Analysis of dark web pages using GoW and Topic Modelling

Final year project report submitted to

Veermata Jijabai Technological Institute

in fulfilment for the award of the degree of

Bachelor of Technology

in

Computer Engineering

by

Saurabh Raut, Nirmit Joshi, Dhairya Bhuta, Saikumar Nalla

(171070005, 171070010, 171070011, 171070030)

Under the supervision of

Dr. Sunil Bhirud

Department of Computer Engineering

Veermata Jijabai Technological Institute

A.Y. 2020-21

May 24, 2021

DECLARATION OF STUDENT

I declare that the work embodied in the Stage-II of this project titled **Analysis of dark web pages using GoW and Topic Modelling** form my own contribution of work under the guidance of Dr. Sunil Bhirud at the Department of COMputer Engineering, Veermata Jijabai Technological Institute, Mumbai. This report reflects the work done during the academic year 2020-21 as final year project.

Saurabh Raut
Roll No: 171070005

Nirmit Joshi
Roll No: 171070010

Dhairya Bhuta
Roll No: 171070011

Saikumar Nalla
Roll No: 171070030

Date: 24-05-2021

Place: VJTI, Mumbai

Department of Computer Science & Information Technology

VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE, MUMBAI

Certificate

This is to certify that this is a bonafide record of the project presented by the students whose names are given below during 2020-21 in fulfilment of the requirements of the degree of Bachelor of Technology in Computer Science and Engineering.

Roll No	Names of Students
171070005	Saurabh Raut
171070010	Nirmit Joshi
171070011	Dhairya Bhuta
171070030	Saikumar Nalla

Dr. Sunil Bhirud
Project Guide

Dr. M.M. Chandane
Head of CE & IT Department

Date: 24-05-2021

Place: VJTI, Mumbai

Department of Computer Science & Information Technology

VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE, MUMBAI

Approval Sheet

The Project titled, Analysis of dark web using Graph of Words and Topic Modelling, submitted by Saurabh Raut(171070005), Nirmit Joshi(171070010, Dhairya Bhuta(171070011), Saikumar Nalla(171070030), is found to be satisfactory and is approved for the Degree Of Bachelor Of Technology (Computer Engineering))

Examiner 1
Dr. Sunil Bhirud

Examiner 2
Dr. Satish Devne

Date: 24-05-2021

Place: VJTI, Mumbai

Abstract

Name of the student: **Saurabh Raut, Nirmit Joshi, Dhairya Bhuta, Saikumar Nalla** Roll No: **171070005, 171070010, 171070011, 171070030**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Computer Engineering**

Thesis title: **Analysis of dark web pages using GoW and Topic Modelling**

Thesis supervisor: **Dr. Sunil Bhirud**

Month and year of thesis submission: **May 24, 2021**

Text Analysis is the process of extracting meaningful information from unstructured text. The aim of Text Analysis is to create structured data out of free text content. We are applying Text Analysis on Dark webpages. The Dark Net is a dark side of the internet that became a perfect hosting ground for criminal activities and services, including significant drug marketplaces and likewise. The nature of the Dark Net makes it indifferent to searching through the indexed mechanism. Dark Net requires special tools to access it. We have used Tor to access the Dark Net. It helps us to remain anonymous while surfing the Internet.

Diverse approaches are suggested and attempted to gain useful insights from the Dark Net. We are exploring and analysing the dark web using graph of words and graph embedding. Dark web contains data such that a threat can be anticipated by proper analysis. Data is scraped in the form of web pages using a Deep Web crawler. We build a Graph-of-Words (GoW) model and analyse it for gaining insights from the data and also generate graph embedding from GoW to observe further results

Acknowledgements

We would like to thank all those people whose support and cooperation has been an invaluable asset during the Stage I of this Project. I would also like to thank our Guide Dr. S. G. Bhirud for guiding us throughout this project and giving it the present shape.

We would like to extend our gratitude to Mrs. Ashwini Dalvi for supervising our project and guiding us in every step right from literature survey to implementation. It would have been impossible to implement this project without her support, valuable suggestions, criticism, encouragement and guidance throughout the time frame amidst this lockdown.

We are highly indebted to our Head Of Department, Dr. M. M. Chandane for providing us with this opportunity to implement this project as out Final Year B.Tech Project.

Lastly, we are also grateful for the Department of Computer Engineering and Information Technology, VJTI Mumbai for providing us with the access to several research papers and journals that have helped us do a thorough literature review for this project and increase our knowledge manifold.

Contents

List of Figures

Abbreviations

TOR	**T**he **O**nion **R**outer
TF-IDF	**T**erm **F**requency **I**nverse **D**ocument **F**requency
BOW	**B**ag **O**f **W**ords
NLP	**N**atural **L**anguage **P**rocessing
IP	**I**nternet **P**rotocol
LDA	**L**atent **D**irchlet **A**ssociation
HMM	**H**idden **M**arkov **M**odel
FBI	**F**ederal **B**ureau of **I**nvestigation
ARIMA	**A**uto **R**regressive **I**ntegrated **M**oving **A**verage
GoW	**G**raph **O**f **W**ords
DGEF	**D**iachronic **G**raph **E**mbedding **F**ramework

Chapter 1

Introduction

1.1 Problem Definition

Over the past few years, there has been an exponential growth in the data available on deep web. We propose a two-fold approach to gain insights from the deep web data. In particular, we first analyze the data by performing key-words extraction from each document. We then evaluate the results produced by this method by comparing it with the results generated by the Latent Dirichlet allocation (LDA) on the same data. This will have impact in various domains including cybercrime prevention, customer care service, fraud detection through claims investigation, social media data analysis and business intelligence.

1.2 Motivation

To most users, the web is what they experience through their web browser every day, but there are a lot of expansive services that function in the background and the “web” is just one part of it. There are numerous layers behind that web browser that the average user mights encounter tangentially or never. The web is divided as Surface Web, the Dark Web, and the Deep Web.

The Surface Web is what users access in their regular day-to-day activity. It is available to the general population using standard search engines and can be accessed using standard web browsers that do not require any special configuration, such as Mozilla Firefox, Microsoft's Internet Explorer or Edge, and Google Chrome.

The Deep Web is the portion of the web that is not indexed or search able by ordinary search engines. Users must log in or have the specific URL or IP address to find and access a particular website or service. Some pages are part of the Deep Web because they do not use common top-level domains (TLDs), such as .com, .gov, and .edu, so they are not indexed by search engines, while others explicitly block search engines from identifying them. Many Deep Web sites are data and content stored in databases that support services we use every day, such as social media or banking websites. The information stored in these pages updates frequently and is presented differently based on a user's permissions.

The Dark Web which is the part of the internet this project focuses on is a less accessible subset of the Deep Web that relies on connections made between trusted peers and requires specialized software, tools, or equipment to access a particular website or service. The darknets which constitute the dark web include small, friend-to-friend peer-to-peer networks, as well as large, popular networks such as Tor, Freenet, I2P, and Riffle operated by public organizations and individuals.

The dark web is the World Wide Web content that exists on darknets: overlay networks that use the Internet but require specific software, configurations, or authorization to access. Through the dark web, private computer networks can communicate and conduct business anonymously without divulging identifying information, such as a user's location. The dark web forms a small part of the deep web, the part of the Web not indexed by web search engines, although sometimes the term deep web is mistakenly used to refer specifically to the dark web.

1.3 Scope of the Thesis

Due to the limitations enforced by the nature of the dataset, we limit ourselves to only a small family of machine learning algorithms which are unsupervised, can analyze the static data, and are also computationally efficient.

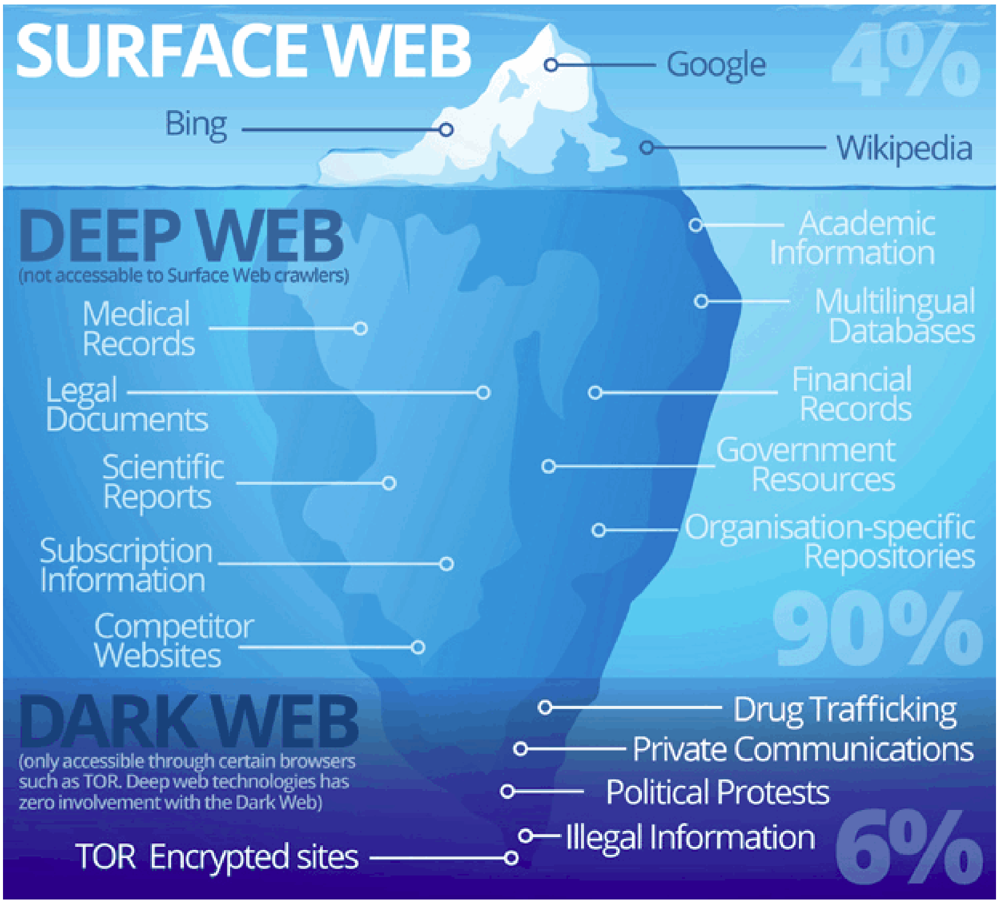


Figure 1.1: D2Web Representation by Faisal Khan (2018)

- Scaling to new languages requires new embedding matrices and does not allow for parameter sharing, meaning cross-lingual use of the same model is not an option.
- The analysis is limited to textual data.
- Our text analysis methods have problem recognizing things like sarcasm and irony, negations, jokes, and exaggerations and failing to recognize these can skew the results.

