Universiteit Leiden
The Netherlands

Bachelor Computer Science

Topic modelling and clustering for error recognition in system logs

Stephan van der Putten (*S*1528459)

Informatica & Economie

Supervisors:
Matthijs van Leeuwen & Marcel Kolkman

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
`www.liacs.leidenuniv.nl`

17/08/2018

Abstract

Our data is a collection of server logs. The servers logs have been aggregated but have no labels to indicate what type the servers logs are. This thesis focuses on the discovery and clustering of these server logs. The focus has been on logs with the term 'error'. The research makes use of the unsupervised machine learning technique Latent Dirichlet Allocation (LDA). We extract the server logs, transform them using the standard data preprocessing pipeline. With that we created a dataset which we can call the corpus and the logs are the documents. Using the best practices from Blei the founder of LDA, we create multiple models only varying in the topic count. The models are evaluated using multiple metrics. Topic modelling can distinguish itself by being one of the few machine learning techniques which depends on human readability of its models. We take a look at the topics generated by the models and conclude that a human has a hard time understanding the topics. Clustering the documents based on their highest probable topic, shows that models only have a few dominant topics where the bulk of the documents go. The clustering has a great performance based on silhouette coefficient on lower levels. At the end of the thesis we do not recommend topic modelling for latent topic discovery on server logs. Topic modelling is not human readable on server logs and applying semantic analysis metrics does not help a lot. The clustering however appears to create solid clusters when using low topic counts.

Contents

List of Tables

Chapter 1

Introduction

1.1 General introduction

The computers that are used today have been generating data with an explosive rate in the last few years. The data is collected from different sources, transformed and aggregated to be put into a database or data warehouse. The data stays in these databases and has a huge probability to never be used and eventually to be lost. Although companies acknowledge the value of data, it is challenging to make use of said data and even more so to add value to their core business processes. A few challenges brought are due to the volume, velocity and variety of data, to name a few. The recent field of machine learning, or in a more general term data science, embraces data. The exploitation of data has improved the operation of many day to day applications in recent years.

This thesis is formed with the data Capgemini provided. Capgemini is an international IT consultancy firm that offers its customers IT services. One of these services is the big data lake which allows enormous amounts of data to be stored for further use. This data lake is built with the open source Hadoop framework. Capgemini allowed us access to their big data lake containing millions of server logs of their customers and systems. The server logs are used to find explanation for whenever server failure occurs. Manually inspecting server logs is time consuming and is costly as only domain experts understand the server logs. This brings us to the request of the company to help them get more use out of their data.

With enough creativity and time we would be able to use such a source of data to infinite use cases. Sadly, such extensive research is not possible. The research started with the analyses of the data and eventually led to applying topic modelling. Topic modelling is a form of unsupervised machine learning. Topic modelling can be described as the extraction of latent patterns (hidden topics) from data through semantic analysis. Readers who are further interested why topic modelling has been chosen, are encouraged to read Section 6.2.2.

1.2 Problem statement

Using the data to extract value is too general for a problem statement and that is why we need to specify the exact motivation and goals of this thesis. The only premise is the application of machine learning on the data. The general steps of research are exploration of the data, to see what machine learning tool is applicable and to apply and optimise the model generated (if possible). The server logs generally consist of textual messages. The messages describe the servers and range from simple informational messages to severe warning messages where user intervention is necessary. The only problem with the data is that it does not make a distinction between the types of messages. The domain expert might know which messages to search for when the servers fail, but with millions of logs the domain expert cannot be aware of all the necessary logs. Especially when the logs contain no label to indicate their type. This is one of the reasons why we propose topic modelling. Topic modelling serves as a statistical technique mostly used to reduce the dimensions of data, but can also be used to cluster similar messages.

1.3 Research question

The main goal of this research is to apply topic modelling to cluster these messages in distinct groups, we comparing these models and evaluating which model is most suitable.

We propose the following research question:

- ***Can we use topic modelling to classify and cluster error messages?***

To help us answer this questions we will distinguish the research in three sub questions:

- How does the topic count influence the topic models?
- Why are the chosen topic models suitable?
- What are our findings when applying topic modelling and optimising the model on our data?

1.4 Thesis Overview

The remaining thesis is structured as follows:

Chapter 2: Two examples of previous research are discussed. Furthermore, previous works on feature extraction from server logs is described. The building blocks used to perform this research are also described here.

Chapter 3: This chapter gives the necessary theoretical background. The first section explains machine learning. The second section is about topic modelling. The last section contains definitions of the Latent Dirichlet allocation (LDA) model.

Chapter 4: This chapter described the process of the data and the steps that are taken to prepare the data. The second section is about the 4 evaluation metrics for the model.

Chapter 5: This chapter shows the results from the application of the LDA model on the data set. The performance of our model based on the metrics, which measure the quality of the topic model showing different scores depending on the given parameters.

Chapter 6: The conclusion based on the found results. The possible future work, discussion and our final recommendation and thoughts about this research.

Chapter 7: The appendix contains examples and figures of the data Capgemini provided and the data that has been preprocessed. The pyLDAvis figures generated of our models. The topics inferred for different topic counts. Lastly the documents distribution of each topic in each of our models.

Chapter 2

Research Background

In this chapter, related research will be described. In Section 2.1, research conducted on twitter tweets and cyber security are reviewed. In Section 2.2, research that has been conducted on extracting and transforming logs to features are described. The last Section 2.3 introduces the tools and frameworks used during this research.

2.1 Topic modelling in Twitter and NCSA

In this section, we briefly describe and show two researches that have applied topic modelling for different use cases. The descriptions will end with an overview explaining which parts are relevant to our current research.

A research conducted in 2011 makes use of the numerous amounts of tweets on Twitter. Twitter is an online platform used to send messages about social media and news. Users make posts called tweets that are restricted to 140 characters. The authors are interested in finding news topics from twitter feeds and comparing the topics with traditional news feeds using Latent Dirichlet allocation (LDA), see Chapter 3 for more explanation.

The authors start with preparing three monthsẃorth of tweets of users and news feed data from New York Times. The tweets are filtered on stop words and tweets appearing more than 70% of the time and less than 10 times are removed. The authors recognise that LDA does not perform very well on small tweets. To better fit the data to LDA, the authors aggregate the tweets of each user to a single document. The data is represented in a Bag of Words matrix and fitted to their custom Twitter-LDA model. The authors continue comparing the traditional LDA model and their own model using two human judges. The judges compare the generated topics, which are 10 words long, using a self-created scoring mechanism. The scored is based on each topics distinction and cohesion of words. The final results show that their Twitter-LDA model is better compared to the traditional LDA model. The results of their model showed more informative topics, which could not be extracted from the traditional news source [ZJW+11].

The second research is published in 2014 with the intention of exploring a big data use case. The National Centre of Supercomputing Applications (NCSA) generates around 4.5 GB of data each day which are collected from security monitoring and system logs. The authors study a new approach of LDA on logs files for intrusion detection, moreover the authors propose that LDA can be used to detect patterns in log events. The model is trained to recognise normal activities and abnormal behaviour patterns found in the probability distribution of the topics over their data set. The resulting trained model is capable of detecting the semantics of the log events, which provide a higher level description of the intention of an invading user [HKN14].

In our research we will apply ideas mentioned in both researches. In the first paper we recognise two similarities with our own research. Our data set is similarly structured to tweets, which are short and domain specific. The second paper proposes an approach for using log files to detect intrusion, which is similar to our problem of detecting errors. Furthermore, the evaluation in both researches of their own model is based on human judgements and the semantic pattern recognition quality, which both depend on the distinct nature of the topics using cohesion and distinctiveness. In other words, the quality of our model is based on the topic quality detected after applying LDA.

2.2 Feature extraction from logs

Numerous log files are outputted by different servers, e.g. web server logs, system logs, etc. A log includes data that can be numerical or non-numerical data, depending on the format and source of the log. Intel researched log based predictive maintenance back in 2014 [SFMW14]. The logs contained 3 types of information which was used for feature extraction. The content of each log can be used for feature extraction which depends on the type of data. Keywords from textual messages were extracted using parsing and transformed to a Bag of Words representation. The numerical values were decoded and event codes for sequential analysis. We apply the same text mining techniques to extract only the textual content from our data and transform the data to a usable state.

2.3 Building blocks

This section has an oversight of the main tools and packages used. This section might be mentioned or referenced in later parts of this paper. The packages were required for the extraction, loading and transforming the data in a usable form.

2.3.1 Packages and libraries

1. **Hadoop** (`http://hadoop.apache.org/`)
   The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.

(a) **HDFS** (`http://hadoop.apache.org/`)
A distributed file system that provides high-throughput access to application data. Used to store big data.

(b) **Apache Spark** (`https://spark.apache.org/`)
Apache Spark is a fast and general engine for large-scale data processing.

(c) **Apache Zeppelin** (`https://zeppelin.apache.org/`)
Web-based notebook that enables data-driven, interactive data analytics and collaborative documents with SQL, Scala and more.

2. **Scikit-learn** (`http://scikit-learn.org/`)
The Scikit-learn package contains tools for efficiënt data mining and data analysis with machine learning in Python.

(a) **Pandas** (`http://pandas.pydata.org/`)
Pandas is a library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

(b) **Numpy** (`http://www.numpy.org/`)
Numpy is a scientific package with Python for powerful array objects, functions and lots of mathematical capabilities.

(c) **SciPy** (`http://www.numpy.org/`)
SciPy is a scientific package for mathematical, scientific and engineering tools used in Python.

(d) **Matplotlib** (`http://matplotlib.org/`)
Matplotlib is a 2D Python plotting library very similar to MATLAB.

(e) **Seaborn** (`https://seaborn.pydata.org/`)
Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

3. **Gensim** (`https://radimrehurek.com/gensim/`)
Gensim is a free python package created for scalable statistical semantics, analysing plain-text documents for semantic structure, retrieving semantically similar documents.

(a) **pyLDAvis** (`https://github.com/bmabey/pyLDAvis`)
The pyLDAvis package is designed to help users interpret the topics in a topic model that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualization and can be used in combination with sklearn and gensim.

(b) **Nltk** (`https://www.nltk.org/`)
Nltk is a leading platform for python to work with the human language. The tool can be used to

perform every step to transform the human language in a workable form.

4. **Conda** (`https://www.anaconda.com/`)
   Conda is an open source package and environment management for Python. Conda allows easy setup with out-of-the-box environments for quick testing and removal of environments.

   (a) **Jupyter Notebook** (`https://jupyter.org/`)
   Jupyter notebook is included in the standard data science conda package. A fast web application used to create documents in Python code to easily share code and visualise data.

   (b) **Python 3.6.X** (`https://www.python.org/`)
   An user-friendly and elegant programming language which has a great scientific community.

Chapter 3

Theoretical Background

In this chapter we explain the necessary background of our research. First, in Section 3.1 we describe the general term of machine learning. This brings Section 3.2 with general explanation of topic modelling. The last Section 3.3 describes Latent Dirichlet allocation.

3.1 Machine learning

The idea of self-learning computers has been conceived multiple decades ago, but has only recently been greatly applied in our society. In the field of computer science, machine learning is defined as follows [Sam59]:

Definition 3.1.1. Machine Learning
Machine learning gives computers the ability to learn without being explicitly programmed.

Machine learning exploits the abundance of data which it applies to learn in one of the following three ways:

1. Supervised learning
2. Unsupervised learning
3. Semi-supervised learning (or reinforcement learning)

The distinction between the learning methods depends on the data. Either data is labelled or unlabelled data, e.g. pictures of named butterflies or unnamed butterflies. Supervised learning makes it easy to train and evaluate the model using labelled data. Unsupervised learning makes use of the (latent) patterns found in the unlabelled data. Reinforcement learning uses a mixture of labelled and unlabelled data to train itself.

