Design and development of Travel Guide using Wikipedia text

FIFTH REVIEW REPORT

Submitted by,

CB.EN.P2EBS18006 Sejal Nilkanth Badgujar

in partial fulfillment for the award of the degree

of

MASTER OF TECHNOLOGY

IN

EMBEDDED SYSTEMS



Department of Electrical and Electronics Engineering

AMRITA SCHOOL OF ENGINEERING, COIMBATORE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641112 (INDIA)

MAY – 2020

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE, 641112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled “**DESIGN AND DEVELOPMENT OF TRAVEL GUIDE USING WIKIPEDIA TEXT**” submitted by “**SEJAL NILKANTH BADGUJAR (REG NO: CB.EN.P2EBS18006)**” in partial fulfillment of the requirements for the award of the **Degree** **Master of Technology** in “**EMBEDDED SYSTEMS**” is a bonafide record of the work carried out under our guidance and supervision at Amrita School of Engineering,Coimbatore.

|  |  |  |
| --- | --- | --- |
| Dr. Anju S. Pillai | Dr. S. Balamurugan | Mr. Sagar Patil and  Mr. Sudhanshu Trivedi |
| SUPERVISOR | CHAIRPERSON | EXTERNAL GUIDE |
| Assistant Professor | Associate Professor | Senior Software Engineer (Big Data and Machine learning) |
| Dept. of EEE | Dept. of EEE | Glisten Pvt. Ltd |
| Amrita Vishwa | Amrita Vishwa | Pune |
| Vidyapeetham | Vidyapeetham |  |
| Coimbatore | Coimbatore |  |

This Project was evaluated by us on

INTERNAL EXAMINER EXTERNAL EXAMINER

# AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING.**

# DECLARATION

## I, SEJAL NILKANTH BADGUJAR (Reg. No. CB.EN.P2EBS18006) hereby declare that this project report, entitled “Design and development of Travel Guide using Wikipedia text”, is a record of the original work done by me under the guidance of Dr. Anju S. Pillai, Assistant Professor, Department of Electrical and Electronics Engineering, Amrita School of Engineering, Coimbatore and that this work has not formed the basis for the award of any degree/diploma/associateship/fellowship or a similar award, to any candidate in any University, to the best of my knowledge.

Place: Coimbatore Signature of the student

Date:

COUNTERSIGNED

## Dr. Anju S. Pillai

SUPERVISOR

Assistant Professor

Department of Electrical & Electronics Engineering

Amrita School of Engineering

Coimbatore – 641112

**ABSTRACT**

When one plans for a trip, he needs to search a lot about the main attractions, nearby places to visit, about climate, food. If it’s a historical place, them what is the history, how many battles happened, who won, etc. All these questions can be answered only if he reads multiple sites and go through multiple webpages.

To make this task easier and interesting, this project provides all the travel related information on a single webpage/app. One need not to go through all the history pages, he can find everything at one place. The information will be portrayed in very interesting manner, so that person will be satisfied with the knowledge and scenery of that place. The main objective of the project is to gather all important information at one place which will be very easily readable and interesting. Majority of the information is available on Wikipedia pages, so this project fetches information from Wikipedia, webpages, etc.