1.4 Organization of Thesis

Chapter one introduces the problem statement and motivation of the project idea.

Chapter two is about a literature survey in which works related to different crawlers, crawling algorithms and dark web data analysis.

Chapter three focuses on the methods of representing the dark web data.

Chapter four gives a deep understanding of our methods using the mathematical model.

Chapter five talks about the keyword extraction approach of text analysis.

Chapter six is about the topic modelling approach of text analysis.

Evaluation of our methods is discussed in chapter seven.

The conclusion of this project has been summarized in chapter eight.

Chapter 2

Literature Review

This chapter gives an overview of the papers that were read as part of the literature survey. It highlights the different papers that have tried to analyse dark webpages and forums using several analysis techniques. Next, papers related to specifically analysing drug related marketplaces and forums are covered. Later it explains how crawling mechanisms are being improvised. Lastly, papers providing ways to analyse cyber and hacker related forums on the dark web are discussed.

2.1 Literature Review

2.1.1 General Analysis of the Dark Web Pages and Forums

Faizan and Khan (2019) tried exploring and analyzing the dark web by collecting the dataset using a custom made Python crawler and the data was classified into 31 categories. This helped in analysing the data and explore the content available on the dark web.

The dark web basically consists of hidden services and these services have their IP addresses hidden from the outside world. This is one of the few research papers that also focuses on both English as well as Non-English content.It was discovered that the second most used language on the dark web was Russian after English.

It was found that services related to Bitcoin formed the major chunk of services in the dataset. These services comprise of Bitcoin wallets, Bitcoin doubling, etc. The different categories other than Bitcoin related services include those related to email, anonymity and privacy tools. All these sum up to around 10 % of all hidden services. The category Marketplace contains those services such as black markets, selling illict goods, PayPal accounts that have been compromised and hacked. 165 hidden services had made adult content accessible.

There was great similarity observed in the types of services offered on both English as well as non-English dark web services.

Tavabi et al. (2019) attempt to distinguish activity on the d2web (short for deep and dark web). Messages that were posted to 80 d2web forums spanning a period greater than a year were used to identify topics of discussion using LDA and the evaluation of topics on the forums was modelled using a Hidden Markov Model(HMM)

The LDA was implemented using Gensim to build a model with 100 topics. The topics learned by LDA were explored by focusing at the most important words. Each forum is represented as a time series of topic vectors learned by the model for analysing the changes in the topics on dark web forums. The topic vectors of the weekly posts are averaged to generate the forum's vector and this time series information of weekly topic vectors were used to understand HMM states and compute cross-entropy. HMM deals with the discrete states of the forum dynamics and is consequently less sensitive to noisy data , however this also causes it to miss small meaningful changes. To compensate for the same, we compute the cross entropy of forums to confirm the output obtained using the non-parametric HMM.

Results showed that large and active forums have extended discussions on different topics however their average topic distribution is usually uniform over time.

Sentiment analysis tries to identify and analyze emotions. Affect analysis is the category of sentiment analysis dealing with emotions/affects. It can be used for measuring the presence of hate and violence across extremist groups. In the paper by Abbasi and Chen (2007), they constructed an affect lexicon using a probabilistic disambiguation technique to measure the usage of violence.

The two major components of their system design are:

1) Affect Lexicon Creation : The affect term lists are made by assessing messages using local speakers recognizable with the web lingo for each locale.
2) Affect Analysis Technique : The extraction of affect features from the message text is done by the affect parser. After extracting features from the text, the parser is designed for handling noisy text.

The research findings suggest that the Middle Eastern and U.S. forums have similar levels of hate related emotions. On the other hand, the Middle Eastern test bed gatherings have twice as much barbarity concentrated as the U.S. forums. Also, in the Middle Eastern forums there is a strong correlation between racism and violence.

After developing a system for dark web analysis, their techniques perform affect intensity and relationship analysis for deeper insights into discourse affects. They believe that this approach would also be appropriate for further analysis of extremist group web discourse.

2.1.2 Darknet drug markets and forums

In the last years, governmental bodies have been trying to fight against dark web marketplaces. After FBI and Europol closed ”The Silk Road” in 2013, only in some time, new successors were established.The paper Baravalle et al. (2016) presents a research carried out to know in depth about the products and services sold on one of the larger marketplaces for drugs, fake ids and weapons i.e. Agora. As it turns out there is an obvious predomination by the drug market which accounts for 80% of the total items on sale.

Given that Agora is hosted on TOR, navigation on Agora is anonymous. It was also found that Agora uses a several techniques for authentication in addition to techniques for discouraging web scraping.Also the total market for drugs was huge in all possible ways such as total items on sale, total sellers offering drug related services. From the analysis , it was found that the highest number of sellers are located within the USA, UK while the top countries considering the size of market are Germany, USA respectively. The other two areas that this paper focused was on the sale of counterfeit documents and organized crime on Agora.

Haasio et al. (2020) tried to analyse the lives of drug users using a website which is based in Finland. In order to get insights into the socio-economic conditions of the users, 9300 posts from the site were analysed. This analysis involved using usernames, forum posts to find representation of a user's way of life and to find their association with drugs

A user's identity can be determined based on the language they choose for their name. In the Sipulitori sample, usernames were mainly in Finnish language, these account for 57% of the total usernames. This is followed by English which account for 20% and then 5% usernames were in other langauges. For the remaining 17% usernames, the language could not be identified. This suggests that users would not opt for a global identity but prefer an identity that indicates they belong to local community since there are notions that people belonging to certain ethnicities are not reliable for doing business with. Names based on drugs are relatively low. Infact usernames spanned a wide area of topics such as personal names,made-up words, names related to places, and names of fictional characters. This suggests that most of the users are hobbyists and not people involved in some serious criminal activities.

From the analysis of several messages it could be figured out that the users' habitus was a learned way of keeping one's lifestyle in check without breaking social ethics.

Broséus et al. (2016) gave an overview of the Canadian illegal medicate advertise utilizing information collected on eight cryptomarkets, in this way giving an understanding into the structure and organization of dispersion systems existing online. It depicts how the investigation of information accessible online may inspire information on criminal activities.

2.1.3 Deep and Dark web Crawling

There has been increasing interest amongst researchers for techniques to efficiently find deep webpages as the deep web continues to grow day by day.The two main issues to handle are coverage and efficiency given the dynamic nature of the deep web and huge volumes of resources.A two stage framework namely Smart Crawler was proposed by Zhao et al. (2016) for efficiently harvesting deep web interfaces.

The proposed crawler is divided into two stages: site locating and in-site exploring. The site locating part deals with achieving wide coverage of deep websites , on the other hand, the in-site exploring part handles efficiently performing searches within a site using web forms

The site locating technique uses a reverse searching technique and incremental two-level site prioritizing technique to obtain relevant websites and thus attaining more data sources.

In the in-site exploring stage, so as to have an unbiased link prioritization mechanism a link tree is developed which helps in removing bias towards web pages in popular directories. An adaptive learning algorithm performs online feature selection and these features are then used to construct link rankers. In the site locating stage, highly relevant sites are prioritized and the crawling is focused on a topic using the contents of the root page of sites, achieving more accurate results.

The experimental results on a particular set of areas show the effectiveness of the proposed two-stage crawler, as it successfully achieves higher harvest rates than other different crawlers.

2.1.4 Using Graph of Words and Graph Embeddings to analyse forums on the dark web

Often exploits and techniques are discussed on hacking forums. In a paper by Deb et al. (2018), they proposed an approach to predict cyber events using sentiment analysis on hacker forum posts to better understand hacker behavior.They applied the most commonly used ARIMA model for events related to forecasting. ARIMA stands for autoregressive integrated moving average. The main idea is that the number of current events is a function of past counts and forecast errors. Formally, ARIMA(p,d,q) defines an autoregressive model with p autoregressive lags, d difference operations, and q moving average lags. This methodology predicts potentially dangerous cyber events by analysing the actor's behaviour using sentiment analysis on the hacker forums.

A further improvement in analysing hacker forums was provided by Samtani et al. (2020) who proposed D-GEF (Diachronic Graph Embedding Framework) which operates on a directed GoW representation of hacker forum text. Then state of the art graph embedding algorithms are applied on the GoW to generate low dimensional word embeddings.The last step involves semantic displacement measures applied to identify how word embeddings evolve over time.

The D-GEF comprises of 3 components:
1) Text Graphs : Graph of Words (GoW) Representation
2) Unsupervised Graph Embedding Methods
3) Diachronic Word Embeddings

The Graph of Words representation of the forum text enables preserving the semantic relationships amongst the words and also use several other graph related measures to find deeper insights about the words used in the context of hacker forums. Unsupervised Graph Embedding methods help in creating low dimensional embedding by preserving node proximities at several levels. Although graph embeddings are promising, they fail to capture evolution / shifts in the words used on the forum and the context in which they are used. This is overcome by the third component of the D-GEF.

The detailed literature survey helped in coming up with the idea of using the GoW representation model along with the graph embeddings described above for analysing drug websites on the dark web and gain insights of the structure and content of these web pages.

Chapter 3

Dataset And Representations

3.1 Data Collection

The deepweb is that portion of the internet which comprises of webpages that are not indexed by the surface web and require special software/authorization to acscess them. Moving another layer deeper, we have the darknet which is a subset of the deepweb. The darkweb is known for the dynamic nature of content on its webpages. This basically refers to how one link leads to some content at one time instance and to some other content or no content at all at another time instance. These websites that comprise the hidden network can be accessed using certain software, applications such as TOR, Freenet, etc.