3.1.1 Hard and Soft clustering

Machine learning divides data into clusters, either hard clusters or soft clusters. The results discussed in Chapter 5 contain results based both upon hard and soft clustering. Clustering data can be achieved by giving every element of the data at most one label, this is called hard clustering. Soft clustering allows data to be part of multiple clusters or contain multiple labels. Especially when talking about topic modelling, the distinction between hard and soft clustering can be vague. In topic modelling we assume every document is a mixture of multiple topics; this by itself is already a form of soft clustering. In Section 3.3 we further discuss the meaning of clustering in our model.

3.2 Topic Modelling

Topic models are models used to find latent topics in mostly large unstructured collections of documents. Topic modelling assumes that documents are a mixture of topics, while topics are a distribution of words [MBCD10]. Whereas humans have a hard time to find a structure, topic modelling uses statistical methods for analysing words for topic discovery. This makes it possible to compare topics with each other and to find similar documents without necessarily having any prior knowledge of the collection of documents. The application of topic modelling is wide and is very powerful, making it a very popular method for the exploration of data.

SYMBOL	DESCRIPTION
K	Number of Topics
V	Number of words in the vocabulary
M	Number of documents
N	Number of words in the document
$N_{d=1..M}$	Number of words in document d
α	Collection of all $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_K\}$
$\alpha_{k=1...K}$	Hyperparameter for Dirichlet prior distribution of topic k
β	Collection of all $\beta = \{\beta 1, \beta 2, ..., \beta_K\}$
$\beta_{w=1...V}$	Hyperparameter for Dirichlet prior distribution of a word w in a topic
$\varphi_{k=1...K}$	Distribution of words in topic k
$\varphi_{k=1...K,w=1...V}$	Weight of word w in topic k
$\theta d = 1...M$	Distribution of topics in document d
$\theta d = 1...M, k = 1...K$	Weight of topic k in document d
$z_{d=1...M,w=1...N_d}$	Assigned topic of word w in document d
Z	Topic of all words in documents
$w_{d=1...M,w=...N_d}$	Assigned word w in document d
W	Words in all documents

Table 3.1: Complete notation of LDA

The machine learning and text mining areas have focused a lot on probabilistic topic models in recent years. Models like probabilistic latent semantic analysis (PLSA) and sentiment analysis are used for applications ranging from document clustering, topic modelling and retrieval systems [LMZ11]. Optimising models is also a challenging task, current models may make usage of a high range from statistical inference e.g. variational, stochastic variational and Markov chain Monte Carlo [Hof17]. The model that is used in this research and build upon the before mentioned models will be discussed in great length below.

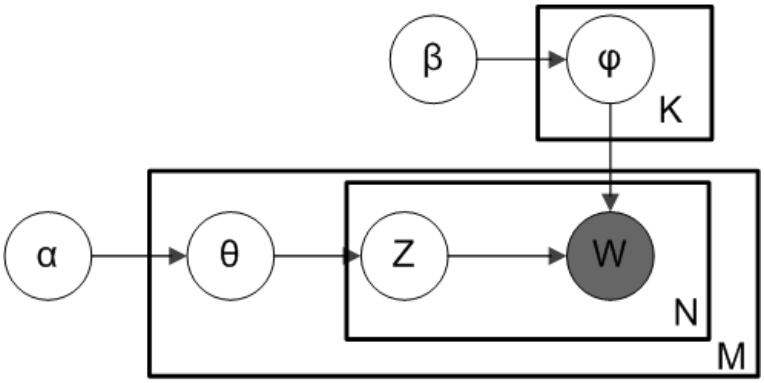


Figure 3.1: The smoothed LDA plate notation [Dav03]

3.3 Latent Dirichlet allocation

In natural language processing, *Latent Dirichlet allocation* (LDA) is an unsupervised machine learning technique introduced in 2003 for Topic modelling [Dav03]. The notation used for LDA can be seen in Table 3.1. LDA is part of a larger field called probabilistic topic modelling [MBCD10].

LDA makes use of a generative probabilistic model of a collection of documents **M** (corpus) to discover latent topics. Fig 3.1 represents the plate notation of LDA. For a more understandable model consider Fig 3.2. The model assumes that each document **N** in the corpus consists of a mixture of latent topics. These topics are a mixture of words **W** assigned to a topic from a fixed vocabulary **V**. **Z** notates the assignment of specific words to topics. The distribution of words θ (theta) for each topic is dependent on the sensitivity of α (alpha). The probability distribution of topics in documents φ (phi) are dependent on the sensitivity of β (beta). The number of topics **K** are predefined by the user.

The LDA model is defined in 3 steps and shown in Fig 3.2 [Dav03]:

1. For each document, pick a topic from its assigned distribution over topics.
2. Sample a word from the distribution over the words associated with the chosen topic.
3. This process is repeated for all the words in the document.

Let us once again look at the mentioned Fig 3.2. The topics are shown on the left side with their probability of words. On the right side, the document has a topic proportion. Every word gets assigned to a topic so that the topic proportion matches. In the original LDA model, assignments of words get updated every iteration through the corpus **M**. Restarting the process again until the LDA model converges and the topic and assignment are stale. The eventual quality of the model depends on the assumed hyper parameters α and β and parameters θ and φ. For a better understanding of the parameters take a look at Section 3.3.1 and Section 3.3.2.

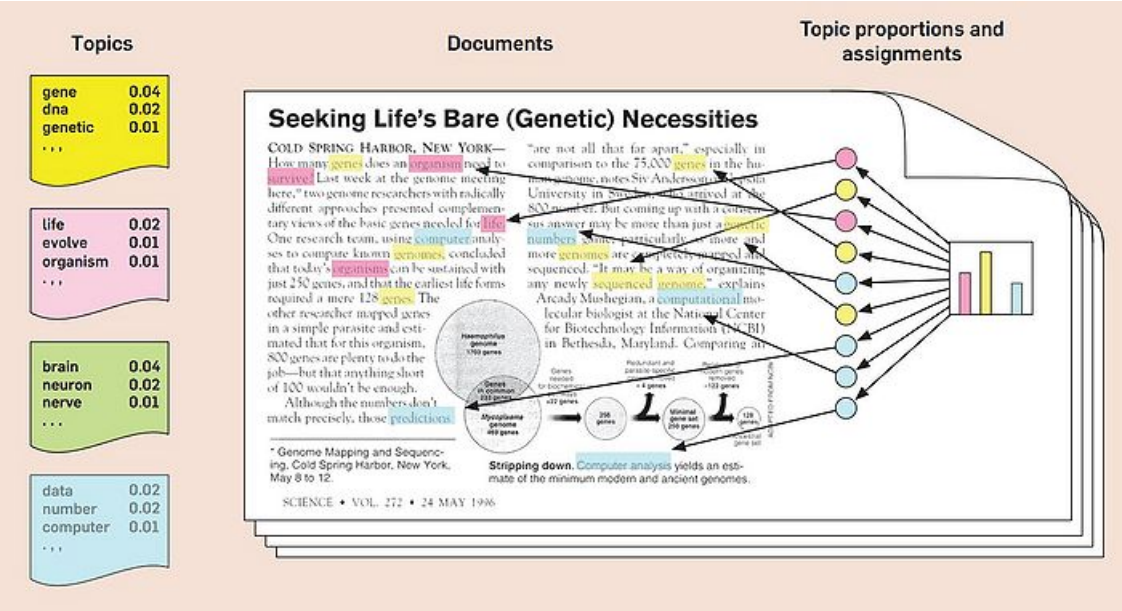


Figure 3.2: LDA applied to a document [Dav03]

3.3.1 α and β hyperparameters

The Dirichlet is defined as the distribution over a distribution. Dirichlet is used to infer a posterior distribution after observations using a prior distribution [Set94]. Hyperparameters are defined as parameters that assume a prior distribution before any evidence or information is taken into account. This distinguishes α and β from the remaining parameters. α and β are both Dirichlet distributions. The hyperparameter value α is the parameter of the Dirichlet prior on the per-document topic distributions. The result of a high value of α is a document with a mixture of most topics, while the low value leads to documents with more distinct topics. The α results in a corpus with distinct documents or more general documents topic assignments. The value β is the parameter of the Dirichlet prior of a word in a topic. A high value for β means that the topics consist of distinct words, the low value of β assumes topics are more generative. The β will influence the general distribution of words in topics in the output of topics. We base our α and β on earlier research of Blei which recommends that α=1.0/T with β=0.01 which shows to work great with many different corpora.

3.3.2 θ, φ parameters

The parameters θ, φ are dependent on the prior distributions α and β. θ is the document-topic distribution. θ is the weight of a topic in a document and because α is a prior distribution, θ assumes a distribution based on the previous assigned distribution. In the same way φ is the weight of words in a topic. φ in this case is dependent on the β.

3.3.3 Online Latent Dirichlet allocation

The online variant of LDA was introduced by Hoffman et al. in 2010. [Mat10] This new variation dealt with the problem earlier LDA models struggled with. The problem that LDA had was the computing of huge collections of documents. The online LDA can be used for massive- and streaming documents without losing performance compared to the original LDA model, because it analyses the documents in batches instead of single observations with stochastic (random) optimisation [Mat12]. It simply allows models to be updated rather than being computed again. This research also assumes the online model for similar performance and improved computational time.

Chapter 4

Methodology

In this chapter, the dataset and methodology are described. Section 4.2 examines the raw data. In Section 4.3 the steps of transforming the data is described. Section 4.4 is a brief look at the data. Lastly, Section 4.5 describes our measures to evaluate our models.

4.1 Data collection

The original data is collected using the Hadoop framework tools named in Section 2.3. HDFS is a storage system which collects the different servers logs from different type of servers. With the help of the available tools we extract the unstructured data and use the existing library to structure the data. PySpark allows quick in memory computation to transform and slice the data further. Which eventually allows us to transform the data to the friendlier pandas dataframe and make local computations possible.

4.2 Dataset

The dataset used in this research is provided by Capgemini containing syslogs from various servers, see Appendix A 7.1. The syslogs contains server logs from different servers provided to their customers and internal staff. The event logs used on their servers were extracted from different operating systems, ranging from the year 2015 to current day. The size of one day of data can easily range into the 20 - 40 million server logs. Due to the size and complexity and computation time of the dataset and the focus on discovering patterns in error logs in the unlabelled data, we extracted the data that contained the word ”error”.

The filtered dataset has been transformed to a more suitable pandas dataframe in Table 4.2. The column syslog.body contains textual messages displaying the messages from the server. Manual inspection has shown that the contextual data sent by the server is displayed after the square bracket. We simply filtered the data away before the square bracket using regular expressions as this contains low to non-existent value for our

research. With that said the example below in Table 4.1 will be left with: *RestClient: HTTP request for url /migrate/ping failed with error code 12029 (source: 5).*

- - - [Originator@6876 eventid="300" keywords="Classic" level="Error" channel="Application" vmw_host ="vbk-dca-esx-020.piv.local" vmw_vcenter_id="7CE97301-0536-455B-9466-475070F453E3" vmw_vcenter= "PRD vSphere Environment" providername="Engine" vmw_vr_ops_id="58640a59-4d7f-42fe-9ff1-d54fa1f595f6" vmw_cluster="css01-piv-01" vmw_datacenter="Pivotal-DCA" task="None" vmw_object_id="vm-6643" eventrecordid="64509969"] RestClient: HTTP request for url /migrate/ping failed with error code 12029 (source: 5)

Table 4.1: Full length syslog.body message

	hostname	uuid
0	piv-prd-os-362.iddsaprod.lan	a6e7c3d3-c8b4-4cc4-835f-cf6643b76622
1	piv-prd-os-362.iddsaprod.lan	ae4ad420-5120-4345-8c43-11b3db6cae65
2	vbk-dca-esx-033.piv.local	1797b5ff-d4f7-43dd-b99c-64ae6e688fdf
3	piv-prd-os-362.iddsaprod.lan	a47434ef-07ba-47a5-85d6-e2927065b4c1
4	piv-prd-os-362.iddsaprod.lan	5f9c1433-d3a5-4f15-bfa4-50ffdc913a0a

	syslog.body	timestamp
0	- - - [Originator@6876 eventid="300" keywords=...	2017-05-01T21:04:20.0Z
1	- - - [Originator@6876 eventid="300" keywords=...	2017-05-01T21:04:19.0Z
2	sfcb-CIMXML-Processor 7620520 - [Originator@68...	2017-05-01T21:04:20.557Z
3	- - - [Originator@6876 eventid="300" keywords=...	2017-05-01T21:04:20.0Z
4	- - - [Originator@6876 eventid="300" keywords=...	2017-05-01T21:04:20.0Z

Table 4.2: The local dataframe

Now that essential part has been extracted from our data, we discard the remainder and will continue the process in the next section.