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| ABSTRACT |  |
| CONTENTS |  |
| LIST OF FIGURES |  |
| 1. INTRODUCTION |  |
| 1.1 Introduction | 1 |
| 1.2 Objective | 2 |
|  |  |
| 2. SYSTEM OVERVIEW AND METHODOLOGY |  |
| 2.1 Introduction | 3 |
| 2.1 Methodology | 3 |
|  |  |
| 3. TOPIC MODELING |  |
| 3.1 Introduction | 7 |
| 3.2 Latent Dirichlet Allocation (Lda) | 7 |
|  |  |
| 4. WORD EMBEDDING | 7 |
| 4.1 Introduction | 8 |
| 4.2 Word2vec | 8 |
| 4.3 Doc2vec | 10 |
|  |  |
| 5. CLASSIFICATION |  |
| 5.1 Introduction | 11 |
| 5.2 K-Means Clustering | 11 |
| 5.3 Timeline Based Clustering | 12 |
| 5.4 Tagging | 12 |
|  |  |
| 6. ALGORITHM IMPLEMENTAION |  |
| 6.1 Introduction | 13 |
| 6.2 Work Done | 13 |
|  |  |
| 7. RESULTS |  |
| 7.1 Introduction | 14 |
| 7.2 Similarity Sets | 14 |
| 7.3 Topic Modelling | 14 |
| 7.4 K-Means Clustering | 15 |
| 7.5 Timeline Wise Classification | 14 |
| 7.6 Tagging | 15 |
|  |  |
| 8. CONCLUSION | 18 |
|  |  |
| References | 19 |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| 1.1 Holidify screenshot | 1 |
| 1.2 Transindiatravel | 1 |
| 2.1 System overview | 3 |
| 2.2 Methodology | 4 |
| 2.3 Cosine similarity | 5 |
| 3.1 CBOW model | 8 |
| 3.2 Skip gram model | 9 |
| 6.1 Overall work done | 13 |
| 7.1 Summarized text with similarity sets | 14 |
| 7.2 Topic modeling – LDA | 15 |
| 7.3 K-means clustering | 15 |
| 7.4 Timeline wise classifications | 16 |
| 7.5 Word Cloud for proximity words of 1664 | 16 |
| 7.5 Tagging | 17 |

|  |
| --- |
|  |

**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

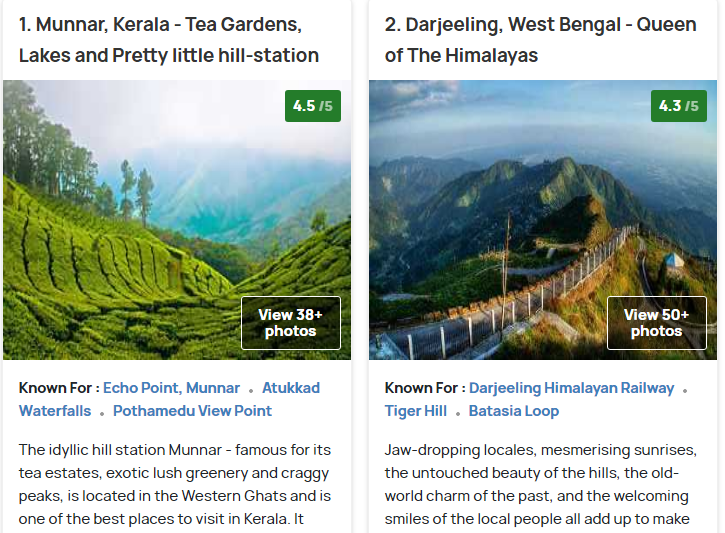
Whenever a traveller is travelling, one needs all the information about the place to explore it fully. All the information is available on multiple sites now-a-days like holidyfy.com or transindiatravel.com like below figures. One needs to plan his tour wisely by covering all the nearby attraction points. Also interested people always seek for the historical background of the visiting place.

Figure 2.1 Holidify screenshot

<https://www.holidify.com/collections/tourist-places-in-india>

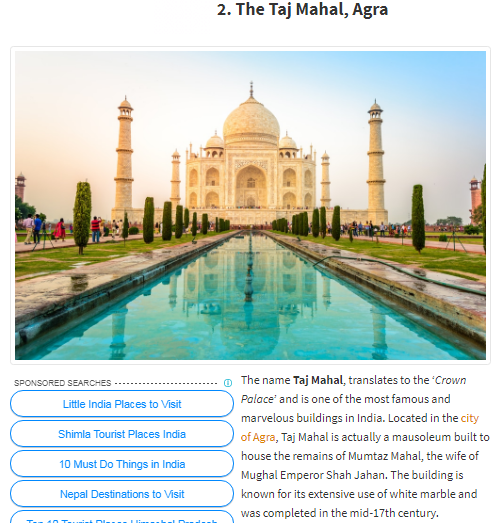


Figure 1.2 Transindiatravel

<http://www.transindiatravels.com/india/best-places-to-visit-in-india/>

These sites provide information but to get all the detailed information. To explore details of a place, one needs to go through many sites to collect details of the transportation, climate, food, history, nearby attractions, etc. But all this information is not available on single site, one need to search and read a lot from different sites.

