" + 0.007\*"post" + '

'0.006\*"system" + 0.006\*"article" + 0.006\*"university" + 0.005\*"new" + '

'0.005\*"speed" + 0.005\*"host"'),

(14,

'0.006\*"leave" + 0.006\*"apartment" + 0.006\*"sumgait" + 0.005\*"mouse" + '

'0.004\*"organization" + 0.004\*"helmet" + 0.004\*"man" + 0.004\*"door" + '

'0.003\*"neighbor" + 0.003\*"azerbaijani"'),

(15,

'0.010\*"people" + 0.008\*"god" + 0.007\*"write" + 0.006\*"christian" + '

'0.006\*"believe" + 0.004\*"organization" + 0.004\*"question" + 0.004\*"give" + '

'0.004\*"article" + 0.004\*"time"'),

(16,

'0.023\*"space" + 0.008\*"orbit" + 0.008\*"launch" + 0.007\*"mission" + '

'0.007\*"nasa" + 0.006\*"satellite" + 0.006\*"datum" + 0.005\*"earth" + '

'0.004\*"image" + 0.004\*"spacecraft"'),

(17,

'0.011\*"team" + 0.008\*"gun" + 0.008\*"year" + 0.007\*"organization" + '

'0.007\*"game" + 0.007\*"player" + 0.007\*"write" + 0.006\*"article" + '

'0.006\*"hockey" + 0.005\*"league"'),

(18,

'0.008\*"msg" + 0.008\*"food" + 0.007\*"organization" + 0.005\*"circuit" + '

'0.005\*"ground" + 0.005\*"write" + 0.005\*"article" + 0.004\*"point" + '