4.3 Data preprocessing

Our goal now is to prepare the data such that our model can accept the data. The data needs to be converted to the Bag of Words representation. Preprocessing involves normalization, tokenization and stop word removal discussed in Section 4.3.1 - 4.3.4. Preprocessing is important and can greatly influence the final results. This can be related to the garbage in, garbage out principle in computer science, where flawed input brings nonsense output. Following the same principles laid in the Section 2.2 we will walk through every step. This said, we can only use the data contained in the dataset with feature 'syslog.body'. This feature contains the messages.

The preprocessing of our data is defined in 4 steps:

1. Normalization
2. Stop Words
3. Tokenization
4. Bag of Words

Before such steps are taken we have an overview of our current dataset and complete dataset mentioned in Table 4.3. The corpus contains 426905 records and has messages consisting of small twitter like sizes. The complete dataset has documents which contain a few columns like hostname, severity, port, priority, valid, protocol, body. The features contain little information and will not be used, only the body feature has been extracted which contains the textual message of the syslog. Furthermore, we assume that our syslogs are normal textual messages and our Bag of Words matrix will only contain 1-gram words, unless we state otherwise. The data will be referred as the corpus and the syslogs as documents in the remaining paper.

Collection	Documents	Words	Vocabulary
Complete dataset	18369485	310 million	N.A.
Error dataset	426905	3428621	1694

Table 4.3: Statistics about the dataset before processing

4.3.1 Normalization

In the text mining world, we define normalization as follows:

Definition 4.3.1. Normalization
Text normalization is the process of transforming text into a single canonical form that it might not have had before [Wik18].

Normalization in normal documents is simply done by removing the punctuation marks and the lowercase of each word. The nature of our server logs does not allow that. The message concealed in Table 4.1 has an error code and extra information between the parenthesis's. We choose to keep the additional information closed between the parenthesis's, using python to write a custom function. The next steps are removing the digits and remaining punctual marks, lowercase every word and remove the remaining unnecessary whitespace.

4.3.2 Stop word

The second step for our preprocessing is removing Stop Words. Once again Stop Words are:
Definition 4.3.2. Stop Words
Stop Words are words which are filtered out before or after processing of natural language data [LRU14].

LDA assumes that each word is equally important. We assume that each word is not equally important which is why we remove the unimportant words, called stop words. Words such as 'the', 'a' and 'an' are not important, generic English words can be removed to only keep the most distinguishable words left. The tools in NLTK provide a standard tokenization option with stop words from the English vocabulary. This tool removes remaining ambiguous words and leaves the most important words left that are specific to the error message.

4.3.3 Tokenization

The third step in our preprocessing is the parsing of the documents. After the stop words have been removed, we are left with documents containing highly specific words. The tokenization process takes care of the remaining text and parses the documents. Further vectorization is needed to transform the tokens into the Bag of Words representation.

4.3.4 Bag of Words

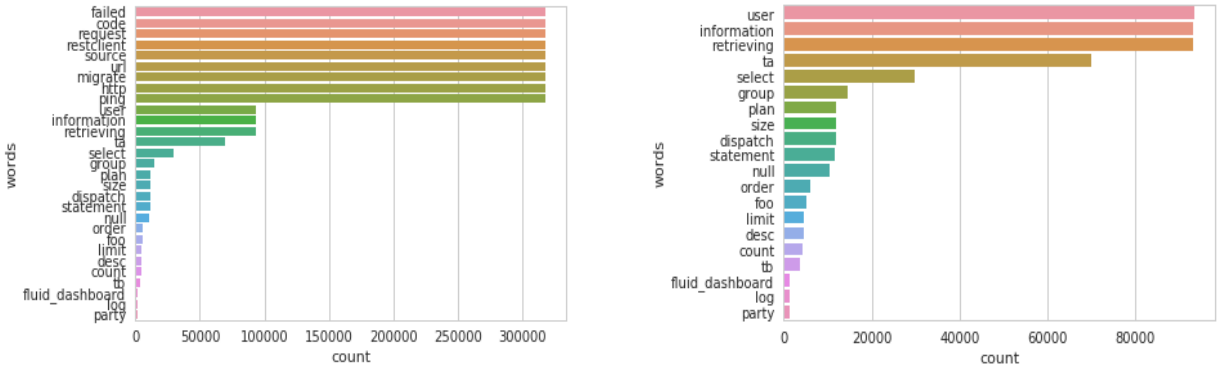
The text preprocessing in this research makes usage of the Bag of Words representation as LDA makes the assumption that the order of the words does not matter [MBCD10]. Bag of Words counts the words that appeared in the document and represent the words in a document as a term-frequency matrix (tf-matrix). The Bag of Words representation of a document does not take in the order or semantic structure in a document, but LDA discovers these semantic structures itself.

4.4 Data exploration & Visualisation

In this section we will further explore our resulting tf-matrix and mention the model parameters.

4.4.1 Term frequency matrix

The remaining tf-matrix will be once again filtered. The words which occur in more than 90% of the occurring documents and words appearing less than three times will be removed. Having words that are too frequent have no to little information gain and terms that occur less than two times will not be relevant enough to keep. Our resulting words with their respective frequency is displayed in Fig 4.1. This leaves our dictionary with a vocabulary of 1570 words in total.



(a) Atleast 1000 times

(b) More than 1000 and less than 250000 times

Figure 4.1: Words and counts appearing in the corpus

4.4.2 Splitting the data in Train Test and Held out

The data is shuffled using sklearns build in shuffle. We use 90% of the data for out train and test data and the remaining 10% for held out data. The size our train test is 384215 documents and held out is 42690 documents.

4.4.3 Dimensionality reduction

The reason LDA is widely applied for text clustering is because LDA actually reduces the dimension, reducing the computation time. LDA reduces the dimension of a document (in our case with the shape (doc, term) to a (doc,topic) shape. The generated output is soft clustered, but will be also hard clustered. Each topic in the (doc, topic) output corresponds to the probability the documents belongs to that topic.

4.4.4 Model Building

Building further on the research done on LDA, the before mentioned research in Chapter 2 and more recent research with online LDA. Table 4.4 displays the parameters set to test our model. The named parameters are explained in more detail on the gensim webpage in Section 2.3.

Parameter	value	description
α (alpha)	1.0/Number of topics	A prior belief of each topic
β (beta)	0.1	a prior belief of each word
κ (kappa)	0.5	weight of word remembered each topic
τ (tau)	64	weight of topic remembered each document
num topics	K2-38	Number of latent topics to be extracted
chunksize	2000	Number of documents in each batch
passes	1	number of passes through the corpus
update every	1	sets the model to online version
iteration	50	maximum number of passes through the corpus when infering corpus
scorer	perplexity	uses perplexity to train the model on held out test data
gamma threshold	0.0001	minimum chance in document needed to continue iterating
eval every	10	Estimates perplexity after update
id2word	dictionary	used to map the Id to word

Table 4.4: Parameter settings

4.5 Model evaluation

Each step in our process needs to be evaluated. Some evaluation metrics are already included and applied by our models. Others are common practice and applied to our results. The following metrics to evaluate are coherence, perplexity, silhouette coefficient, human readability and Jensen shannon.

4.5.1 Coherence

We use coherence to evaluate the topics coherence. Coherence is the distinctiveness of each topic. This can be achieved using a human interpretability score [CGWB09] or other measures. In a research conducted in 2015 we find that the Cv measure outperforms other topic cohesion measures. The measure correlates well with human interpretability [RBH15].

4.5.2 Perplexity

In the original paper Blei introduces a general model evaluation metric [Dav03] to compare topic models. Perplexity can be used to compare the generalisation of a model on new unlabelled dataset.

$$perplexity(\mathbf{D}_{test}) = \exp\left\{ -\frac{\sum_{d=1}^{M} logp(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d} \right\}$$

Perplexity shows the perplexity on the test set of held out documents **D**. The nominator shows the sum in corpus M with document d, where the likelihood of each word in d is computed. The denominator consists of the count of words N in document d. The lower the perplexity score the better a model generalises.

4.5.3 Silhouette coefficient

The silhouette is used to measure between the cohesion and the separation of intra-clusters. In our model this measures the mean intra-cluster distance for each document and compares distance to the nearest-cluster distance.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Where s_i is the silhouette of sample i in the cluster. a_i is the average distance for i from all the objects in the cluster and b_i the distance of i from the closest cluster b not containing i.

$$-1 \leq s \leq 1$$

The value of s will be contained between -1 and 1. If $s(i) = 1$ then we can say that the distance i is a lot less in its own cluster then the nearest other cluster. If we take $s(i) = -1$ then the similarity of i is higher in the other nearest cluster then its current cluster [Rou87].. Commonly used with cosine for document cluster evaluation.

4.5.4 Jensen-Shannon divergence and KL-divergence

The Kullback Leiber divergence was introduced to measure the density between two distributions [HO07]. Based upon this important and popular measure the Jensen-Shannon divergence (JSD) was introduced [FT04]. Which is better used to measure similarity between two text documents based on their probability distributions.

$$JDS(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M)$$

Where P and Q denote a probability distribution and M the set of probability distributions. Taking the square root of JSD makes it truly a metric which can be used to measure similarity [Hua08]. The metric is especially useful to find the distinctiveness and cohesion between topics.

4.5.5 Human perception

Although LDA can be used to find latent patterns, explore, tag recommend in a document corpus the final result of topics do not necessarily match up with the human expectation of a topic. Especially in an unsupervised learning model with only mathematical measures [TRH16]. The reason of this paragraph is to make readers aware that the suitability of a model in an unsupervised learning and NLP environment still need support of a human factor. Research from Chang et al. [CGWB09] and Blei et al. [CB12] provide more in depth research in this topic.

With that in mind, the highly dimensional LDA is actually suitable in contrast to most machine learning algorithms to be evaluated using visual tools. One such tool is LDAvis, a tool that got developed in R and D3 [SS14]. LDAvis is a web-based interactive visual of the topics on a fitted LDA model. The multidimensional LDA is scaled to two dimensions, making it possible to visually see the distance between topics and quickly determine their distinctiveness. Simultaneously the visual tool shows the relevance of each term in their selected topic, based on their exclusiveness and occurs within that topic compared to different topics. The implementation that we use is created with python and is named pyLDAvis, see Section 2.3.

The evaluation metrics will be applied on the models. Based on the exploration of the data we will also choose the optimal number of topics. In the following Chapter 5 the results are shown.

Chapter 5

Results

This chapter examines the results achieved through our experimental work. Starting off we will show the resulting models through pyLDAvis in Section 5.4. The topics are evaluated and compared with a coherence score in Section 5.5. The tables show multiple topics with their top terms in Section 5.3. The document distribution based on most relevant topic after applying the model on the corpus in Section 5.6. Ending with a silhouette coefficient score for the models in Section 5.7. The models were generated using the gensim package and the results compare the train, test and held out data where applicable.

5.1 Model results

Our results are created and evaluated with multiple aspects in mind. The human interpretation and semantic analysis of the documents have been leading for our models and results. Each section discusses one of these aspects. The experiments are conducted with the gensim package. The models have been created with 2 till 38 topics. The default settings in gensim allowed our models to be trained on our test and train data, further explained in Section 4.4.4. Section 5.2 - 5.5 used only the train and test data, the remaining sections compare the held out data.