In this project, all the information about different tourist places is gathered from web data like Wikipedia and shown at single place. Mainly historical events are mapped as per timeline which will be very helpful to interested people. Wikipedia provides lots of information. The information will be fetched and processed to get appropriate details and history.

**1.2 OJECTIVE**

To design and develop a Travel Guide with the help of Wikipedia data using topic modeling and word embedding.

**CHAPTER 2**

**SYSTEM OVERVIEW AND METHODOLOGY**

**2.1 INTRODUCTION**

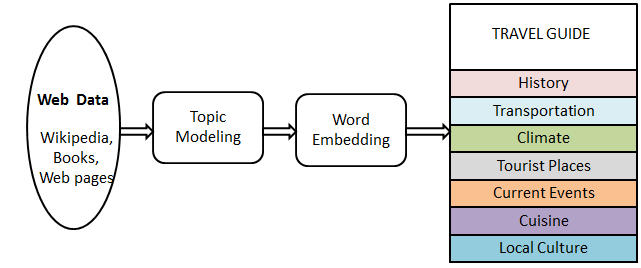


Figure 2.1 System overview

The overall project idea is shown in the figure 2.1. Data is fetched from websites like Wikipedia. Data is converted to plain text from html and stop words are removed and pre-processing is done. To find and observe large bunch of words in cluster of data topic modelling is done. In word embedding, word vector is prepared by using those words. From all these steps, summarized information is gathered and with this information travel guide is prepared which includes the information like history, transportation, climate, tourist places, current events, cuisine and local culture.

**2.2 METHODOLOGY**

Methodology consists of six steps, explained below,

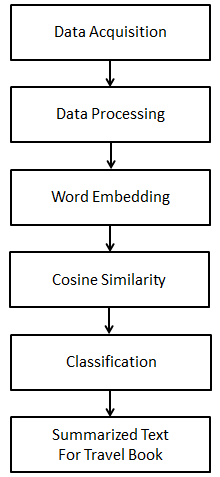


Figure 2.2 Methodology

**2.2.1 Data Acquisition**

Data is fetched from Wikipedia as html data. Section wise parting is done for processing the information. Wikipedia is one of the biggest data collections available for all the people, places and things. It provides detailed information about all. The principal significance of Wikipedia is that it has demonstrated that hundreds of thousands of people with highly varying backgrounds, views, interests, and native languages can work together to produce something of value with only a very loose governing framework. For this reason, Wikipedia data is considered as first source of structured information for travel guide.

**2.2.2 Data Processing**

Data pre-processing is done by removing 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 are removed by storing a list of words that you consider to be stop words. NLTK (Natural Language Toolkit) in python.

**2.2.3 Word Embedding**

Word embedding is type of word representation that allows words with similar meaning to have a similar representation. It is this approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems. Word vector and document vectors are prepared in word embedding. Details of the vectors are given in the next chapter.

**2.2.4 Cosine similarity**

Cosine similarity is calculated to find out the difference between two sentences. If the sentence is same or as it is, angle between these two will be zero otherwise there will be angle difference between sentences. As figure shows, A and B sentences are shown with cosine similarity. By using cosine similarity, similarity sets are formed and after applying threshold, most important similarity sets can be fetched to get summarized data.

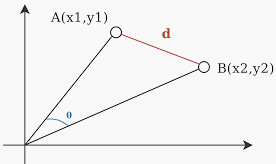


Figure 2.3 Cosine similarity

**2.2.5 Clustering algorithm**

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. K-means clustering is applied on similarity sets to classify the text data into groups. Also, to get timeline wise events, proximity words are found out using word2vec and doc2vec.

**2.2.6 Summarized text**

Travel book is prepared which contains detail information like State/Country, Distance from city, Co-ordinates, Height from sea level, Owner, Controlled by , About structure, Origin of name, Climate, Historical Timeline, Historical events, Restoration, Culture, Travel Modes, Famous Food, Present Events, Similar places, Nearby Places. From above classifications, summarized text can be fetched.