The darkweb dataset was made available using a custom darkweb crawler. This crawler crawled the dark webpages with the help of certain seed links that were provided to the crawler. These seed links form the starting point for the crawler and further links are explored as they are encountered by the crawler. In this way, the data is collected by crawling the darknet. It is important to note that the webpages collected belong to only one particular time instance.

3.2 Properties Of The Dataset

1. **Large:**
   The dataset obtained comprises of 50000 dark webpages which accounted for 12.2 GB data.

   - Due to such a large dataset, a lot of diversity was found in the languages of the webpages. The different langauages that were observed in these webpages included English, Russian, Chinese, French, Spanish, etc.
   - The structure of the dataset included .html, .php and .htm files and all of them having HTML code of the webpages crawled. The large size of the dataset can also be attributed due to the raw nature of the data i.e. HTML code of all the webpages. For example if a particular webpage has content equivalent to 100 words, but the corresponding HTML code along with the CSS styling and Javascript can increase the file size by a significant amount.

   The above two features of our dataset force us to perform the primary data cleaning steps namely:- removing non-english webpages and removing the HTML tags, Javascript from the files.

2. **Unlabelled:**
   The dataset in hand is completely unlabelled. No manual labelling is performed and there is no availability of any information using which, the topic of a partiicular webpage can be inferred. This also restricts the data analysis to be performed using only unsupervised machine learning algorithms. This is because of the fact that supervised machine learning algorithms require some form of labelling attached with the data, using which the supervised algorithms learn.

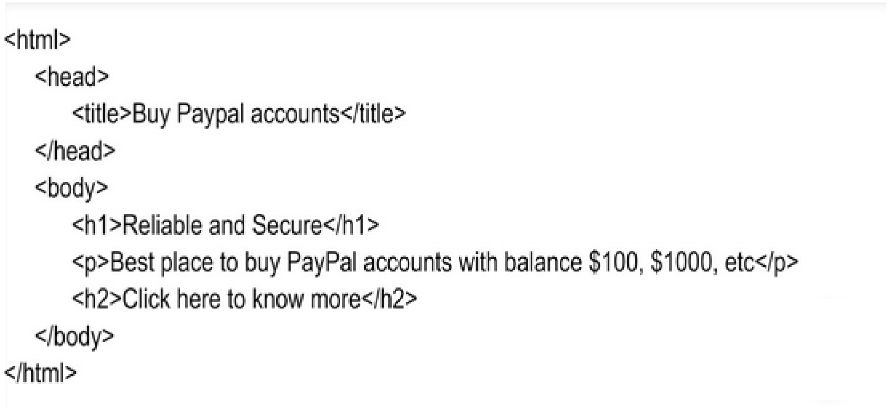
3. **Static:**
   The dark webpages are dynamic in nature and are constantly updated, but as previously mentioned, the dataset obtained comprises of webpages belonging to only a particular time. Thus the static nature of the dataset removes the possibility of any temporal analysis of the darknet webpages that could have been performed.

3.3 Data Preprocessing

The data preprocessing part involved the following three steps:

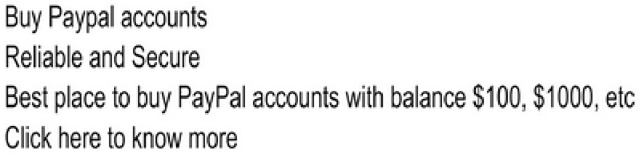
1. Removing HTML tags
2. Removing Non-English webpages
3. Removing Facebook related dark webpages

- **Removing HTML tags:**
  As already mentioned, the data collected comprises of HTML webpages. Thus to perform meaningful analysis, it was important to first extract the main body of the webpages by removing HTML tags and JavaScript code from the documents.
  This was achieved with the help of **Beautiful Soup** library. This library helps in pulling data from HTML, XML and other markup languages. For our use case, Python's html.parser was used.
  Figure 3.1 refers to a sample document from the dataset and Figure 3.2 refers to the same document after cleaning it i.e. after running the python program to remove the HTML tags.



```
<html>
   <head>
      <title>Buy Paypal accounts</title>
   </head>
   <body>
      <h1>Reliable and Secure</h1>
      <p>Best place to buy PayPal accounts with balance $100, $1000, etc</p>
      <h2>Click here to know more</h2>
   </body>
</html>
```

FIGURE 3.1: Document before cleaning



Buy Paypal accounts
Reliable and Secure
Best place to buy PayPal accounts with balance $100, $1000, etc
Click here to know more

Figure 3.2: Document after cleaning

- **Removing Non-English webpages:**
  As the scope of our analysis was confined to only English dark webpages, the next data preprocessing step was to remove all the webpages that are not in English language. This was achieved with the help of the **langdetect** library in python.
  This library provides support for 55 different languages. Depending on the computations performed by the **detect** function available with the library, it returns the ISO 639-1 code for the language the text could belong to. For example if the function finds that the input text is English content, then it returns 'en'. After the Non-English webpages were removed, we were left with approximately 22,000 webpages.

- **Removing Facebook related dark webpages:** On further analysing the reduced set of 22,000 documents, it was found that many of these were Facebook webpages, which mostly comprised of posts having personal opinions, status updates, description of vacations. Thus these wouldn't contribute much to our analysis of dark web content , on the contrary these could have potentially distorted our dataset and thus we removed them.
  This step was performed manually as we had to simply remove the folders having "facebook" as the substring in their names. There were only around 89 such folders of the total 533 folders and thus could be easily removed manually.This concludes the data preprocessing steps and finally we are left with a dataset comprising of 2660 webpages.

3.4 Different data representations

In machine learning and deep learning, the algorithms cannot process strings in their raw form. In order to extract valuable information from the document vocabulary, different representation models have been developed. On a very broad level, there are two types of embeddings namely word embeddings and graph embeddings. Word embeddings are further divided into two types: Frequency-based and Prediction-based. Following is the description of some popular word embeddings and then a general overview of the Graph of Words representation and how graph embeddings serve as a richer representation in preserving the semantics and relationships of the document.

3.4.1 Bag of Words

The Bag of Words is a simple representation wherein the text such as a sentence or a document is represented as a multiset of its words. In this model, the grammar and order of words are not given any consideration. This model considers only the multiplicity of the words in the text. Despite its simplicity, it has seen great success in language modeling and document classification problems.

3.4.2 TF-IDF

TF-IDF short for term frequency-inverse document frequency is a measure of how relevant a word is to a document in a collection of documents. The TF-IDF can be seen as providing more information about the relevance of a word in a document rather than just giving the count of the total number of times the word appears in the document as in the BOW representation.

3.4.3 Co-occurrence matrix

Bag of Words and TF-IDF do not capture the position in semantics, co-occurrences in the document. In the Co-occurrence matrix representation we store the co-occurrence of words and count the number of times each word appears inside a window of a particular size around the word of interest thus preserving semantic relationships between words.

3.4.4 Word2Vec

Word2Vec uses a neural network to learn word associations from a large corpus of documents. It results in each word represented as a vector such that a simple mathematical function represents the level of semantic similarity between the words represented by those vectors.

3.4.5 Graph of Words

Text Graphs are gaining significant attention within the NLP community because they are able to capture and reveal relationships, patterns, and regularities within a corpus not captured using standard representations (e.g. bag of words) and language models (e.g. TF-IDF).

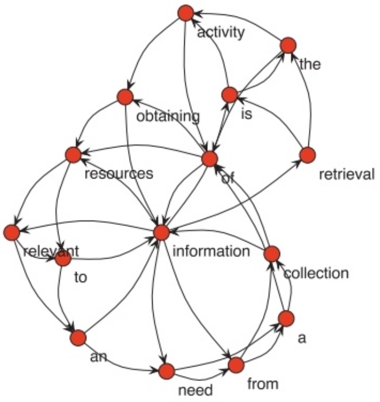


Figure 3.3: Graph of Words: an Example by Rousseau and Vazirgiannis (2013)

The Graph of Words (i.e. word co-occurrence network) is one of the most prevalent text graph formulations in which nodes are words and edges indicate whether two words appear in a specified text unit. If the edges are weighted then that would represent how frequently the words co-occur. This representation enables researchers to calculate a suite of network and node level statistics such as network density, clustering coefficient, etc.

The Graph of Words helps in

- preserving structural and semantic information
- considering term dependence
- determining co-occurences of terms

Given a corpus of words C, the Graph of Words (GoW) is formally represented as $G = (V, E)$ where V is the set of nodes, $\{v_1, v_2, v_3, \ldots v_n\}$ of all words in the corpus, and E is the set of edges, $\{e_1, e_2, e_3, \ldots e_n\}$. The edges between the nodes can indicate different relationships such as similarity, co-occurence , etc depending on the problem being addressed.

Chapter 4

Mathematical Model

In this chapter, we present the mathematical model for our project. We first start the discussion by mentioning some key properties of our dataset in Section 4.1. We then carry the discussion, in Section 4.2, by describing the proposed methodology, which is influenced by the nature of datset. Finally, in Section 4.3, we discuss main conclusions that we want derive from the chapter.

4.1 Dataset properties and their implications

Please recall from Chapter 3 that the dataset, after pre-processing, contains 2660 webpages having plain English text. The dataset we have at our hand affects our choice of models. Thus, it is important to establish how the properties of the dataset mentioned in Chapter 3 affect the methods deployed by us.

- **Large:**

  One of the key properties of our dataset is that each document contains a myriad number of words.

  Effect: The large volume of the dataset forces us to use only those methods which are computationally efficient. Any algorithm having a higher degree polynomial runtime (in terms of the input size) is computationally intractable. However, we will also propose some alternative approaches that one may adopt in the future when computational power may increase.

- **Unlabelled:**

  The dataset in hand is completely unlabelled that is we don't have any labels or information about a document that may help us to infer the content in the document.

  Effect: An important conclusion to derive here is that this property restricts us to a small family of machine learning methods, which is unsupervised learning. The absence of labels makes it impossible to provide any supervision to our methods. Thus, we cannot analyze this data using supervised methods such as Naive Bayes, SVM, and logistic regression, etc. as proposed by Khare et al. (2020).