5.2 Wordcloud

We represented our terms in a wordcloud, Fig 5.1, which as the name implies simply shows the terms in the topics created by our models in a cloud. Nothing special so far to behold, but a quick glance on the words show words which are very domain specific as such we cannot easily interpret the meaning behind the words or their relation.

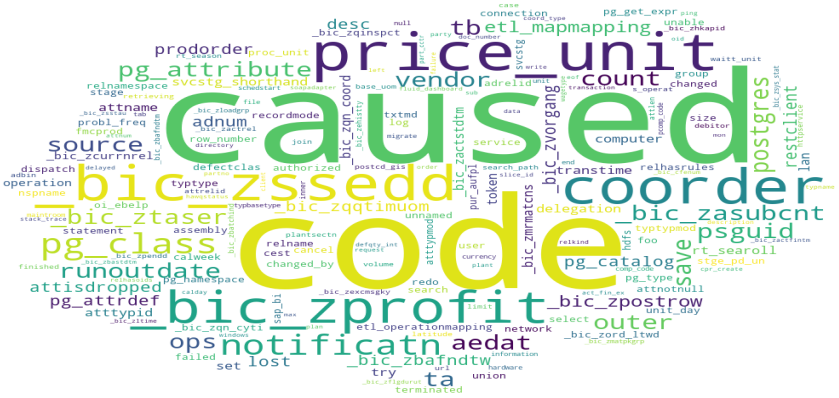


Figure 5.1: A wordcloud containing terms found in clusters

5.3 Topic overview

First we examine the latent topics our models have generated. We showcase the topic count 2, 5 and 10. We will use these as a reference for further topic exploration. The remaining topics can be found in the Appendix A 7.4. Our topics are shown in Table 5.1, 5.2, 5.3. The topics should provide a clear view of the type of errors in our dataset.

Topic	Terms				
0	user	information	retrieving	ta	select
1	failed	code	request	restclient	ping

Table 5.1: Topic 1..2 with top 5 terms

Examining Table 5.1, we cannot simply deduce what these two topics are about. We can at best say that the second topic is probably an error topic based on the simple word failed. The topic count is fairly low and earlier exploration expects us to have more latent topics in the dataset. The extracted topics are probably the most common occurring terms. This is actually true if we compare these topics with Fig 4.1a.

Topic	Terms				
0	ta	select	f	group	plan
1	fluid_dashboard	text	tblasset_ser	tbljob_sow	group
2	v	r	order	null	tb
3	failed	code	request	source	restclient
4	user	information	retrieving	log	party

Table 5.2: Topic 1..5 with top 5 terms

We continue on to our next table, Table 5.2. We see similar words in this table as Table 5.1, topic 3 and 4 have the same terms as topic 0 and 1 in the first table. It is probable that these terms are common occurrences throughout our corpus. Once again looking at our Fig 4.1a, we recognise more common terms. Although the

count of topics has increased our ability to deduce the topics has not been increased.

Topic	Terms				
0	request	migrate	restclient	url	ping
1	fluid_dashboard	text	tblasset_ser	tbljob_sow	coord_type
2	ordcateg	cancel	token	postgres	_bic_zpmrsord
3	orderitem	redo	record	unit_day	length
4	f	foo	count	ta	v
5	ops	hawqstatus	down_indic	set	_bic_zlongit
6	request	ping	migrate	url	http
7	failed	material	recordmode	notificatn	group
8	ta	select	group	plan	dispatch
9	user	information	retrieving	tblasset_ser	tbljob_sow

Table 5.3: Topic 1..10 with top 5 terms

The final table is Table 5.3. Once again the topics are hard to read, although some topics have become more clear. Topic 2 and 3 are probably about database changes to orders. The terms ”failed” and ”user” are once again shown, except ”failed” is now grouped with a very different set of terms. Some terms appear more times, e.g. ”ta” and ”tbljob_sow”, ”url”. It shows us that more latent topics are hidden within our dataset, which the low count of 2 and 5 topics cannot show. However due to the ambiguity of the terms we cannot tell if we need more topics to generalise or reduce the amount to make topics more specific.

The remaining topics shown. Trying to infer topics through the top terms has so far left a undesirable result. Servers logs which are created for domain specific programs and systems make it harder to interpret a coherent and distinctive topic. Noticeably in higher topic counts the amount of similar top terms, e.g. term ’ta’ in the model with 11 topics appears 3 times as top term in 3 separate topics. This might simply be a popular word for multiple documents or the model is not sufficient. Lower topic counts have topics which are clearly not representing the smaller and important infrequent occurring server log messages.

The results leave us with the desire to better interpret the topics and gave us a clear view that interpreting is not an easy task.

5.4 Comparison of the inferred topics through pyLDAvis

In this section we will discuss the inferred mapping of pyLDAvis. The visual representation of 5 topics in Fig 5.2 and of 10 topics in Fig 5.3 will be used to compare other models. The pyLDAvis mapping includes the distance metric Jensen Shannon and as such computes distance of each topic to each other, as mentioned before at Section 4.5.5. The package allows closer inspection of topics through term relevance. It does not show the reality of the documents being clustered, only an estimate of the topic size compared to other topics based on the term frequency. The remaining topics can be found at Appendix A 7.2.

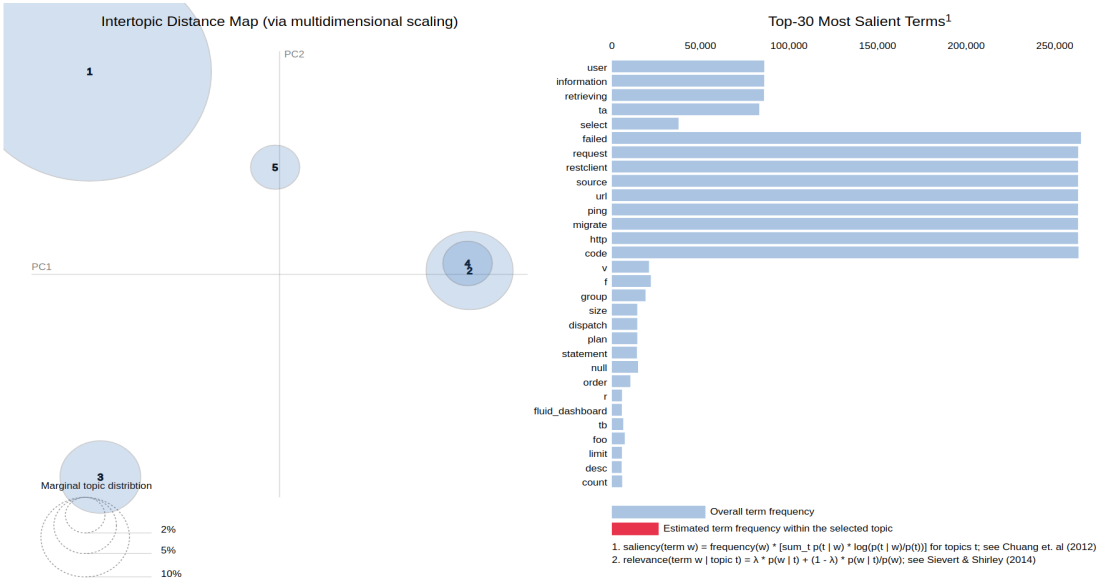


Figure 5.2: pyLDAvis topic visualisation with 5 topics

We first inspect Fig 5.2. Interestingly the figure implies that there is already an overlap of topics with as few as 5 topics. Topics 2 and 4 are overlapping, this could simply be a result of the dimensional reduction applied by pyLDAvis, meaning topics have overlapping terms. The remaining topics have a clear distance from each other. Although topic 1 is very large, it is expected though because the dataset contained a lot of similar terms.

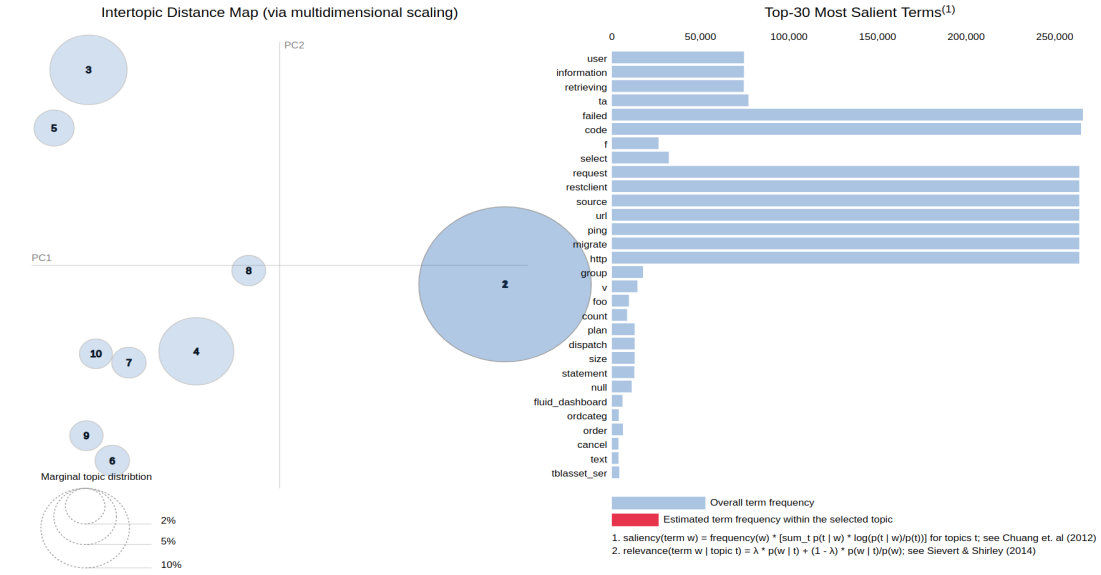


Figure 5.3: pyLDAvis topic visualisation with 10 topics

In Fig 5.3 we jump to 10 topics. Closer inspection of the pyLDAvis shows overlap in topic 1 and 2. The remaining topics are reasonably well distinguished and show no overlap.

Further examining the remaining figures we like to state the following:

1. The remaining figures show no clear preference of topic count

2. The majority of figures appear to have overlapping topics

The first statement is probably because pyLDAvis is created to help show a global topic distinctiveness but also help the user interpret the topics relevant term. This means it is not necessarily built for different topic count comparison. The second statement is once again due to the nature of LDA which has overlapping terms in different topics, like the topics in 1 and 2 in the 10 topics model.

5.5 Coherence

The next results show a topic coherence score in correlation to topic count. In comparison to earlier results, we use a calculated score rather than our own judgement. The coherence model is based on the coherence measure Cv, which measure human interpretability. Hopefully the results make show a topic count with a clear Our Fig 5.4 shows the average value each model scored.

Figure 5.4: Coherence values based on the measure Cv

The results in the figure are interesting. As we already discussed in earlier sections, we did not find a clear topic count to be better based on our visual interpretation. The figure shows that topic count 11 has the highest score with 23 being the second highest, in contrast topic count 35 has the lowest. When we look at topic count 11 in Table 7.11, we see multiple similar top terms. The topic count 11 also shows a lot of overlapping topics in pyLDAvis. The figure also shows a clear fluctuating score from low to high topic count. The average score is 0.46 and 13 of the total 25 topics are above average. Solely based on this figure we could say that 11 has to be the best topic count, however our previous evaluation measures do not clearly agree with this so far.

5.6 Document distribution based on hard clustering

This is the first section where we compare held out data to the train and test data. The models transformed the data into document and topic distributions. Our Fig 5.5 and Fig 5.6 represent the count of documents labelled with their highest probable topic count. The figures show to topic count 2 till 11. The remaining document distributions are found in append A 7.3.