**CHAPTER 3**

**TOPIC MODELLING**

**3.1 INTRODUCTION**

Topic Modelling is the rule-based text mining approach which uses regular expressions or dictionary based keyword searching techniques. It is an unsupervised approach used for finding and observing the bunch of words in large clusters of texts. 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. Topic modelling is an unsupervised machine learning technique scans a set of document, detect word and phrase patterns within them, and automatically cluster word groups and similar expressions that best characterize a set of documents.

Topic modelling can be done in different ways; Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA) etc. In this work, we have used Latent Dirichlet Allocation (LDA)

**3.2 LATENT DIRICHLET ALLOCATION (LDA)**

Latent Dirichlet Allocation (LDA). LDA is a probabilistic topic model and it treats documents as a bag-of-words. Latent Dirichlet Allocation (LDA) model is improved way of PLSA and LSA. This was happened in 1990, so the classic representation theorem lays down that any collection of exchangeable random variables has a representation as a mixture distribution—in general an infinite mixture [3]. latent topic modeling has become very popular as a completely unsupervised technique for topic discovery in large document collections. This model, is an Algorithm for text mining that is based on statistical topic models and it is very widely used. LDA is a generative model that tries to mimic what the writing process is. So it tries to generate a document on the given topic. It can also be applied to other types of data. In a simple way, the basic idea of the process is, each document is modeled as a mixture of topics, and each topic is a discrete probability distribution that defines how likely each word is to appear in a given topic. These topic probabilities provide a concise representation of a document. Here, a "document" is a "bag of words" with no structure beyond the topic and word statistics.

In this work, we have created LDA model to get important words from cluster of text.

**CHAPTER 4**

**WORD EMBEDDING**

**4.1 INTRODUCTION**

Word embedding is vector representations of a particular word. Word Embedding is really all about improving the ability of networks to learn from text data. By representing that data as lower dimensional vectors. These vectors are called Embedding. This technique is used to reduce the dimensionality of text data but these models can also learn some interesting traits about words in a vocabulary. Popular embedding models are as followes: Word2Vec (by Google), GloVe (by Stanford), fastText (by Facebook).

In this work we have used Word2Vec and Doc2vec embedding.

**4.2 WORD2VEC**

Word2Vec is one of the most popular technique to learn word embeddings using shallow neural network. It was developed by [Tomas Mikolov in 2013 at Google](https://arxiv.org/pdf/1310.4546.pdf).

Word2Vec is a method to construct such an embedding. It can be obtained using two methods : Skip Gram and Common Bag Of Words (CBOW)

**4.2.1 CBOW Model:**

This method takes the context of each word as the input and tries to predict the word corresponding to the context. It tries to predict a target word using a single context input word. More specifically, it uses the one hot encoding of the input word and measure the output error compared to one hot encoding of the target word.In the process of predicting the target word, it learn the vector representation of the target word.

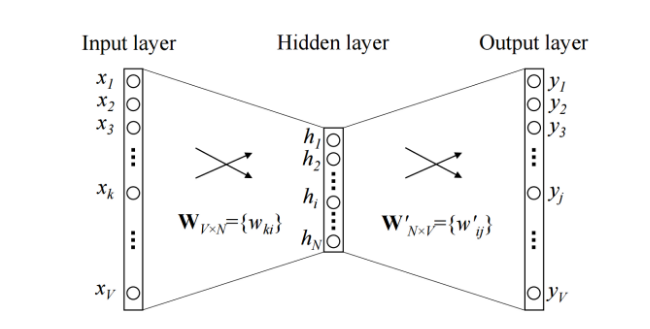


Figure 4.1 CBOW model with one word in context

The input or the context word is a one hot encoded vector of size V. The hidden layer contains N neurons and the output is again a V length vector with the elements being the softmax values. The hidden layer neurons just copy the weighted sum of inputs to the next layer. The only non-linearity is the softmax calculations in the output layer. It can use can use multiple context words to predict the target.