- **Static:**

  The dataset has states of the webpages at a fixed time instance. However, the information on a darknet webpage is generally evolving over time, which provides some key insights about the nature of that webpage.

  Effect: The static nature of the dataset removes the possibility of any temporal analysis of the darknet webpages that could have been performed. For example, the methods like Samtani et al. (2020) that uses Diachronic framework are not suitable for our dataset.

4.2 Methodologies

The methodology we deploy is twofold. In particular, we first analyze the data by performing key-words extraction from each document. We then evaluate the results produced by this method by comparing it with the results generated by the Latent Dirichlet allocation (LDA) on the same data. In the former (key-words extraction), we use the Graph of Words (GoW) representation of each document, whereas, the LDA model uses the Bag of Words (BoW) representation of the documents.

Due to the limitations enforced by the nature of the dataset, we limit ourselves to only a small family of machine learning algorithms which are unsupervised, can analyze the static data, and are also computationally efficient. We want to emphasize that both of the methods are unsupervised and do not require the states of the webpages at multiple time spells. Below we discuss the key-words extraction method and the LDA model in details and view them from the perspective of computational aspects.

4.2.1 Key-words extraction:

The main intuition behind this method is: central nodes make good keywords that is nodes with high centrality in the GoW of a document usually corresponds to the keywords for the document, which are well-understood by humans. We will formalize this intuition further in Chapter 5. But for now, let's focus on an important question: how do we compare the centrality of different nodes in a graph?

In the graph theory, there are various graph centrality measures such as closeness, betweenness, clustering coefficients, and eigenvector centrality, etc. Each one captures some specific aspects of the graph. However, the algorithmic implementations of calculating any of these measures require more than linear time (in terms of the input size). This may be possible for standard graphs. But the GoWs of the documents are large and thus it is not possible to calculate these measures for all nodes in the GoW of a document. An important remark is that, in the future, when the computation power may increase, potentially due to the arrival of quantum technologies, one may use these measures to get better results that find more accurate keywords for a webpage. But current computational setups and the massive GoWs hinder the possibility to use these measures effectively for the analysis.

Due to the above limitation, we use the **graph degeneracy** method which is efficient. Roughly speaking, the method decomposes a graph into its cores such that it preserves a community of words that are good candidates to be the keywords for the document. After the core decomposition, we are left with a smaller subgraph with only a constant number of nodes. We then compute the closeness centrality measure of only a constant number of nodes in the residual subgraph. Finally, we pick the set of best four possible words as the potential keywords for the document. But a naive implementation of even this algorithm is computationally inefficient. We use an improvised implementation that use a binsort algorithm to achieve $O(n+m)$ runtime, i.e. a linear time complexity in terms of the input size.

4.2.2 Topic Modeling using the Latent Dirchlet Allocation (LDA)

Probabilistic machine learning is a standard approach to topic modeling. In probabilistic machine learning, the task of predicting a function of data is performed by answering an inference question. The main assumption used by probabilistic machine learning methods is that the data is generated by a stochastic process with some hidden and observed random variables. The stochastic process is formally called as the generative process for that model. The generative process is a collection of hidden and observed random variables and their joint distribution. The generative process enables us to factorize the joint.

LDA is one of the baseline models for topic modeling from in the seminal work of Blei et al. (2003). A topic model is a generative model of documents, where each document is a collection of observed words. The main assumption of a topic model is that each document comes from a mixture of categoricals. These categoricals are produced by Dirchilet processes with some hidden parameters. The goal is to find the best estimates of these parameters that best justify the observed documents. In the language of Bayesian machine learning, this is called as calculating the posterior distribution of the hidden random variables from the observed random variables. Let X_H and X_O respectively denote the set of hidden and observed random variables. Then, the entire above discussion can be summarize by the Baye's theorem as follows:

$$P(X_H|X_O) = \frac{P(X_H, X_O)}{P(X_O)}$$

In words, the posterior distribution of the hidden random variables given the observed r.v.s is the ratio of the joint (of hidden and observed r.v.s) to the marginal (of the observed r.v.s).

Calculating the posterior is an NP-hard problem for many probabilistic models. But there are many methods developed such as Gibb's sampling and variational inference (Blei et al. (2017)) to approximate the posterior. We use Gensim, a fast and memory efficient implementation, for the LDA to produce the results.

One of the main advantages of using a probabilistic ML method is that it not only gives the predictions but also tells how confident we are about our predictions. In other words, it is not only about ‘What do we know?’ but also about ‘How well do we know what we know?’. If the model fits well then the posterior distributions are observed to be highly concentrated around their estimates (means). We see that the LDA model fits well on the dark-web data and is able to produce meaningful/interpretable topic assignments to each webpage. We will discuss this in more details in Chapter 6.

4.3 Trade-off between the GoW and BoW representations

As we described, the keywords extraction uses the GoW of the documents, whereas the LDA uses the BoW representation. Intuitively, the GoW representation seems to be more powerful than the BoW representation. This is because every information that can also be retrieved from the BoW representation of a document can be retrieved from a GoW of the same document. We formalize this notion in the following theorem and give an information theoretic proof for the same.

Theorem 4.1. *For a document D, there exists a Graph-of-Words representation $G(D)$, that can entirely generate the Bag of Words $B(D)$ by only looking at $G(D)$, without needing the actual document D.*

Proof. Consider $G(D)$ that has a node for each word in a document. The edges can be arbitrarily assigned with some meaning. The following procedure constructs the Bag-of-Words representation $B(D)$ from $G(D)$. Initialize $B(D)$ to be a zero vector with dimension of the size of the vocabulary. For each node in the graph $G(D)$, increase the frequency by incrementing the entry associated to the word by one. Repeating this procedure for all nodes in $G(D)$ gives us the correct $B(D)$; this follows from the definition of the Bag-of-Words. As such we conclude the proof. □

A natural question that may arise here is that if the LDA uses BoW representation, then why did we choose it to compare it with the the results generated by the keywords extraction method that enjoys richer representation power of GoW. A probabilistic ML method is must to do the self-evaluation of the model. But why didn't we choose a probabilistic method that uses the GoW? Why didn't we choose a method that assumes that all the GoWs are sampled i.i.d. from some generative process with some hidden parameters over the space of random graphs?

The answer to this question lies at the computational limitations which are inherent to our data. It is very hard to model randomness over graphs by only using a few number of hidden parameters. Estimating these parameters and finding their posterior distribution can be intractable. Thus, we use the BoW, compromising on the representation power but gaining the advantage by achieving computational efficiency as discussed in Section 4.2.2. And this where the trade-off between the representation powers and the efficiency of GoW and BoW lies. (Add the diagram)

We would again like to emphasize that one may use the alternative proposed methods in the future if the computational power is not an issue. In fact, these methods seemingly may give us better results in practice. With this remark, we conclude this chapter.

Chapter 5

Keyword Extraction Using Graph Of Words

In this chapter, we present a methodology and its efficient implementation that extracts a set of key-words for each document in a corpus.

5.1 Preliminaries

We first mention some basic definitions in graph theory that are necessary, in the context of unweighted, undirexted graphs.

5.1.1 Basic Definitions in Graph Theory

A graph $G = (V, E)$ is basically a collection of vertices and edges between them. Formally, V its set of vertices (also known as nodes) and E is its set of edges. Edges may be directed, and/or weighted. We denote by n the number of vertices ($n = |V|$) and m the number of edges ($m = |E|$). We will only be dealing with unweighted and undirected graphs. In the context of Graph-of-Words, ————————————-.

Definition 5.1. Connectivity: A graph is connected if there are paths between every pair of vertices, otherwise it is said to be disconnected.

Definition 5.2. Shortest Path: The shortest path between two vertices u and v is the path of minimum length. It is denoted by $d(u, v)$.

Definition 5.3. Degree: The degree $deg(v)$ of a node v is defined as the sum of the weights of its incoming edges.

5.1.2 Centrality Measures

The main intuition behind the methodology is that the "central nodes" in a GoW representation of a documents make good key-words. Thus, it is very important to measure a *centrality* of each node with respect to certain centrality measures. Thus, we formally define some centrality measures and describe what they mean intuitively.

- **Closeness:** The closeness $C_C(v)$ of a node v is defined as the inverse of the total distances from every other node to v in the graph.

$$C_C(v) = \frac{1}{\sum_{u \neq v} d(u, v)}$$

- **Betweenness:** The betweenness $C_B(v)$ of a node v is defined as the fraction of shortest paths from all vertices (except v) to all others (except v) that pass through v.