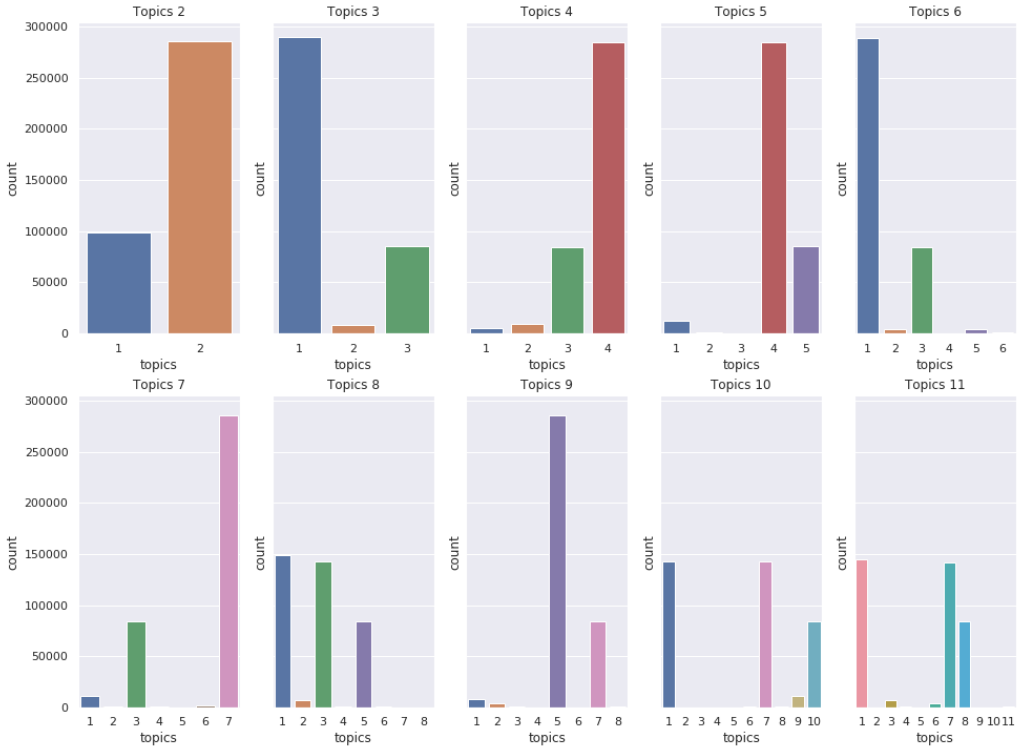


Figure 5.5: Document distribution on the train test data

The results that are shown in Fig 5.5 are actually very agreeable with what we have seen so far. The distributions based on the topics in pyLDAvis as well the term frequency match up. What we could not see clearly before was how the document would be distributed based on clustering in higher topic counts. Most documents are highly related to at least 2 topics, otherwise 3 topics. The remaining topics are either empty or so small in comparison that our figure cannot show them clearly. If we take a look at the results in the appendix A 7.3, which contain our higher topic counts we can see a high variety of distributions. One thing is clear that the increase of topic counts, increases the smoothness of the distribution of documents. A few exceptions have either a singular or duo of topics which contain noticeably more documents. Our results show that increasing the topic count while the corpus does not contain so much latent topic decreases the distinctiveness quality of each topic and as such flattens the distribution of documents.

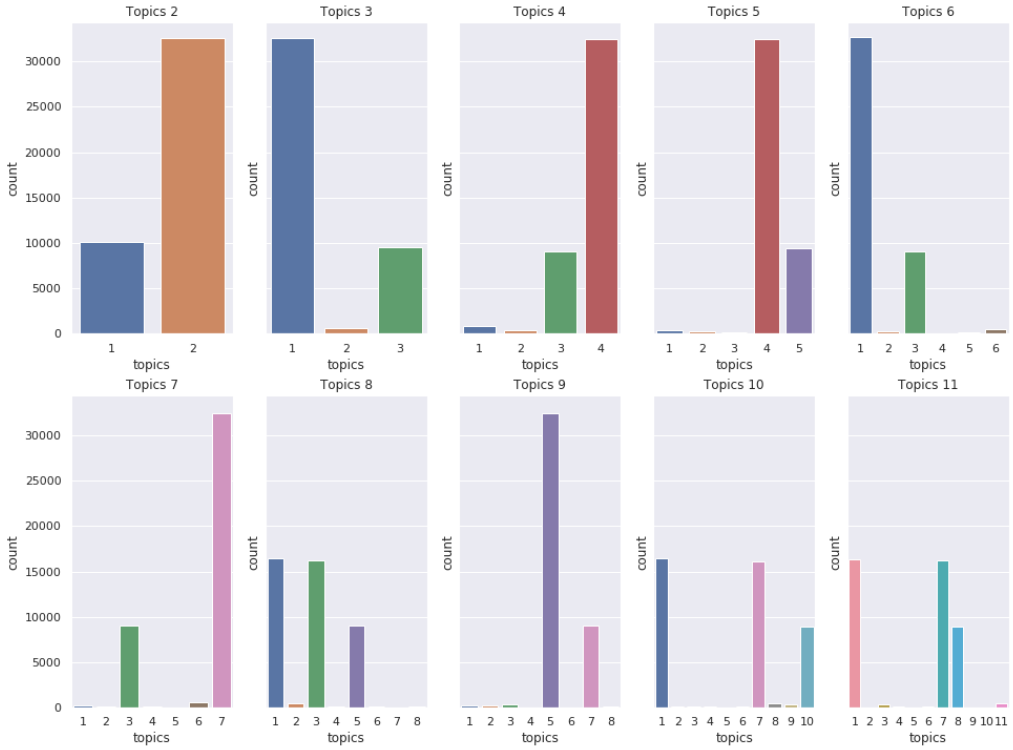


Figure 5.6: Documents distribution on the held out data

The same can be repeated for Fig 5.6. Although the held out data is created after shuffling the data and then splitting it, the distribution of documents in each model is remarkably similar. The held out dataset appears to be very similar to the train and test data based on the clustering. Which also brings us to the same deduction as before. As we expected, the results in the appendix appear similar.

In this first comparison between held out and our trained models, we see that results are quite similar. The results have shown that increasing the topic count leads to more evenly distributed documents. The model shows the same output on held out and the train and test data. This is great news as such, this model might be able to indicate errors similar to the topic count of choice.

5.7 Silhouette values

The final evaluation metric is the silhouette, which will use the cosine metric. The complexity and memory requirements of silhouette leaves us with a sampled set to be evaluated. Our held out and train and test data are sampled on 10000 documents and hard clustered based on their highest probable topic. It appears to be impossible to calculate the silhouette after 12 topics. The mathematical explanation is that the denominator has a value of zero, as such it can not calculate the silhouette of a sample. This means that topics have become

so small and overlapping that the average distance for each sample in a cluster and the nearest cluster has become 0. Silhouette gives a clear score that can only be between -1 and +1, the higher the silhouette score the better our models.

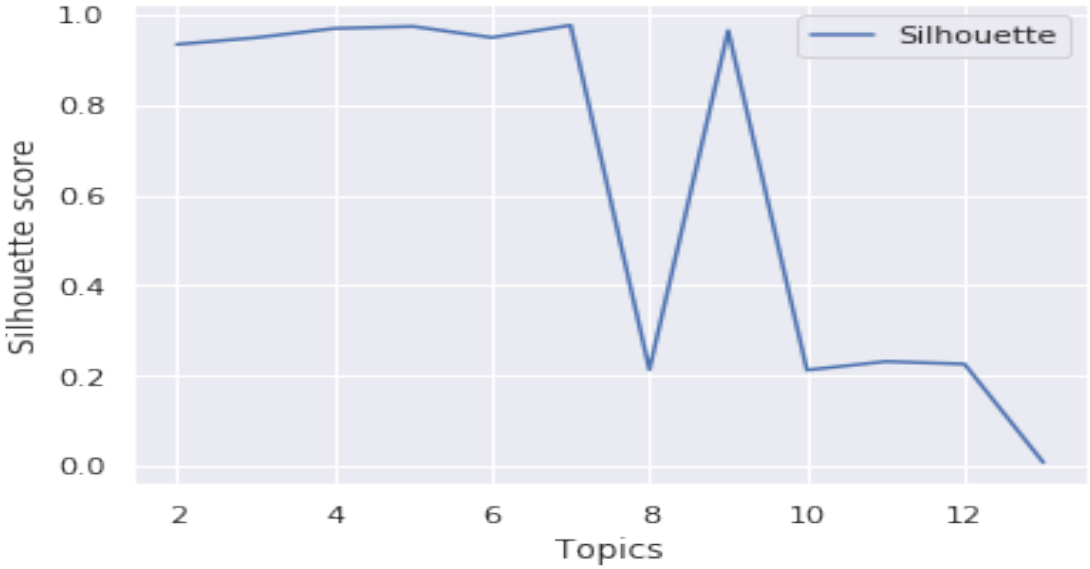


Figure 5.7: Silhouette values based on the train test data

The results shown in our Fig 5.7 show a higher score of silhouette on the lower end, being constantly around 0.9 to 1.0. The only topics that show lower values are topic 8 and all topics after topic count 9. The results do show a preference for lower topic counts, increasing the topic count after 7 topics decreases the silhouette score in general.

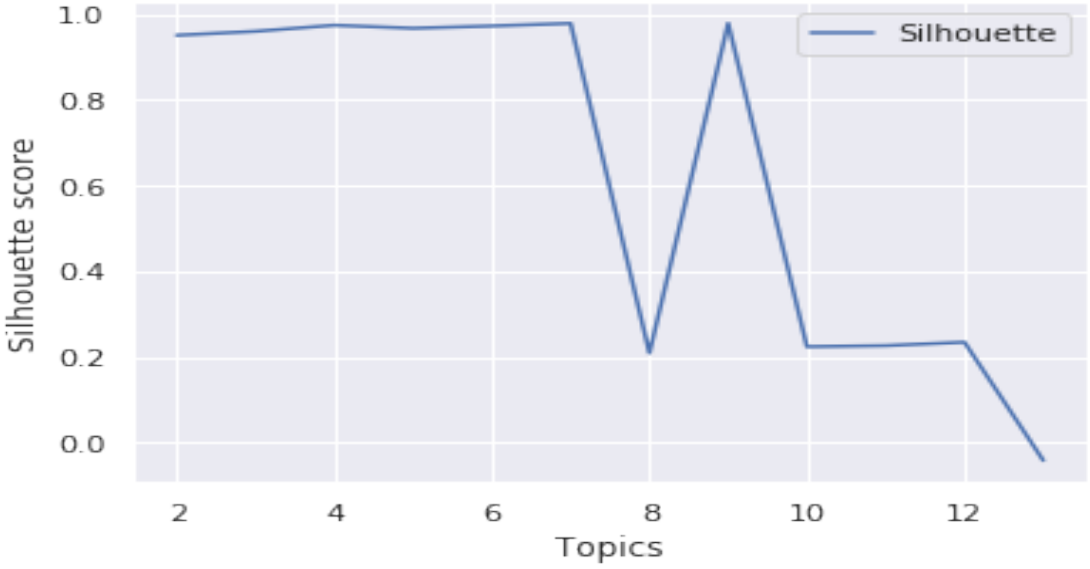


Figure 5.8: Silhouette score based on the held out data

Comparing the Fig 5.8 with the earlier Fig 5.7 shows an expectable similar result. In the previous section we already deduced the held out dataset to be very similar to our train and test dataset. The models show

great promise based on the score of the silhouette, although it is noticeable that even the lowest topic count of 2 has a high silhouette score. The silhouette score can be said to evaluate the models based on their created document clusters, which appear to be of great quality.

5.8 Contribution

In this section we will discuss the specific findings from our research. These findings include the data extraction and exploration of server logs. Applying the online LDA model on the dataset and the general difficulty of evaluating the models.

The data extraction has been interesting, having access to such a huge amount of data for exploration is a treat for every researcher. In the past research has been performed on extracting server logs for analyses, but not a lot of research is performed with access to a great quantity of data as has been used in this thesis. Furthermore, our data can be said to be unique in the aspect of having typical labels, which identify the type of server logs.