**4.2.1 Skip-Gram model:**

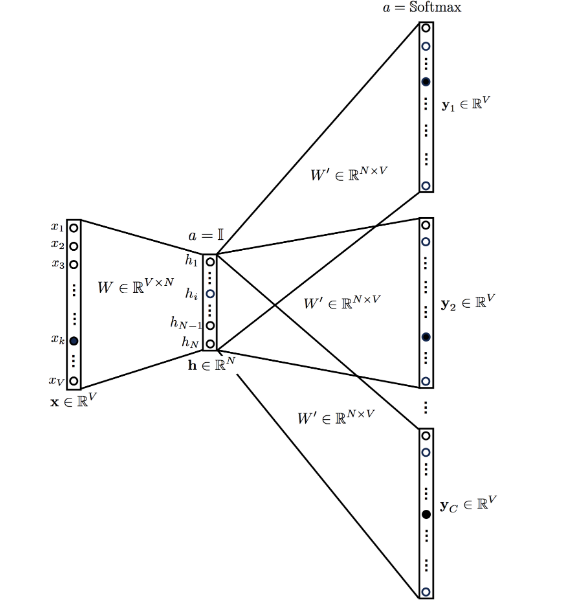


Figure 4.2 Skip gram model with one word in context

In this algorithm, The target word (whose representation we want to generate) to predict the context and in the process, can produce the representations. It is flipped version of CBOW.

Both have their own advantages and disadvantages. According to Mikolov, Skip Gram works well with small amount of data and is found to represent rare words well. On the other hand, CBOW is faster and has better representations for more frequent words.

**4.3 DOC2VEC**

Doc2vec modifies the word2vec algorithm to unsupervised learning of continuous representations for larger blocks of text, such as sentences, paragraphs or entire documents. The goal of doc2vec is to create a numeric representation of a document, regardless of its length. But unlike words, documents do not come in logical structures such as words, so the another method has to be found.

The algorithm then runs through the sentences iterator twice: once to build the vocab, and once to train the model on the input data, learning a vector representation for each word and for each label in the dataset.

**CHAPTER 5**

**CLASSIFICATION**

**5.1 INTRODUCTION**

The classification is important to get the most important information out from the cluster of text. K-means clustering forms clusters of closely related information, from which we can get text data. After fetching date or years from the cluster of data, events happened can be found out using word2vec.

**5.2 K-MEANS CLUSTERING**

K-means algorithm is an iterative algorithm that tries to partition the dataset into *K* pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster’s centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids

It halts creating and optimizing clusters when either:

* The centroids have stabilized — there is no change in their values because the clustering has been successful.
* The defined number of iterations has been achieved.

In this work, similarity sets are clustered, to get most similar sentences in text. Each cluster sentences are summarized data.

**5.3 TIMELINE BASED CLUSTERING**

To calculate event, happened in the year, all dates are fetched from text data. The dates can be fetched using tagging- name entity recognition. By using word2vec, proximity words to these years are fetched and event is formed. These events are used to build final travel book.

**5.4 TAGGING**

Name entity recognition classify named entity in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, percentages, etc. NER is used in many fields in Natural language processing (NLP), and it can help answering many real-world questions.

NER systems have been created that use linguistic grammar-based techniques as well as statistical models such as machine learning. Hand-crafted grammar-based systems typically obtain better precision, but at the cost of lower recall and months of work by experienced computational linguists. Statistical NER systems typically require a large amount of manually annotated training data. Semi-supervised approaches have been suggested to avoid part of the annotation effort.

## 5.4.2 spaCy NER Model

Being a free and an open-source library, spaCy has made advanced Natural Language Processing (NLP) much simpler in Python. spaCy provides an exceptionally efficient statistical system for named entity recognition in python, which can assign labels to groups of tokens which are contiguous. It provides a default model which can recognize a wide range of named or numerical entities, which include company-name, location, organization, product-name, etc to name a few. Apart from these default entities, spaCy enables the addition of arbitrary classes to the entity-recognition model, by training the model to update it with newer trained examples.