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

5.2 Mathematical Model

In this section, we describe the model we use to extract the keywords from each document in the corpus of our dataset. The model first decomposes the graph (GoW of a document) into a smaller subgraph that preserves an important set of words. Then, we reduce our search of key-words to only the residual sub-graph. Formally speaking, we do a $k-$core decomposition of the graph for a suitable value of k. We define what is a $k-$core of a graph in the following:

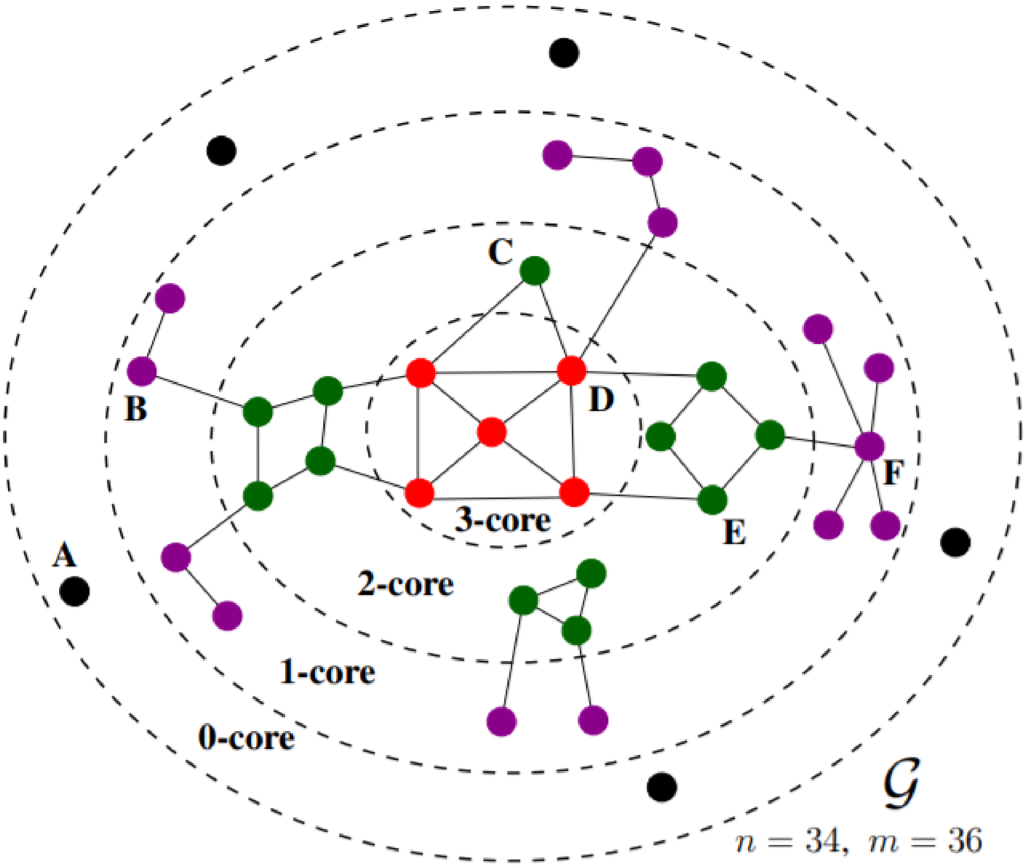


FIGURE 5.1: k-core of a graph for $k = 0, 1, 2, 3$

Definition 5.4. A subgraph $H = (V_H, E_H)$, induced by the subset of vertices $V_H \subseteq V$, is called a k-core or a core of order k if and only if $\forall\ v \in V_H$, $deg_H(v) \geq k$ and H is the maximal subgraph with this property. I.e. it cannot be augmented without losing this property.

The following figure shows the k- decomposition of a graph for 3 different values of k. We first do a core decomposition of a GoW for the smallest value of k such that we have fewer than 30 nodes in its $k-$core. After this step, we find the best four words that have the highest **closeness** centrality measure. Note that any other centrality measure such as betweenness could also be used.

The approach of first decomposing a graph into its k-core and then finding the keywords in the smaller subgraph is termed as graph degeneracy. The main purpose here is to achieve computational efficiecy to restrict our search of key-words to only a small community of words which are closely related.

5.3 Algorithm for Key-words Extraction

Below is the learning algorithm we use to extract keywords from the documents of the dataset.

Algorithm 1 Keyword-Extractor from a document doc

```
Require: Text document doc after pre-processing steps(may be of any language)
Ensure: Set of keywords for doc and associated probability prob for each word
  $G(V, E) \leftarrow$ gow_construction(doc)
  $k \leftarrow 0$
  while $|V| > 30$ do
    $G(V, E) \leftarrow$ k_core($G$)
    $k \leftarrow k + 1$
  end while
  for all $v \in V$ do
    $C_C(v) \leftarrow \frac{1}{\sum_{u \neq v} d(u,v)}$
  end for
  prob $\leftarrow$ softmax($C_C$)
  keywords $\leftarrow$ the best four words with minimum closeness measure
  Return (keywords, prob)
```

We run the above algorithm for each document of our dataset. This extracts keywords for all the documents, which help us the figure out the content of these pages.

To understand the algorithm intuitively, we give an example of an execution of the algorithm on a document in our dataset that was containing pornographic content. The generation of a GoW for the document generated the following massive graph as shown in Figure 5.2. For better visualization, we have not shown the words associated with the nodes in the graph.

As one can see, this is a massive graph. Thus, we apply graph degeneracy method to construct a degenerated subgraph by decomposing it into its core that has fewer than 30 nodes. For example, Figure 5.3 shows the degenerated graph.

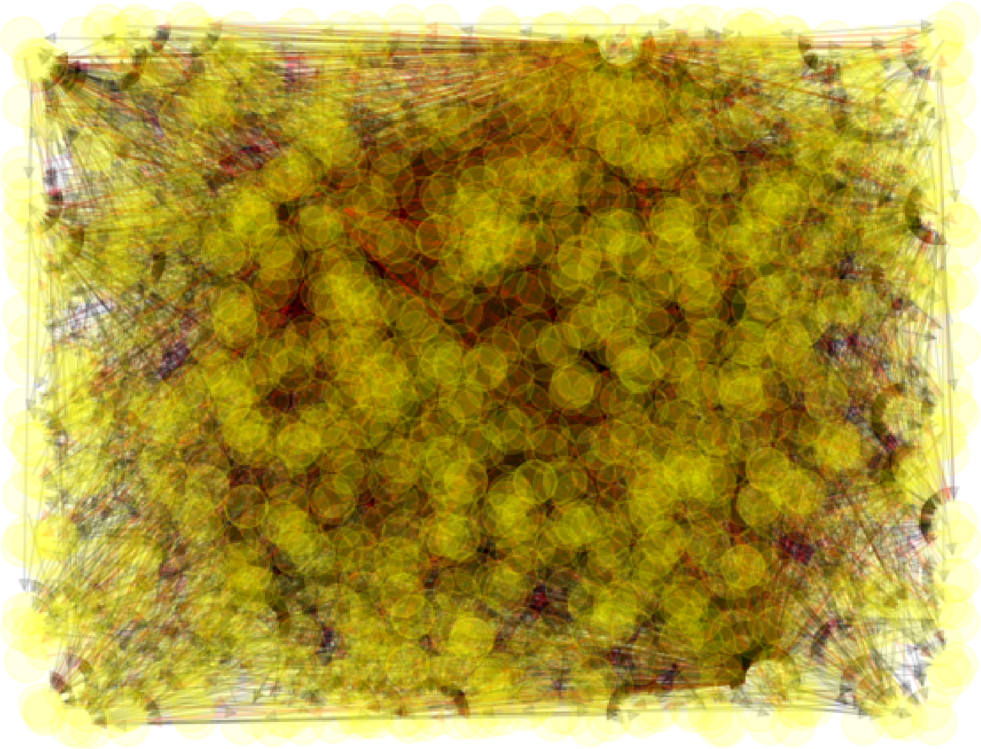


FIGURE 5.2: GoW for a document that has pornographic content

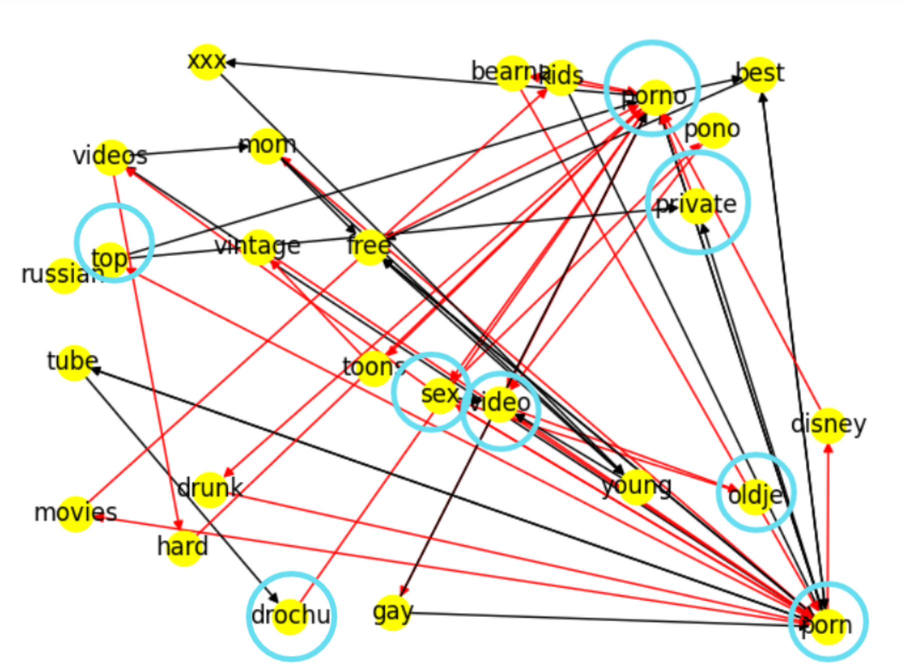
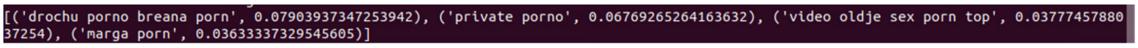


FIGURE 5.3: Caption



```
[('drochu porno breana porn', 0.07903937347253942), ('private porno', 0.06769265264163632), ('video oldje sex porn top', 0.03777457880
37254), ('marga porn', 0.03633337329545605)]
```

Figure 5.4: Keywords extracted for a document that has pornographic content

After this, we compute the closeness centrality measure for each node in the degenerated graph. We simply return the nodes with the least closeness measures as the keywords for the document. The circled nodes in Figure 5.3 are having the high closeness measures. Thus, they form a set of keywords. Hence, we get the following output as shown in Figure 5.4

5.4 Implementation

We implemented the algorithm mentioned using Python numpy library. We implemented the core decomposition module in a linear time. This is the best known (most efficient) implementation of the algorithm, which uses bin-sort as the sorting algorithm. Thus, this implementation gives us a time efficient method to extract keywords.

This implementation was run on each document of the dataset, which then gave us a set of keywords for all the documents in the dataset. This concludes the chapter.

Chapter 6

Topic Modelling

6.1 Dirichlet process

The Dirichlet process is a stochastic process that is used in Bayesian non-parametric models. Each draw from a Dirichlet process is a discrete distribution. For a random distribution G to be distributed according to a Dirichlet process, its finite dimensional marginal distributions have to be Dirichlet distributed. Let H be a distribution over theta and alpha be a positive real number.