The identifying process of the dataset has been a main focus. As such we found the LDA machine learning model suitable to provide our dataset with the necessary structure. Topic modelling on server logs for added value has not been done with much success or without much effort in the past. The nature of LDA, makes it hard to evaluate. LDA can be evaluated using metrics which show to have a good performance for extracting latent topics however this can easily contradict human interpretation. The nature of this dataset makes it hard for humans to understand it without domain knowledge, even domain experts might not be enough. However, LDA depends on the readability of its users, entailing us to find the best combination or spot between semantic analysis and human dependability.

The results shown in our Chapter 5, show interesting results and even contradicting results. The latent topics inferred were evaluated on distinctiveness and coherence. We took a brief look in Section 5.3 of our topics with their terms and experienced the difficulty hands on of interpreting these topics and choosing the optimal number of topics. Following we use pyLDAvis, which is widely used for topic distinctiveness interpretation and easy interpretation of topics itself. Furthermore, we evaluated the coherence of topics with coherence metric Cv. Finally, we chose to hard cluster the documents in their most probable topics using the silhouette coefficient to see that similar documents are indeed clustered together for certain amount of topics, corresponding with our held out dataset for final confirmation of the quality of our models.

We will discuss the final conclusion drawn from our experiments and answer the general research question in the conclusion next chapter.

Chapter 6

Conclusions

This chapter contains the conclusion in which we will shortly explain the research and derive a conclusion from our experiments. We will answer the research questions and end the chapter with a section discussion and future work.

6.1 Conclusion

This research aimed to apply topic modelling for clustering and discovering an optimal model with the help of server data of Capgemini. Capgemini provided data of their server data warehouse for this research. We extracted a subset of data contained in the server logs filtered on the term 'error'. Based on earlier research we use the unsupervised machine learning model Latent Dirichlet allocation (LDA) to discover latent topics in this data set. As such we recognise our dataset as a corpus and the server logs as the documents in this corpus. To achieve this transformation of data to corpus we used some steps. We preprocessed the data using the standard steps: normalisation, stop word removal, tokenization and creating a bag of words matrix (bow). The bow was divided in a train, test and held out set. Furthermore, we trained multiple models with different amounts of topics 2-38 with our train and test set. Lastly we evaluate the results of each model using multiple metrics. We evaluate distinctiveness, coherence of each topic using pyLDAvis and the coherence metric Cv. We further use human perception to evaluate the topics with their top terms. The documents are clustered based on their highest probable topic and as such we compare the document distribution based on hard clustering. Finally we use the silhouette coefficient to compare the multiple topics.

We will answer each research question and end with answering our main research question based on the results and the best of our knowledge.

How does the topic count influence the topic models?

During this thesis we applied many steps to our data. We chose multiple topic counts using the standard preset the gensim package offered us. For this research we created 25 models with the help of 380000 documents to train and test and the remaining documents to validate the models. Each evaluation measure shows a different side of our models. If we go back to our goal of discovering latent patterns and clustering documents we can explain our results. Our first observation of the pyLDAvis with a topic overview make it clear that the topics are hard to deduce. With pyLDAvis helping us to look fo extra, we see that increasing topics create more overlap of terms. The coherence that we see in Fig 5.4 shows no clear winner, however once again increasing the topic count too much makes the quality even more unpredictable. The contrasting results make it hard to judge these models on quality. Furthermore, when comparing our clustered documents in Fig 5.5 and the remaining distribution we can see that the topics generally stay around 2-3 large clusters, interestingly increasing the topic count flattens the distribution of documents, with at least 1 topic mostly having the largest cluster. The silhouette clearly indicates that higher topic counts have simply to much overlap which makes it not calculable, however lower topic counts show high silhouette scores. The results leave no conclusive decision, but we can say that the higher the topic count the less the quality of the overall clustering. The contrasting results of coherence and pyLDAvis makes it not clear whether interpretability increases with higher topic count for the topics and leaves us inconclusive.

Why are the chosen topic models suitable?

In the earlier part of this thesis we analysed multiple researches each with their own implementation of LDA. Our dataset is based on the optimal parameters Blei researched. While we do not have a streaming corpus, the online implementation of LDA that we applied in our thesis allows the models to be updated with new documents. This allows our current models to learn new terms and be able to cluster unseen documents better. This makes our models suitable for future recognition of error logs.

What are our findings when applying topic modelling and optimising the model on our data?

Evaluation is really hard on topic modelling. Especially when the data is so domain specific. Server data is not the same as natural language and using topic models will not result in clear topics from each model. Our human minds can see some connection between current logs, but we clearly do not possess the skills to interpret the topics objectively. It is simply to hard for a human to deduce the topics with our current dataset and models, which is why a unsupervised machine learning technique as LDA is really optimal to recognise the correlations using semantic analysis.

Leaving us with the research question.

Can we use topic modelling to classify and cluster error messages?

When all is said and done. Topic modelling can be used to analyse great deals of documents, discovering latent topics and clustering new unseen data in a similar way. However, the application of topic modelling on a dataset which is not even human interpretable makes the results afterwards hard to evaluate. If our application were to be to recognise error messages, this model would be up to the task. The model is even able to be updated. The clusters would be based on the scores of silhouette very good coherent and distinct from other clusters. Which once again begs the question, can we recognise the topic we put our new document under? It is not very useful to cluster documents together without understanding the topic this document falls under. The interpretability of topics leaves much to be desired. With the measures we used to evaluate our models, we would not recommend using LDA for classifying and clustering error messages. Topic modelling is better of being used on normal human generated corpora.

6.2 Discussion

6.2.1 Methodology considerations

This research took only a small possible path in the finite amount of paths. This section will further explain this path, which led this research to its logical conclusion and a honourable mention to not forget the time spent.

6.2.2 The reason leading up to LDA

Although the discussed research has been mainly focused around topic modelling (LDA) on finding similar error logs, the original research question started with a related but different subject. During this section I would like to discuss the original focus of predictive maintenance, as a lot of time has also been put in researching this difficult and challenging task.

Predictive maintenance

With the combined application of machine learning and big data, companies try to anticipate when machine hardware failure are due to occur. Predicting instead of reacting to problems saves time and money and allows for a better customer experience which can be found in the before mentioned research of Intel [SFMW14] [Aja13]. It is not hard to imagine why companies like Intel or Google have already been researching the possibility of big data for this problem. Which brings us to the data acquired for this thesis, originally intended by Capgemini for predictive maintenance.

Vrops and syslogs

The data had not only been system logs but also consisted of vrops logs. In a few words, vrops are unstructured logs created by the virtualisation tools of VMware. The vrops data contained health and performance statistics of their multiple servers. Extracting the values from the vrops databases with the Hadoop framework had great promise to lead to the desired research data necessary for predictive maintenance. Furthermore the syslogs contained an indication of the type of log. Early examination of these logs, brought messages from all kinds of type of servers. The following step was to understand and extract the data to use in our research.

Unlabelled and unstructured

Unluckily we soon enough found the syslogs to be lacking their log type. This seemed to be a mistake made by their developers when implementing their streaming pipeline of server logs, which made the data set of the last three years unclear of their type. This in turn made it impossible to use the vrops values, which could be understood combined with the syslogs, to correctly to know when system problems would occur, without having a consulting a domain expert all the time. The only logical step was to find a more mathematical suitable way to label the unlabelled and unstructured data, bringing this research again to the literature study phase for a new solution.

Topic modelling

Going back and forth between different algorithms available, my supervisor finally hinted at Latent Dirichlet allocation (LDA). Lots of research has been done with LDA. Having already been wasting enough time on searching, this research had to settle for a technique. LDA seemed like an applicable algorithm to this problem based on earlier research. Text mining on logs has been done before, although barely on unlabelled logs. Training LDA on unlabelled data was common, expect that a lot of data contained either longer documents or a less domain specific dataset.

6.2.3 Recommendation

In further research I would not recommend LDA for domain specific log research and labelling, unless one has enough time and patience and expertise. If one is to attempt LDA for further research on logs, be sure to know which results are wished for. Evaluating LDA without a proper desired result might leave the evaluation process as a tedious phase.

6.3 Future work

This research leaves a lot of open areas to optimise or further research server logs in general with topic modelling. Experimenting on similar data can be achieved and as such a few points which come to mind when recommending future works are:

1. The way documents and corpora are declared. The corpus in our research was 1 day of data, filtered on the term 'error'. The documents could be an collection of specific server instead of 1 log being 1 document, this will increase the document size tremendously and should help LDA perform better.

2. The pipeline of data extraction. We filtered a lot of server specific features to extract our dataset, which can be a loss of information for LDA.

3. Data preprocessing. This is clearly always important when processing the data and can be experimented on in various ways.

4. Exhaustive search. Most parameters were based on best practices from earlier performed, especially by Blei from the original LDA paper. Hyperparameters and topic count can be changed and better set based on the distribution of your documents. This can be quite exhaustive and time consuming, but could be interesting. Otherwise making use of a super computer speeds up the calculations.

5. The quality of data. LDA is created to handle large amounts of documents, which we did have but not with the desired document length and variety. Using more varied logs, like informational logs etc could be show better latent topic inference.

Chapter 7

Appendix A

7.1 Capgemini server dataset

This section serves as a complete overview of the data. The server data that is provide by Capgemini has been developed by their own developers and as such contains a custom schema. Event logs contain multiple columns shown in Table 7.1 in an order fashion. Due to privacy reason, we will not show the data that is contained within the rows. Only two columns are worth discussing: syslog.body, syslog.severity. The former contained a lot of server specific data. This event log has been formatted to have a lot in common with the syslog message standard. The latter contains the type of message the event log, e.g. 0 indicating an emergency message. Due to a formatting error discovered during the research, the original syslog.severity has been lost and is always labelled 6 meaning informational.

filename	hostname	mime.type	path	syslog.body	syslog.facility
syslog.hostname	syslog.port	syslog.priority	syslog.protocol	syslog.sender	syslog.severity
syslog.timestamp	syslog.valid	syslog.version	timestamp	uuid	

Table 7.1: All the columns in the complete dataset

After manually inspecting the data we extracted only 4 columns named: hostname, uuid, syslog.body, timestamp. A custom solution had to be written to transform the servers logs into a use able matrix. These steps are explained in Section 4.2.