**CHAPTER 6**

**ALGORITHM IMPLEMENTAION**

**6.1 INTRODUCTION**

Implementation is done in python Jupyter notebook. All the modules further used to create a travel guide.

**6.2 WORK DONE**

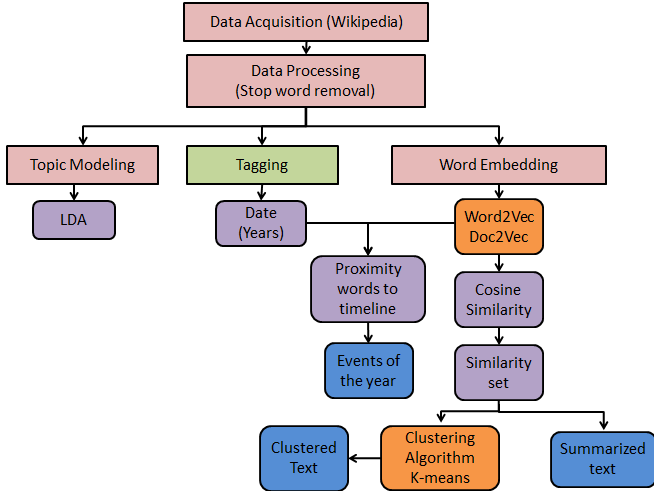


Figure 6.1 Overall work done

1. **LDA** – Topic modelling is done using LDA. These topics will be further used.

**2**. **Events of the year –** They will be further used to create history part of the travel guide

3. **Summarized data** – Data is summarised using similarity sets and further used to create travel book

4. **Clustered data** – Clustered data forms group of similar sentences and further be used to provide some information.**CHAPTER 7**

**RESULTS**

**7.1 INTRODUCTION**

The modules are implemented using Python IDLE and Jupyter notebook. Screenshot for different modules are shown below.

**7.2 SIMILARITY SETS**

From similarity sets, summarized text is generated using Word2vec model. In figure 7.1, total 22 similarity sets are having threshold above 0.7 and after converting vectors to sentences, the text data will be seen which is summarisation of the whole data.

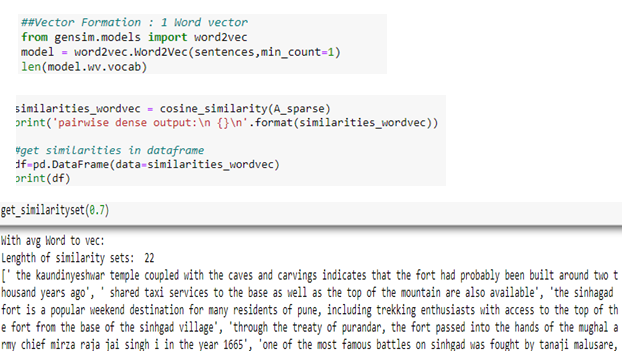


Figure 7.1 Summarized text with similarity sets

**7.3 TOPIC MODELING**

Topic modeling is done using LDA, to find out topics present in the cluster of text. Words are grouped together to form different topic as shown in figure 7.2

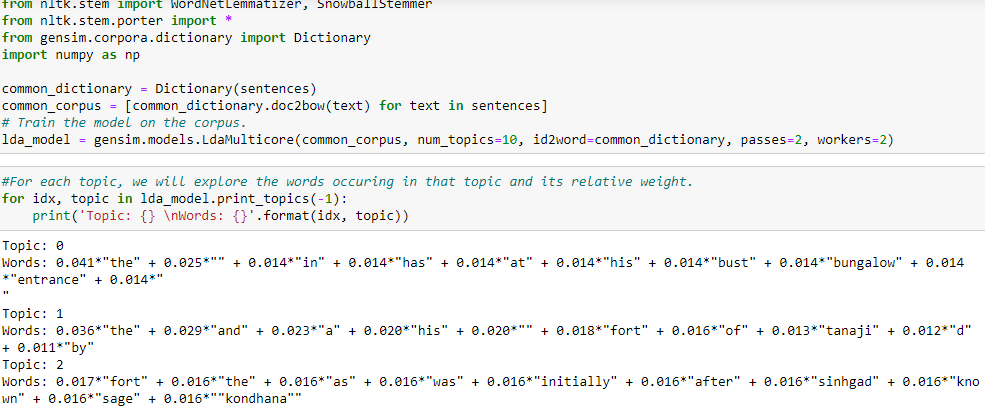


Figure 7.2 Topic modeling – LDA

**7.4 K-MEANS CLUSTERING**

By using k-means clustering, similarity sets are clustered into three different groups. Each group represent different summarized set.