'0.004\*"thing" + 0.004\*"work"'),

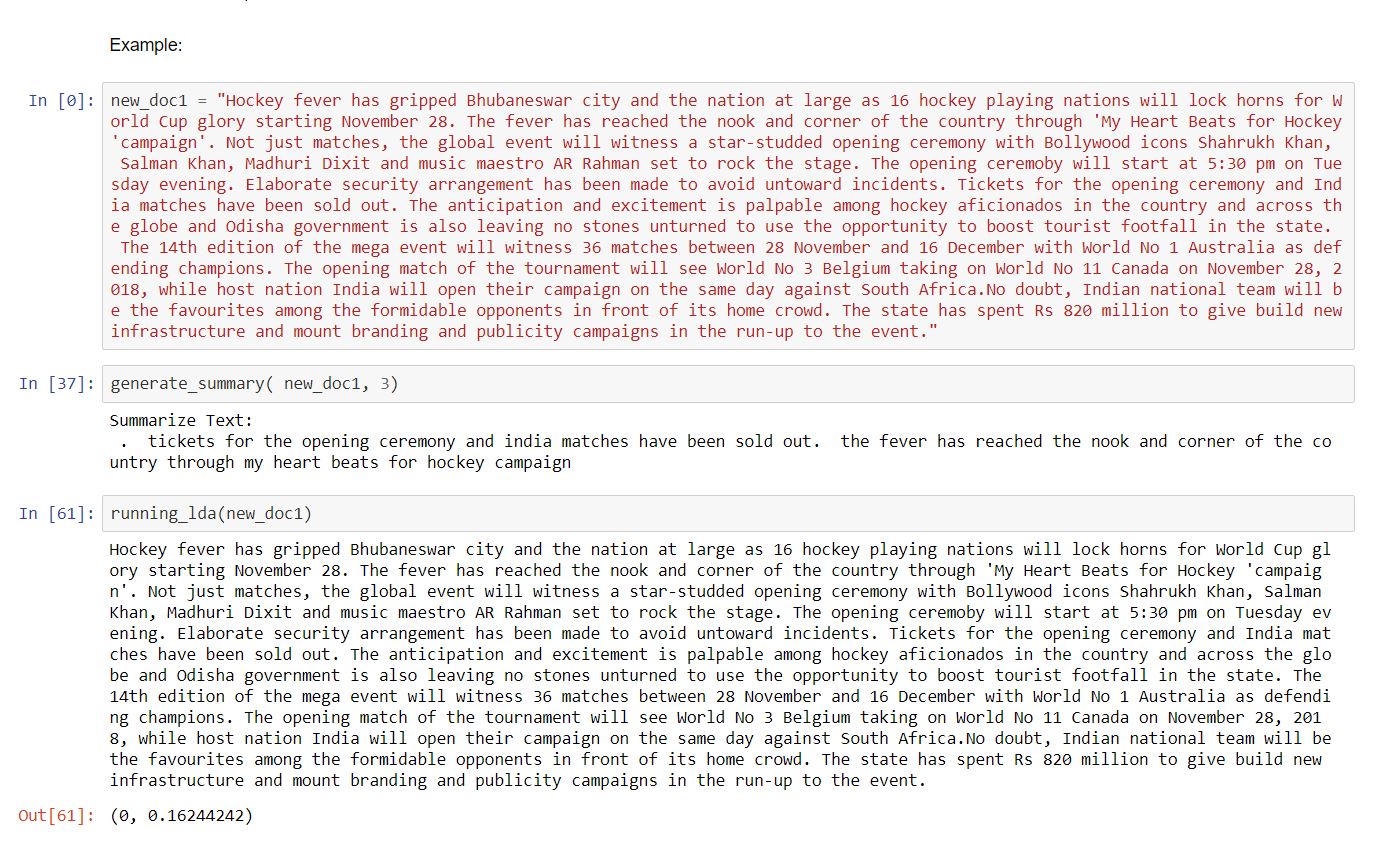
(19,

'0.011\*"bank" + 0.010\*"gordon" + 0.010\*"organization" + 0.005\*"reply" + '

'0.005\*"disease" + 0.005\*"article" + 0.005\*"write" + 0.004\*"patient" + '

'0.004\*"computer" + 0.004\*"medical"')]

**10. Key Result**



In this we can see that new\_doc1 contains the new email which is quite large. The function generate\_summary() will give the short summary where as running\_lda() will give us the topic number.

**11. Conclusion**

**11. 1 Summary**

The research paper focuses on quick and efficient email classification at customer service centers. The topic of the email is determined through topic modeling using LDA technique. Ultimately, the outcome will display the summary of the email and suggested topics based on their content. The combination of outputs from both the modules (Summary and Topic) will help the customer service employees to summarize the incoming emails quickly and grasp the information quickly.

Our work helps the customer service employees to get an overview of the content of the incoming customer emails which further helps them to organize these emails according to their needs and/or interests. It also helps to reduce their corresponding response time and to analyze the customer feedback on a product or its parts.

**11.2 Future Work**

The work can be extended by adding some functionality for the user, such as an interface, so the complaints can be arranged according to the priority. The project can also help in further categorization depending on the departments within the companies. The emails at the customer service centers can be handled automatically. For this, templates can be created which will help to respond to distinctive types of emails appropriately. Also, in future more data analysis can be performed on the data derived from customer service centers as they can serve as data sources to analyze customer needs and future trends.

**12. References**

1. Alghamdi, R. and Alfalqi, K. (2015). “A survey of topic modeling in text mining.”

International Journal of Advanced Computer Science and Applications, 6, 147–153.

2. DâA ̆ ZAndrea, A., Ferri, F., Grifoni, P., and Guzzo, T. (2015). “Approaches, tools and ́applications for sentiment analysis implementation.” International Journal of Computer Applications, 125.

3. Fang, X. and Zhan, J. (2015). “Approaches, tools and applications for sentiment analy-

sis implementation.” Journal of Big Data.

4. Gupta, N., Gilbert, M., Fabbrizio, G. D., Wu, Z., and Serker, N. H. M. K. (2009).

“Emotion detection in email customer care.” Proceedings of the NAACL HLT 2010

Workshop on Computational Approaches to Analysis and Generation of Emotion in

Text, 10–16.

5. Jetley, R. P., Gugaliya, J. K., and Javed, S. (2015). “Using text mining for automated

customer inquiry classification.” Journal of Vibration and Acoustics, 122, 46–51.

6. Joty, S., Carenini, G., Murray, G., and Ng, R. (2009). “Finding topics in emails: Is lda

enough?.” NIPS-2009 workshop on applications for topic models: text and beyond.

7. Mixymol, V. (2017). “Polarity categorization on product reviews.” SInternational Jour-

nal on Recent and Innovation Trends in Computing and Communication, 5, 28–31.

8. Subramanian, S. K. and Ramaraj, N. (2007). “Automated classification of customer

emails via association rule mining.” Information Technology Journal, 6, 567–572.

9. Xie, P. and P.Xing, E. (2013). “Integrating document clustering and topic modeling.”

Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, 29.

10. Blei, D.M., and Lafferty, J. D. ―Dynamic Topic Models‖, Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006.

11. Steyvers, M., and Griffiths, T. (2007). ―Probabilistic topic models‖. In T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds), Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum

12. Hofmann, T., ―Unsupervised learning by probabilistic latent semantic analysis‖, Machine Learning, 42 (1), 2001, 177- 196.

13. Kakkonen, T., Myller, N., Sutinen, E., and Timonen, J., ―Comparison of Dimension Reduction Methods for Automated Essay Grading‖, Educational Technology &Society, 11 (3), 2008, 275-288.

14. Liu, S., Xia, C., and Jiang, X., ―Efficient Probabilistic Latent Semantic Analysis with Sparsity Control‖, IEEE International Conference on Data Mining, 2010, 905-910.

15. Bassiou, N., and Kotropoulos C. ―RPLSA: A novel updating scheme for Probabilistic Latent Semantic Analysis‖, Department of Informatics, Aristotle