7.2 pyLDAvis

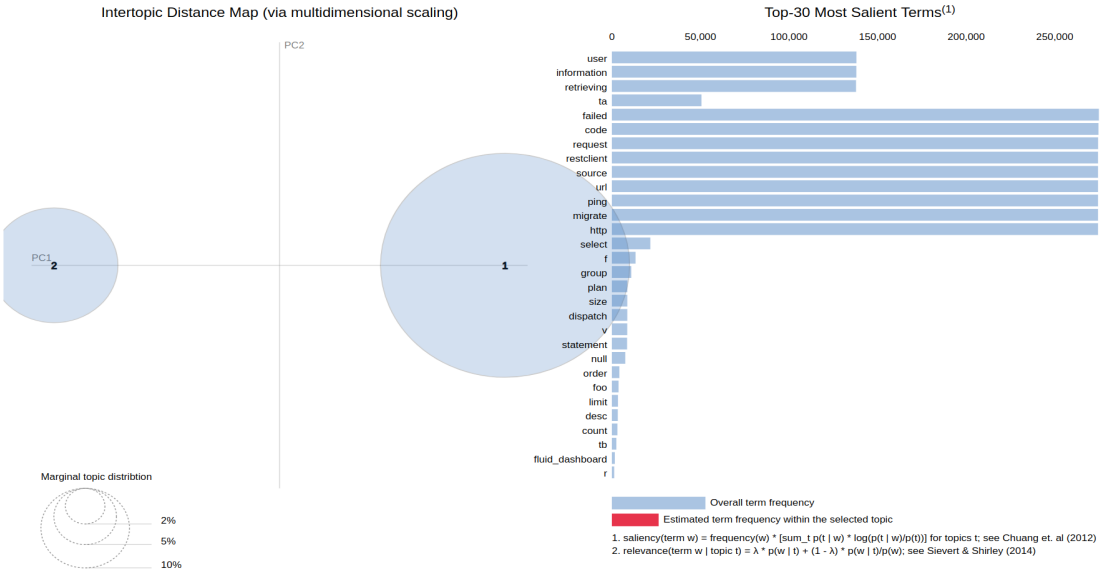


Figure 7.1: PyLdavis topic visualisation with 2 topics

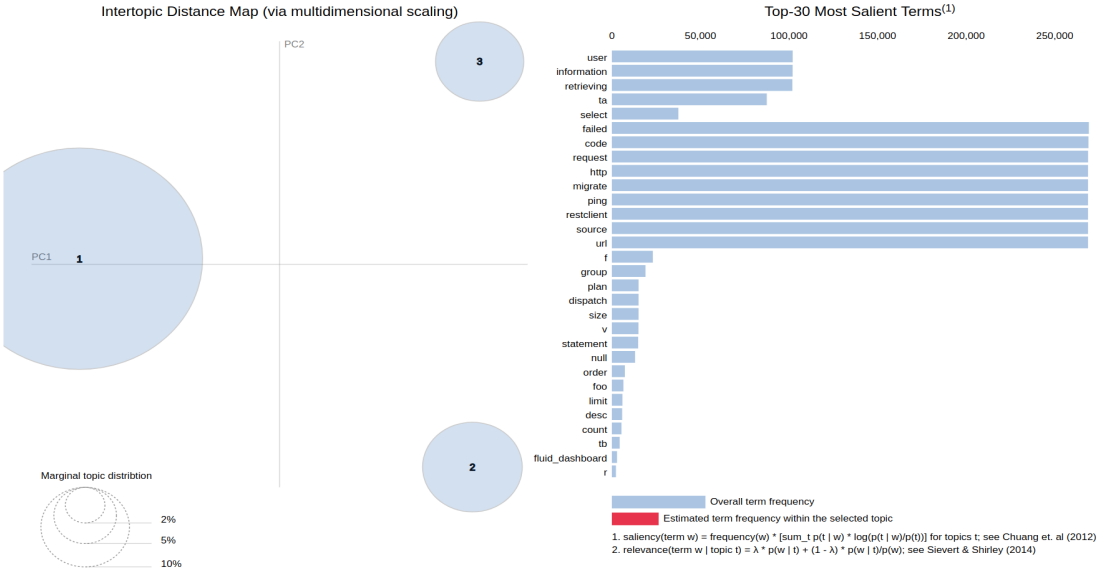


Figure 7.2: PyLdavis topic visualisation with 3 topics

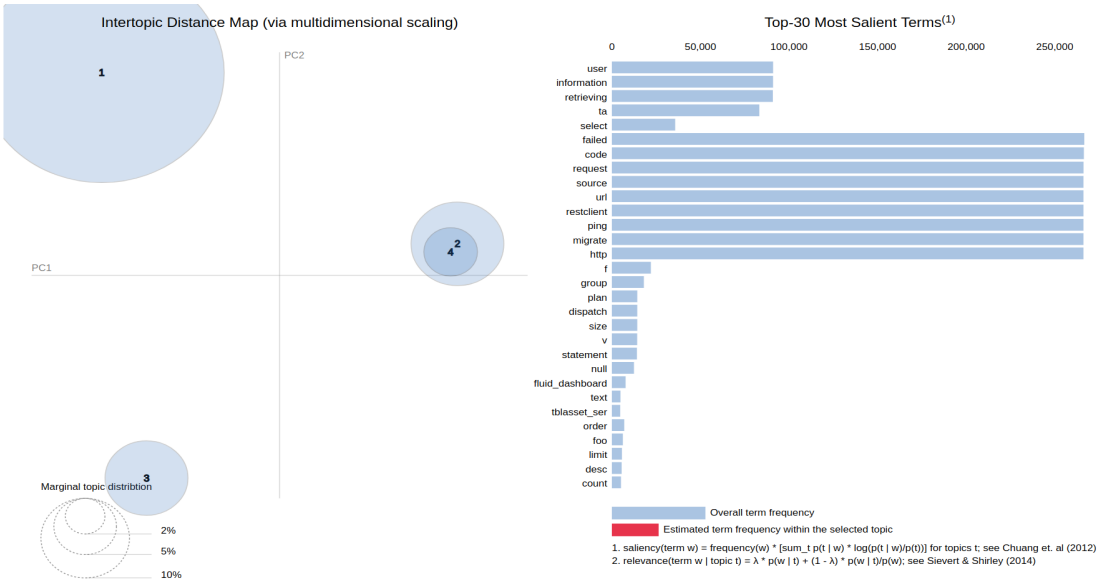


Figure 7.3: PyLdavis topic visualisation with 4 topics

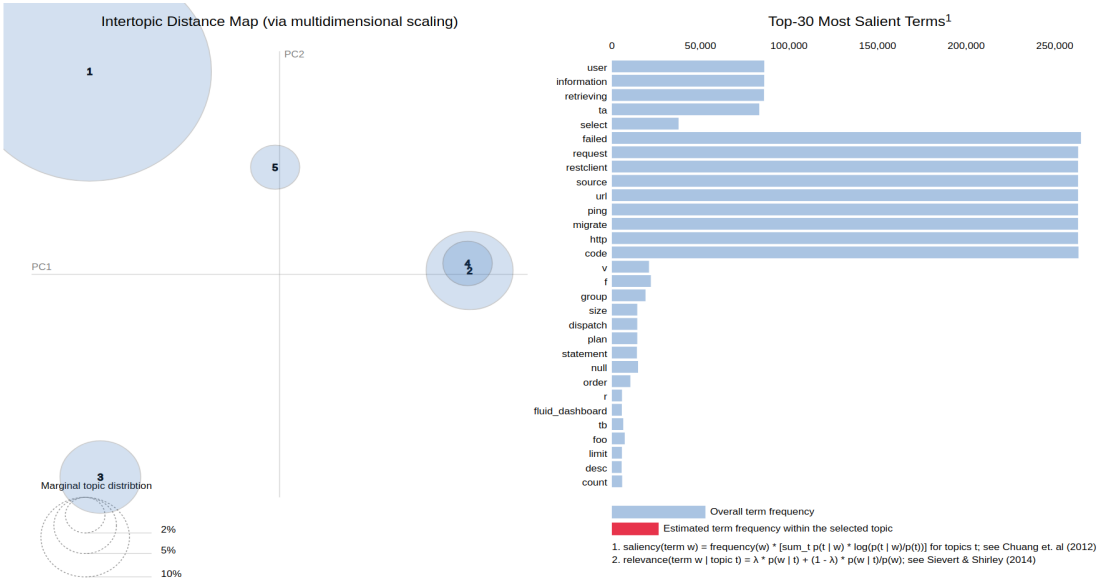


Figure 7.4: PyLdavis topic visualisation with 5 topics

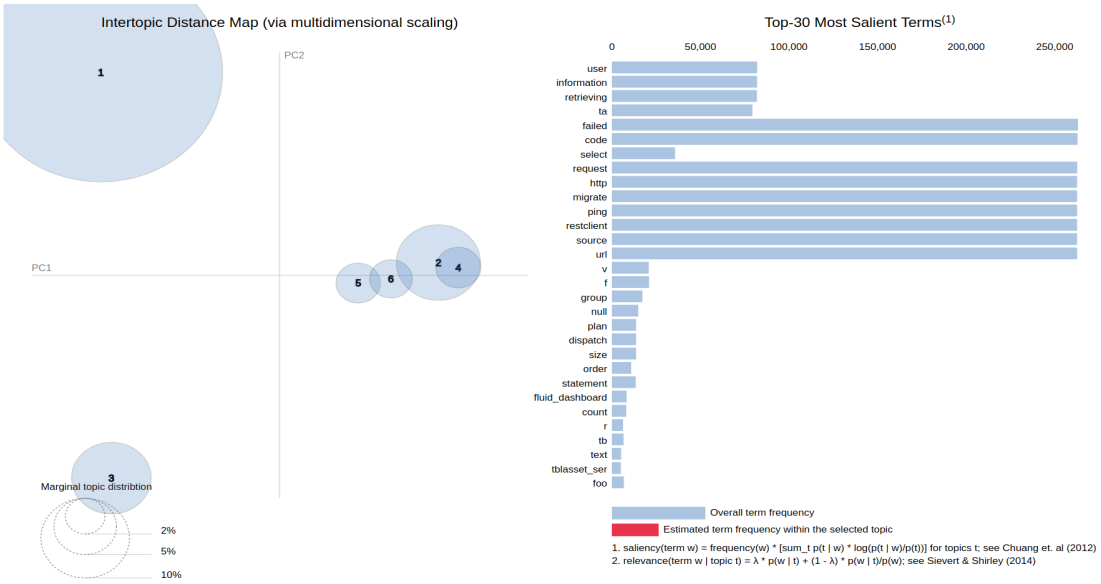


Figure 7.5: PyLdavis topic visualisation with 6 topics

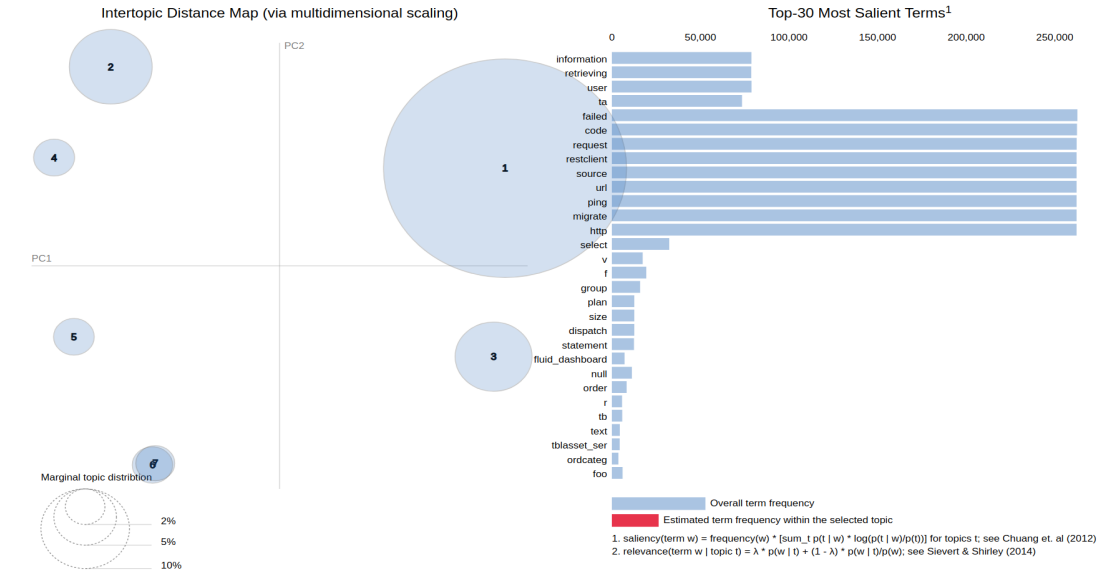


Figure 7.6: PyLdavis topic visualisation with 7 topics