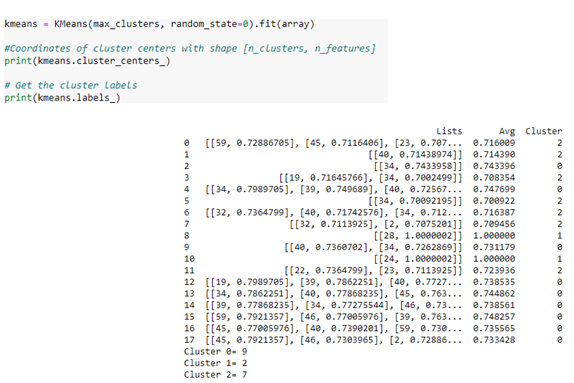


Figure 7.3 K-means clustering

**7.5 TIMELINE WISE CLASSIFICATION**

Timelines i.e. years are fetched from the wiki data and proximity words to that year are found out. From those proximity words, the event must have happed in that timeline can be fetched. Word cloud of proximity words can be formed.



Figure 7.4 Timeline wise classifications

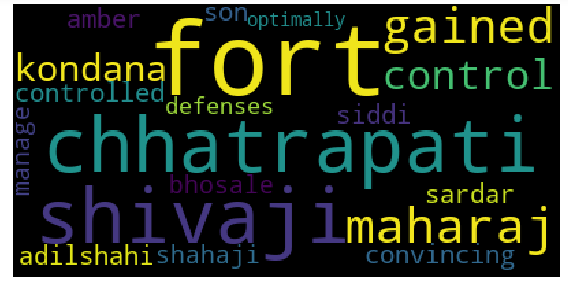


Figure 7.5 Word Cloud for proximity words of 1664

**7.6 TAGGING**

Tagging is also known as Name-Entity Recognition (NER). Every word present in the cluster is classified or named with general terms. For example, if text contains 1780, it will be termed as DATE. It is done bby using Spacy API.



Figure 7.6 Tagging

**CHAPTER 8**

**CONCLUSION**

Data acquisition from Wikipedia, data processing by removing stop words is done. Topics are fetched from that text and word embedding is implemented by creating two models- Word2vec and Doc2vec. These models are used to get cosine similarity and summarized text is formed by applying threshold to cosine similarity. Theses similarity sets are used for K-means clustering to form clusters. To get historical events as per year, timeline wise classification is done using word2vec model. Proximity words to the timeline gives event happened on that timeline.

These modules are coded and further will be used while final development of travel guide.

**REFERENCES**

[1] George Forman, Evan Kirshenbaum “Extremely Fast Text Feature Extraction for Classification and Indexing”, HP Laboratories, HPL-2008

[2] Hong Liang, Xiao Sun, Yunlei Sun and Yuan Gao,” Text feature extraction based on deep learning: a review”, Liang et al. EURASIP Journal on Wireless Communications and Networking, 2017

[3] Blei, D.M., Ng, A.Y., and Jordan, M.I., ―Latent Dirichlet Allocation‖, Journal of Machine Learning Research, 3, 2003, 993-1022.

1. [**https://towardsdatascience.com/**](https://towardsdatascience.com/)
2. [**https://radimrehurek.com/gensim/**](https://radimrehurek.com/gensim/)
3. [**https://github.com/**](https://github.com/)
4. [**https://www.wikipedia.org/**](https://www.wikipedia.org/)
5. [**https://www.machinelearningplus.com/**](https://www.machinelearningplus.com/)