We say that G is a Dirichlet process with base distribution H and concentration parameter alpha if for every finite measurable partition A1,. . . , Ar of theta we have:

$$G(A_1), \ldots, G(A_r) \sim Dirich(\alpha H(A_1), \ldots, \alpha H(A_r))$$

Where Dirich is a Dirichlet distribution defined as:

$$p(x_1, \ldots, x_k) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} x_k^{\alpha_k - 1}$$

6.2 Multinomial distribution

Consider a discrete random variable X that can take one of m values $x_1, \ldots, x_m$. Out of n independent trials, let k_i be the number of times $X = x_i$ was observed. It follows that $\sum_{i=1}^m k_i = n$.

Denote by π_i the probability that $X = x_i$, with $\sum_{i=1}^m \pi_i = 1$

The probability of observing a vector of occurrences $k = [k_1, \ldots, k_m]^T$is given by the *multinomial distribution* parametrised by $\pi = [\pi_1, \ldots, \pi_m]^T$:

$$p(k|\pi, n) = p(k_1, \ldots, k_m|\pi_1, \ldots, \pi_m, n) = \frac{n!}{k_1!k_2! \ldots k_m!} \prod_{i=1} \pi_i^{k_i}$$

The binomial coefficient $\frac{n!}{k_1!k_2! \ldots k_m!}$ is a generalisation of $\binom{n}{k}$.

The discrete or categorical distribution is the generalisation of the Bernoulli to m outcomes, and the special case of the multinomial with one trial:

$$p(X = x_i|\pi) = \pi_i$$

6.3 LDA

Consider a dataset of observations $\mathbf{x}$, where each datapoint is a group of observations $\mathbf{x_i} = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$. As a running example, the dataset is a corpus of documents and each document is a collection of observed words.

The mixed-membership model draws each datapoint from its own mixture model. The mixture components are $\beta = \{\beta_1, \ldots, \beta_k\}$; they are shared across the data. The mixture proportions are $\theta = \{\theta_1, \ldots, \theta_n\}$; they vary from data point to data point. The mixture assignments $\mathbf{z}$ are one-per-observation; the variable z_{ij} assigns observation $x_{ij}s$ to one of the components.

Each datapoint x_i uses all of the components β, but its proportions θ_i vary how much it expresses each one. (When θ_i is sparse then different datapoints express different subsets of the components.) Thus a mixed-membership model captures both *homogeneity*, in that all the data share the same collection of components, and *heterogeneity*, in that each datapoint expresses those components to different degree.

The corresponding joint distribution is

$$p(\beta, \theta, x) = \prod_{k=1}^{K} p(\beta_k) \prod_{i=1}^{n} \left(p(\theta_i) \prod_{j=1}^{m} \Big(p(z_{ij} \mid \theta_i) p(x_{ij} \mid z_{ij}, \beta) \Big) \right)$$

As for the mixture, the likelihood uses the assignment variable z_{ij} to select the component for observation x_{ij},

$$p(x_{ij} \mid \beta, z_{ij}) = \prod_{k=1}^{K} f(x_{ij}; \beta_k)^{z_{ij}^k}$$

This functional form is mathematically convenient.

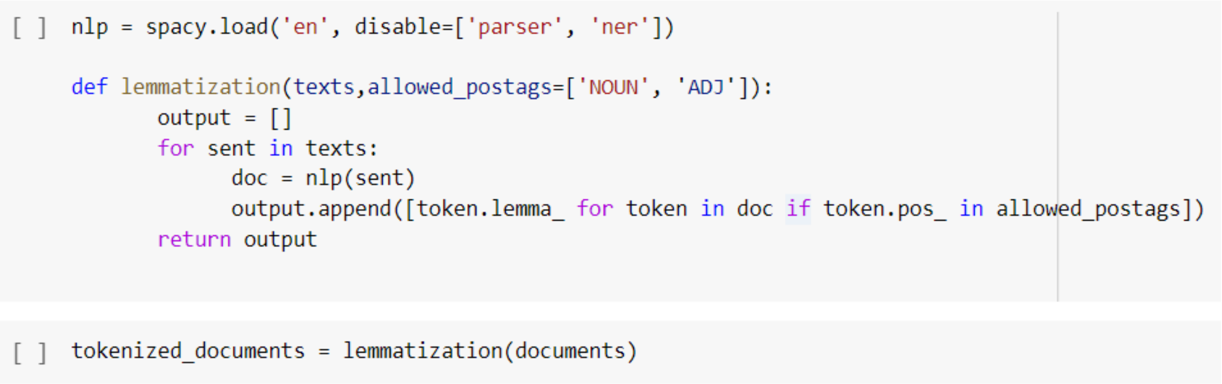
6.4 Implementation

We started with processing of the input data for our LDA model by first removing the punctuation marks from our preprocessed documents. Then we created an array of the documents with each document as an element.

After creation of an array of the documents, we lemmatized that array. Lemmatization is termed as the process of grouping together the different inflected forms of a word so that they can be analysed as a single item(lemma). For example, runs, running, ran are all forms of the word run, therefore run is the lemma of all these words.

So now the next step we performed was creation of dictionary and document term matrix(bag of words)

Here, the Dictionary function traverses the documents, assigning a unique integer id to each unique token while also collecting word counts and relevant statistics. Next, dictionary created previously must be converted into a bag-of-words To generate



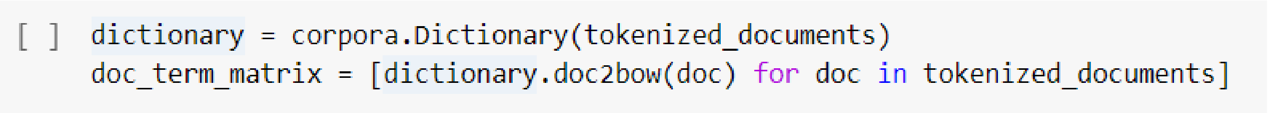
```
[ ]  nlp = spacy.load('en', disable=['parser', 'ner'])

     def lemmatization(texts,allowed_postags=['NOUN', 'ADJ']):
            output = []
            for sent in texts:
                   doc = nlp(sent)
                   output.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])
            return output
```

```
[ ]  tokenized_documents = lemmatization(documents)
```

Figure 6.1: Lemmatization

an LDA model, we need to know how frequently each term occurs within each document. To do so, we need to construct a bag of word.

The doc2bow() function converts dictionary into a bag-of-words.



```
[ ]  dictionary = corpora.Dictionary(tokenized_documents)
     doc_term_matrix = [dictionary.doc2bow(doc) for doc in tokenized_documents]
```

Figure 6.2: Creation of dictionary and document term matrix(bag of words)

The result from doc2bow(), is a list of vectors equal to the number of documents.Each document vector is a series of tuples. This list of tuples represents a sample document.

The tuples are in the form of (term ID, term frequency) pairs, so if the first tuple is (0,2) and print(dictionary.token2id) says drug's id is 0, then the first tuple indicates that drug appeared twice in the sample document.

Then we ran the lda model using gensim library giving the bag of words and dictionary we generated as parameters and some others parameters where number of topics, chunksize, passes.

Parameters:

num_topics: (required) An LDA model requires us to determine how many topics should be generated. By understanding the data and trying different values, we figured out that we should set it to 5.

corpus(Bag of Words or Document Term Matrix): (required) Stream of document vectors or sparse matrix of shape

id2word: (required) The LdaModel class requires our previous dictionary to map ids to strings.

chunksize: (optional) Number of documents to be used in each training chunk.

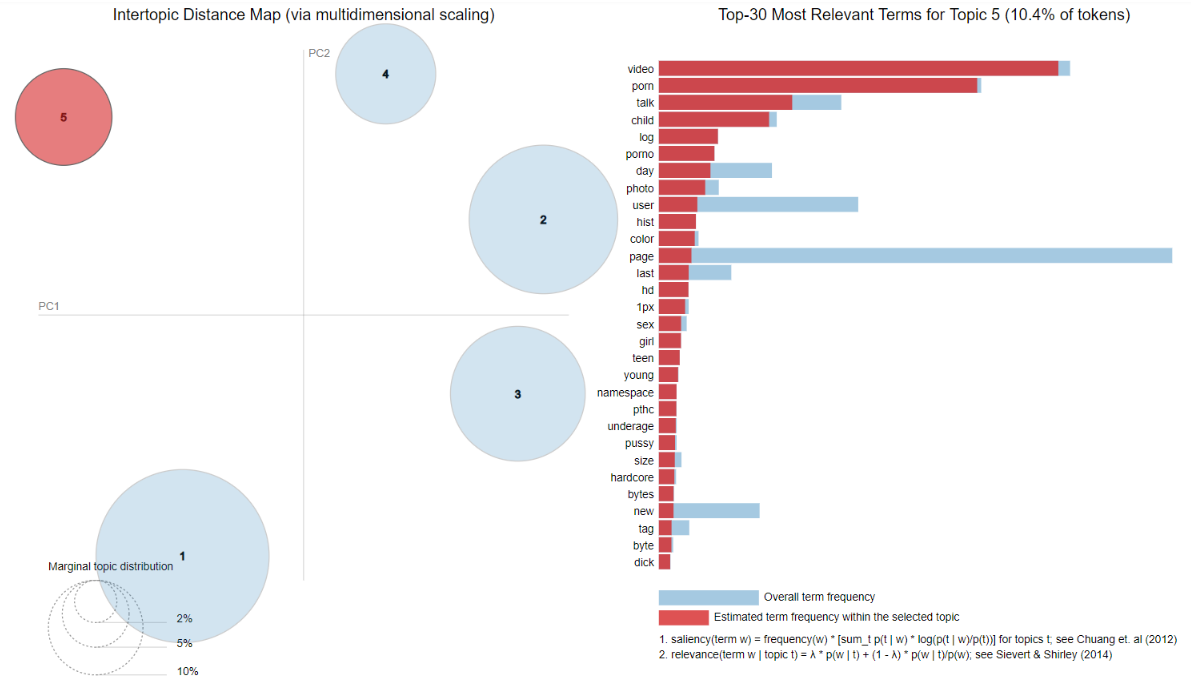


FIGURE 6.3: Visualization of the LDA model

So above is the visualization of the lda model.

In figure 6.3, the 5th topic highlighted has the words such as porn, video, child, porno, which helped as classify it to porn topic or category.

Similarly, we came up with the following topics for labelling the documents:

Hidden Wiki: This comprises a major portion of our dataset and is similar to the wikipedia of the surface web as it acts as a storehouse for onion links of different services

Marketplace: This consists of all webpages selling different products such as drugs, paypal accounts, passports, unregistered guns, bitcoins, etc

Porn: This includes different porn related web-pages

Software: This comprises of blogs, articles , discussions related to security, software updates, etc

Others: This covers every other topic not included in the above categories

Then the array of document is given as input to the LDA model object's get_document_topics() function which calculates the probability of each document in the array of being in each topic.The topic with highest probability is assigned to that particular document.

Chapter 7

Results and Comparison

In this chapter, we present the results generated from GoW Keyword Extraction and LDA model. Firstly the results obtained using GoW have been discussed in Section 7.1. We then move ahead with discussing the results obtained using LDA model in Section 7.2. Finally, in Section 7.3, we conclude our findings by comparing both the results obtained using the accuracy metric.

7.1 Results using GoW Keyword Extraction

As discussed in Chapter 5, the keyword extraction algorithm generates k keywords (in our case $k = 4$) from the individual GoWs. Using these keywords, we manually assign topic labels to all the documents.
Thus the flow of processing comprises of : converting documents to GoWs, extracting keywords from the GoWs generated, and finally manually assigning topic labels with the help of the keywords. An example is illustrated below:

58.	cards	atm	small balance	transfer	**marketplace**

The above example represents document 58, for which keywords are obtained using the Keyword Extraction algorithm and by analysing these keywords namely cards, atm, small balance, transfer, it can be easily inferred that this document must belong to the “marketplace” category.

The following table enlists keywords for two documents of each category. Along with the keywords, the probabilities with which these keywords form part of a particular category are also specified in the table.

CATEGORY	KEYWORDS			
WIKI	wiki (0.214)	account (0.200)	people (0.098)	pages (0.092)
	text page (0.217)	permission (0.182)	related logs (0.178)	title pages (0.143)
MARKETPLACE	cards (0.074)	atm (0.037)	small balance (0.034)	transfer (0.030)
	caution (0.067)	scam (0.065)	credit cards (0.023)	paypal (0.021)
PORN	jasmine sex (0.080)	girls fun camera (0.072)	school (0.053)	adult (0.052)
	glad (0.083)	lessbian (0.083)	groupsex (0.079)	zoo (0.078)
SOFTWARE	mail service (0.132)	email provider (0.098)	privacy (0.085)	inbox (0.056)
	mb uploaded (0.11)	downloads (0.10)	party analytics cookies (0.058)	github (0.050)
OTHERS	books (0.066)	libraries (0.059)	manga collection (0.036)	free ebooks (0.036)
	log type (0.11)	username (0.08)	case (0.07)	sensitive (0.07)

7.2 Results using LDA model

As discussed in Chapter 6, the LDA model generates topic labellings for the documents. The LDA model assigns probabilities to the different topics, a particular document can belong to and then assigns the label which corresponds to the highest

probability.
For example, if the LDA model generates an output of 90% for topic 0 and 10% for topic 1, then the model would label the document to belong to topic 0.

Now, we present the distribution of different words belonging to different topics that was outputted by the model.

1. Topic 1(Wiki):

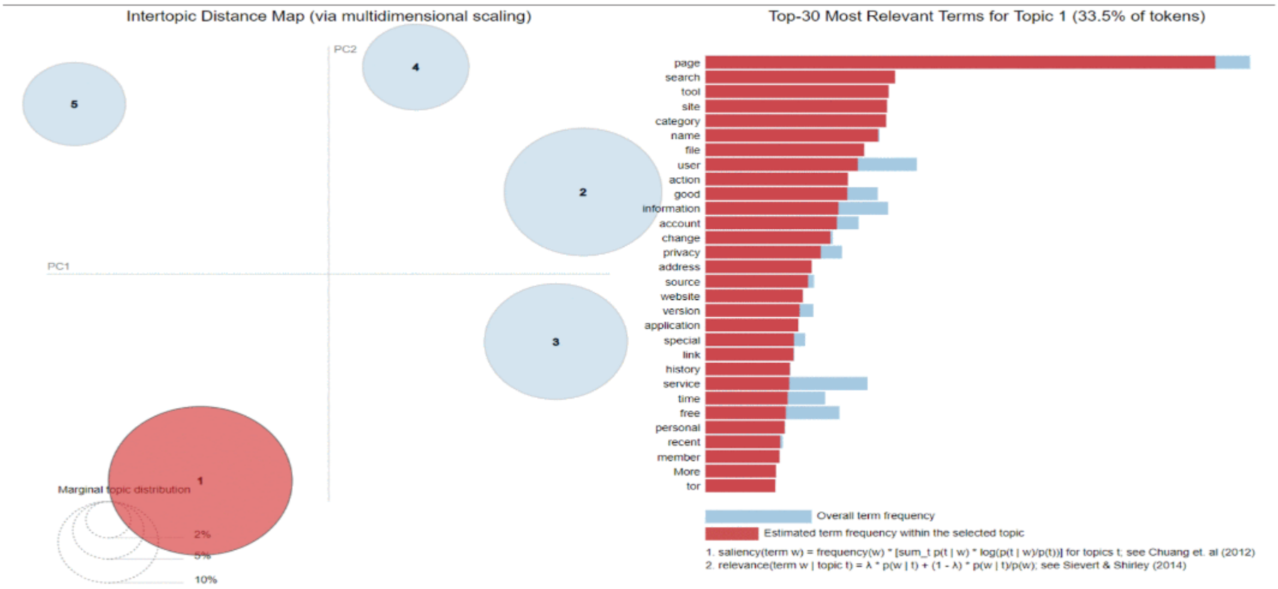


Figure 7.1: Distribuiton of keywords in Topic 1(Wiki)

2. Topic 2(Others):

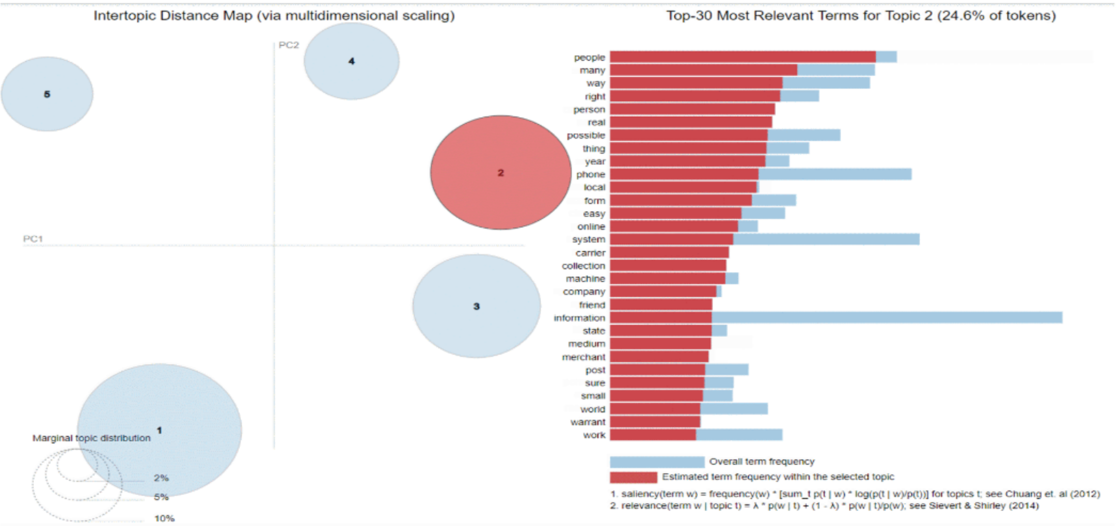


FIGURE 7.2: Distribuiton of keywords in Topic 2(Others)

3. Topic 3(Software):

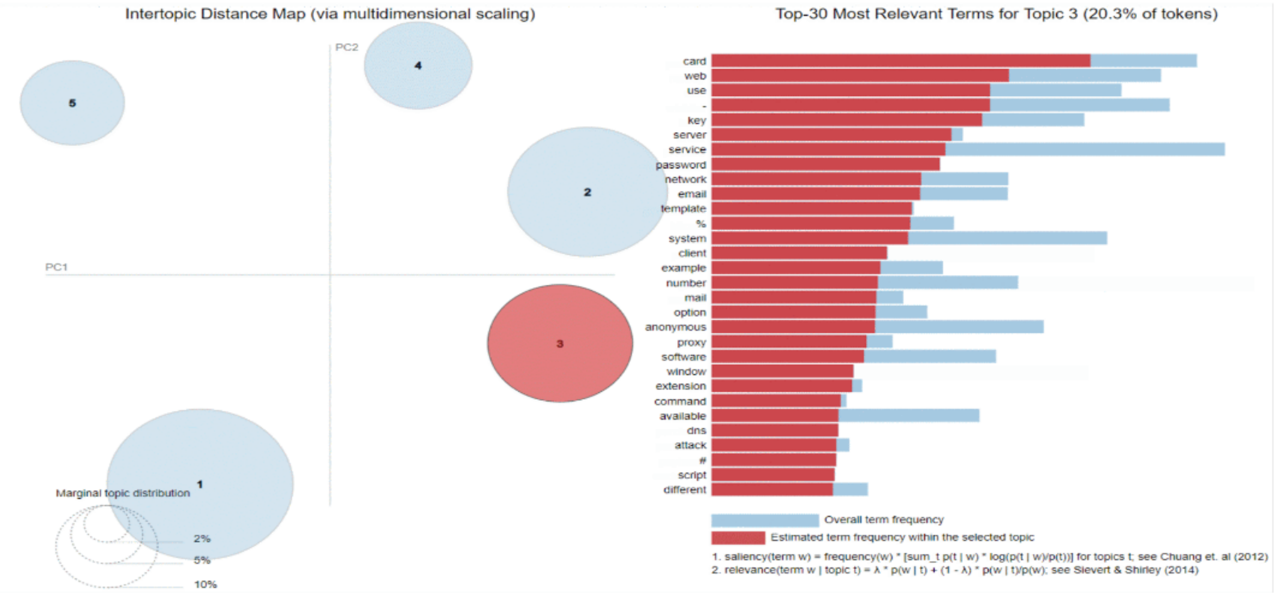


FIGURE 7.3: Distribuiton of keywords in Topic 3(Software)

4. Topic 4(Marketplace):

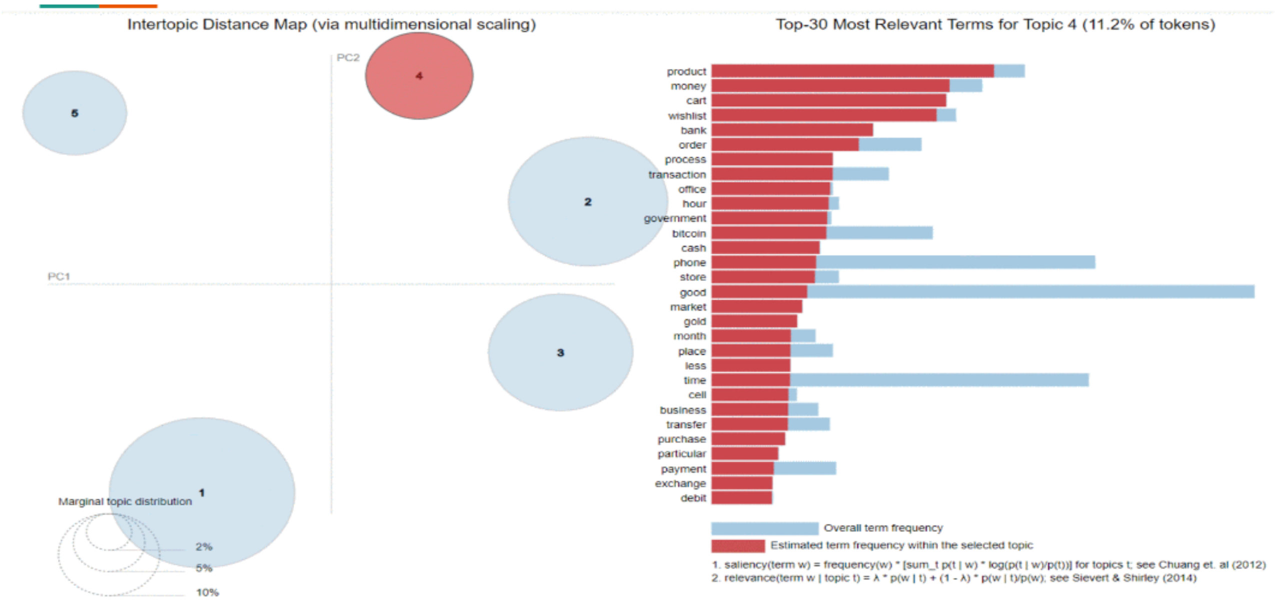


FIGURE 7.4: Distribuiton of keywords in Topic 4(Marketplace)

5. Topic 5(Porn):

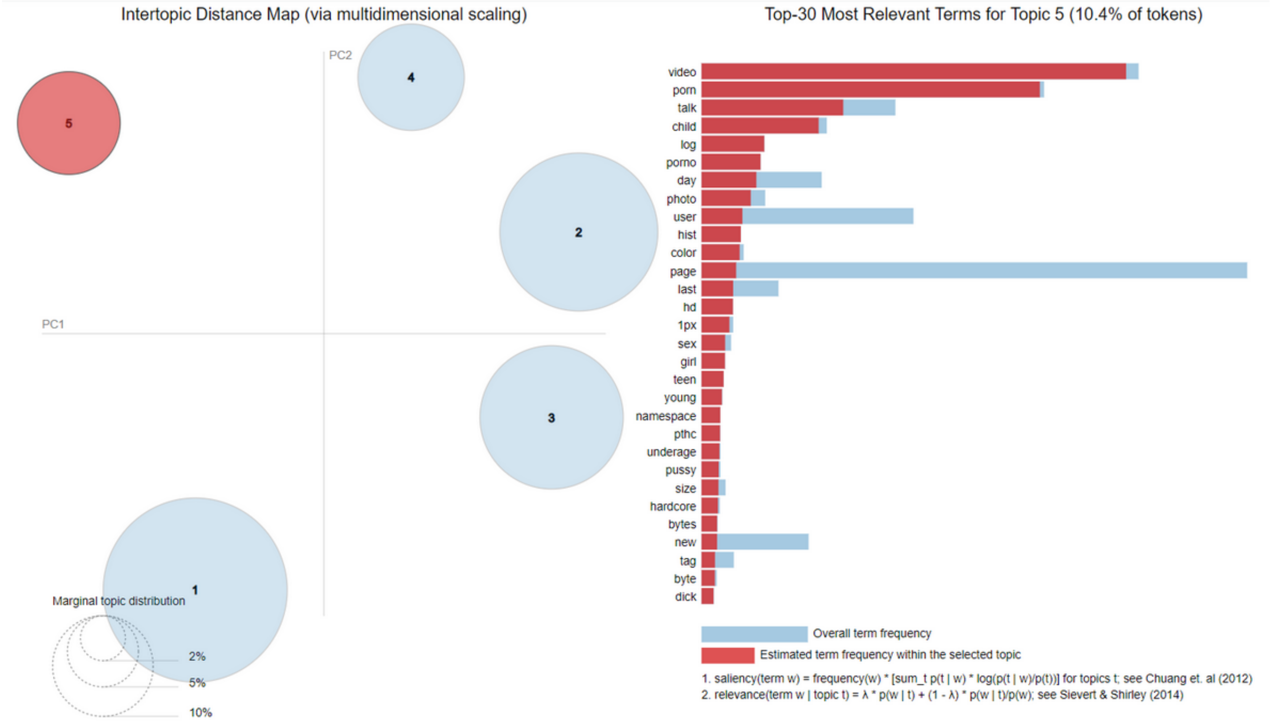


FIGURE 7.5: Distribuiton of keywords in Topic 5(Porn)

7.3 Comparing the results

The labels obtained from both the approaches are assembled into an excel sheet and we find the total number of documents for which both results match, let that be n. The evaluation method adopted to confirm the consistency of both the results is accuracy. Thus we divide this number n by the total number of documents N to obtain the accuracy metric.

The accuracy of the model was calculated to be **78.05%**. This implies that out of the total 2660 documents, nearly 2100 documents have been labelled the same topic by both the approaches. Since the first approach involves manually labelling the documents after observing the keywords, these labels can be considered as the basis, and we can thus conclude that our LDA model does a good job by providing an accuracy of 78.05%.

Chapter 8

Conclusion

While generating Graph-of-Words (GoWs), we tried to find intra-document structure. The intuition behind this was that the central words are important and hence we looked into various centrality measures like Closeness, Betweenness and Degeneracy. The usefulness of GoW representation was evaluated by comparing the graphs with the word cloud of same documents. Keywords were extracted from GoWs to get an understanding of the topic of the documents.

Later, we applied a completely different approach of topic modelling to dive deeper into the data. LDA was used to make educated guesses about how words cohere by identifying patterns in the way they co-occur in different documents. In other words we tried to find inter-document structure.

Keyword extractor using GoWs and LDA are two completely different unsupervised learning approaches and we were successful in analysing the darknet data using them.

Bibliography

Abbasi, A. and Chen, H. (2007). Affect intensity analysis of dark web forums. In *IEEE International Conference on Intelligence and Security Informatics, ISI 2007, New Brunswick, New Jersey, USA, May 23-24, 2007, Proceedings*, pages 282–288. IEEE.

Baravalle, A., Lopez, M. S., and Lee, S. W. (2016). Mining the dark web: Drugs and fake ids. In Domeniconi, C., Gullo, F., Bonchi, F., Domingo-Ferrer, J., Baeza-Yates, R., Zhou, Z., and Wu, X., editors, *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain*, pages 350–356. IEEE Computer Society.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Broséus, J., Rhumorbarbe, D., Mireault, C., Ouellette, V., Crispino, F., and Décary-Hétu, D. (2016). Studying illicit drug trafficking on darknet markets: Structure and organisation from a canadian perspective. *Forensic science international*, 264.

Deb, A., Lerman, K., and Ferrara, E. (2018). Predicting cyber-events by leveraging hacker sentiment. *Inf.*, 9(11):280.

Faisal Khan (2018). The enigma of the 'dark web'. [Online; accessed November 21, 2020].

Faizan, M. and Khan, R. A. (2019). Exploring and analyzing the dark web: A new alchemy. *First Monday*, 24(5).

Haasio, A., Harviainen, J. T., and Savolainen, R. (2020). Information needs of drug users on a local dark web marketplace. *Inf. Process. Manag.*, 57(2):102080.

Khare, A., Dalvi, A., and Kazi, F. (2020). Smart crawler for harvesting deep web with multi-classification. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–5.

Rousseau, F. and Vazirgiannis, M. (2013). Graph-of-word and TW-IDF: new approach to ad hoc IR. In He, Q., Iyengar, A., Nejdl, W., Pei, J., and Rastogi, R., editors, *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 59–68. ACM.

Samtani, S., Zhu, H., and Chen, H. (2020). Proactively identifying emerging hacker threats from the dark web: A diachronic graph embedding framework (D-GEF). *ACM Trans. Priv. Secur.*, 23(4):21:1–21:33.

Tavabi, N., Bartley, N., Abeliuk, A., Soni, S., Ferrara, E., and Lerman, K. (2019). Characterizing activity on the deep and dark web. In Amer-Yahia, S., Mahdian, M., Goel, A., Houben, G., Lerman, K., McAuley, J. J., Baeza-Yates, R., and Zia, L., editors, *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 206–213. ACM.

Zhao, F., Zhou, J., Nie, C., Huang, H., and Jin, H. (2016). Smartcrawler: A two-stage crawler for efficiently harvesting deep-web interfaces. *IEEE Trans. Serv. Comput.*, 9(4):608–620.