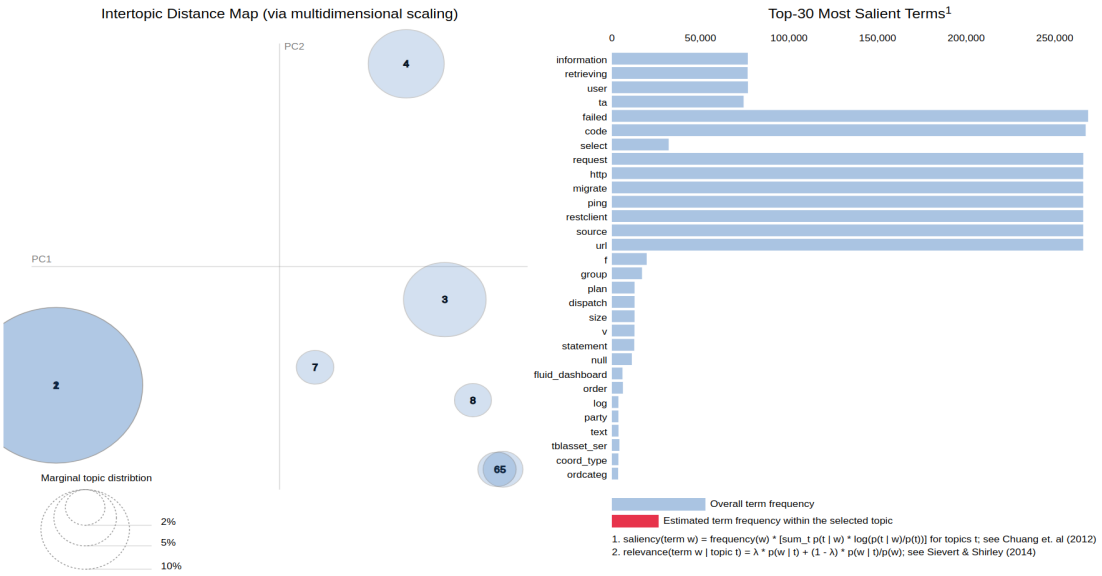


Figure 7.7: PyLdavis topic visualisation with 8 topics

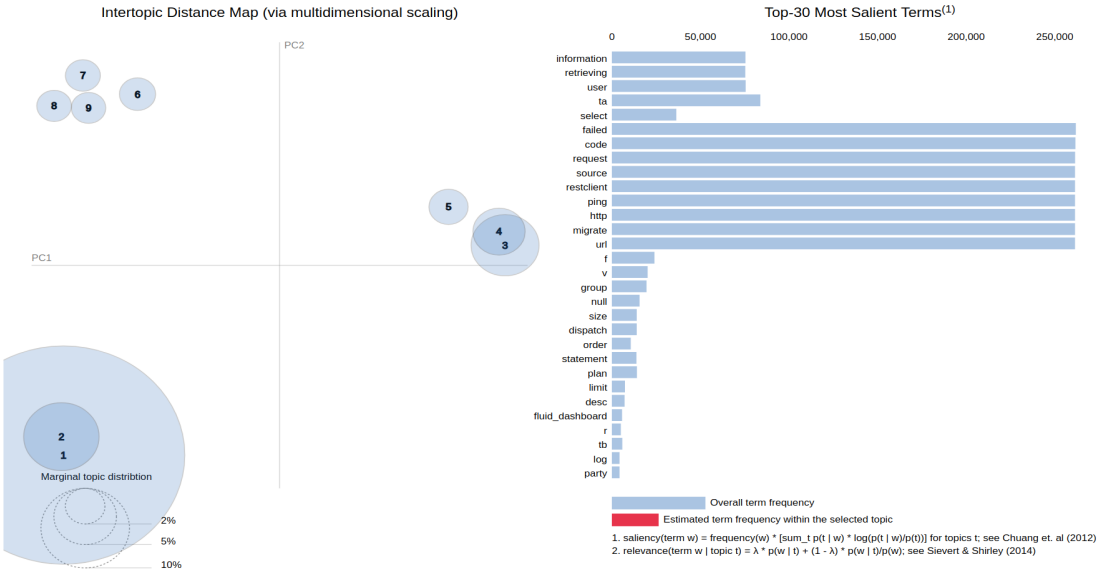


Figure 7.8: PyLdavis topic visualisation with 9 topics

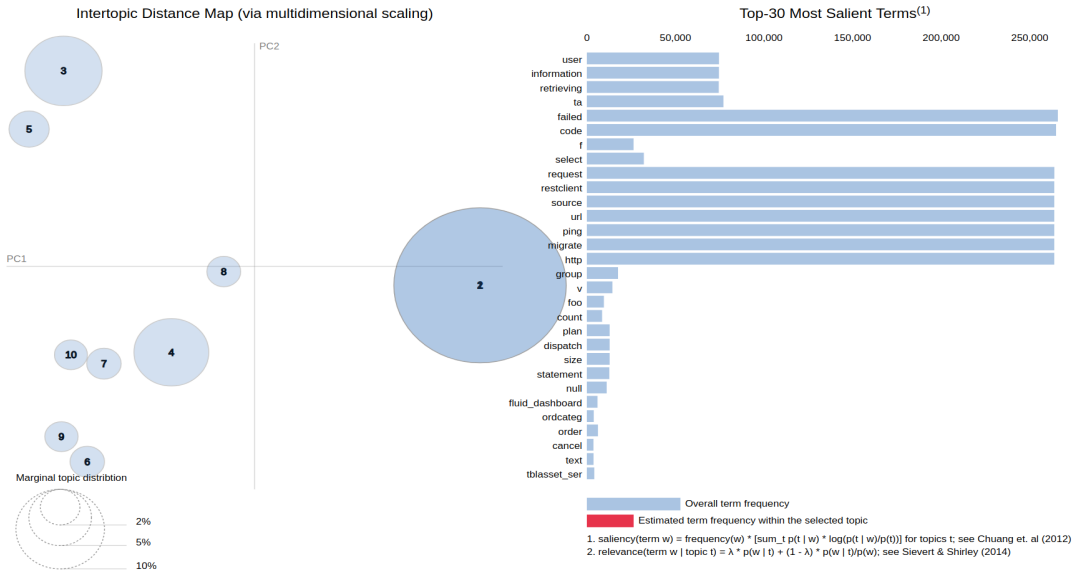


Figure 7.9: PyLdavis topic visualisation with 10 topics

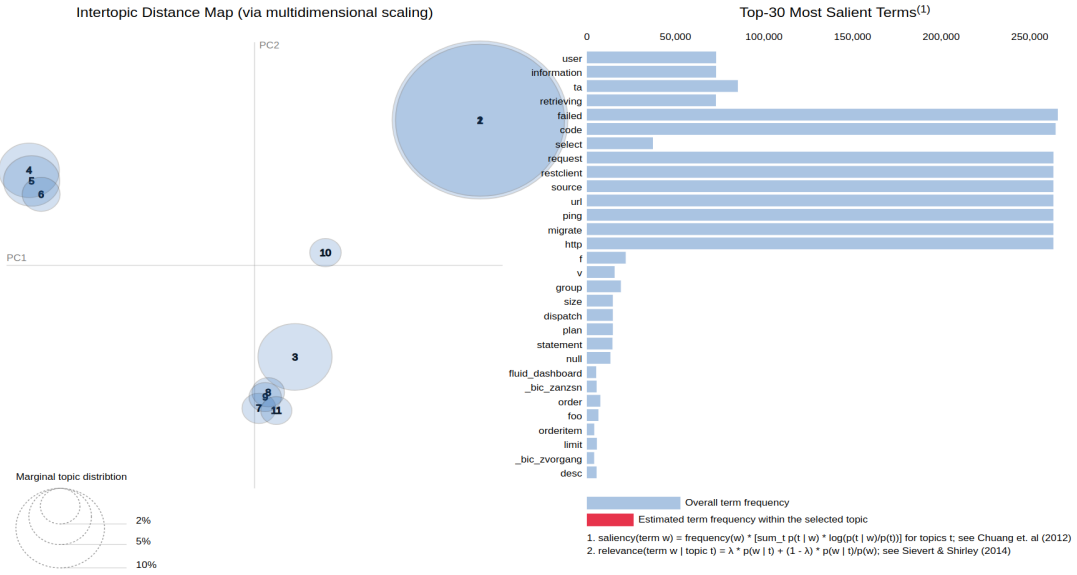


Figure 7.10: PyLdavis topic visualisation with 11 topics

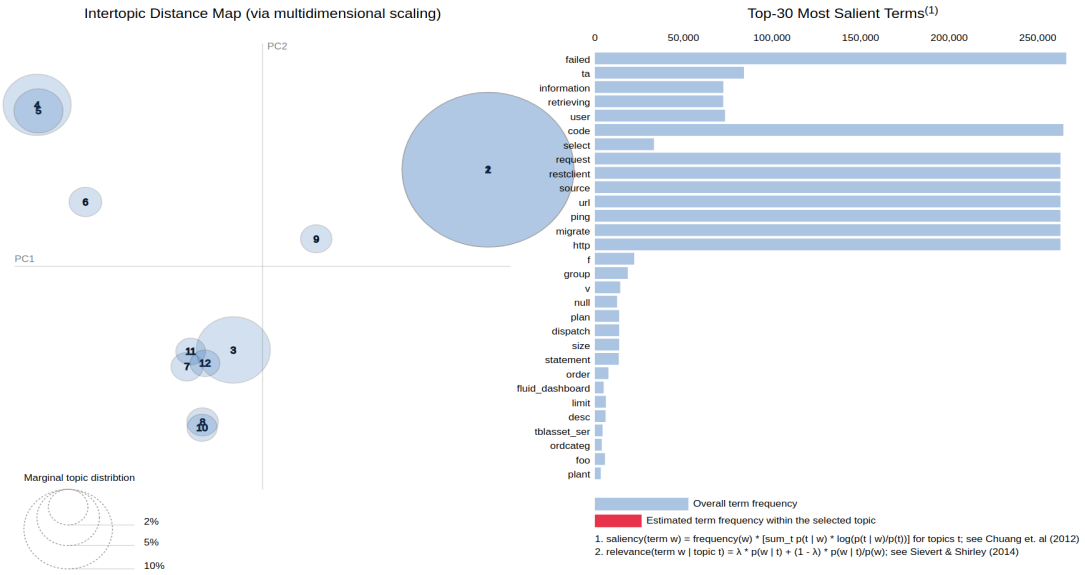


Figure 7.11: PyLdavis topic visualisation with 12 topics

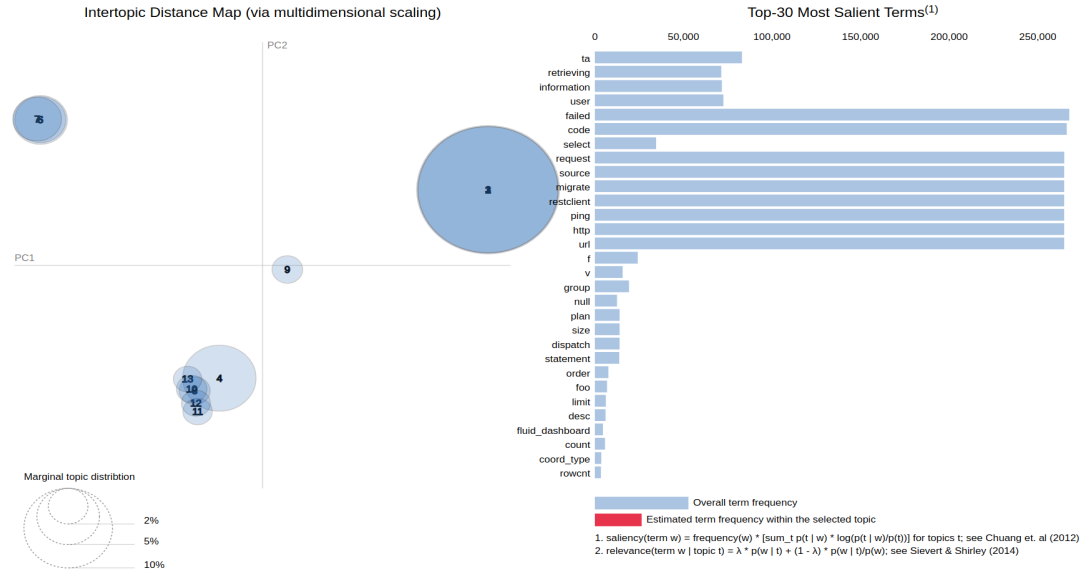


Figure 7.12: PyLdavis topic visualisation with 13 topics

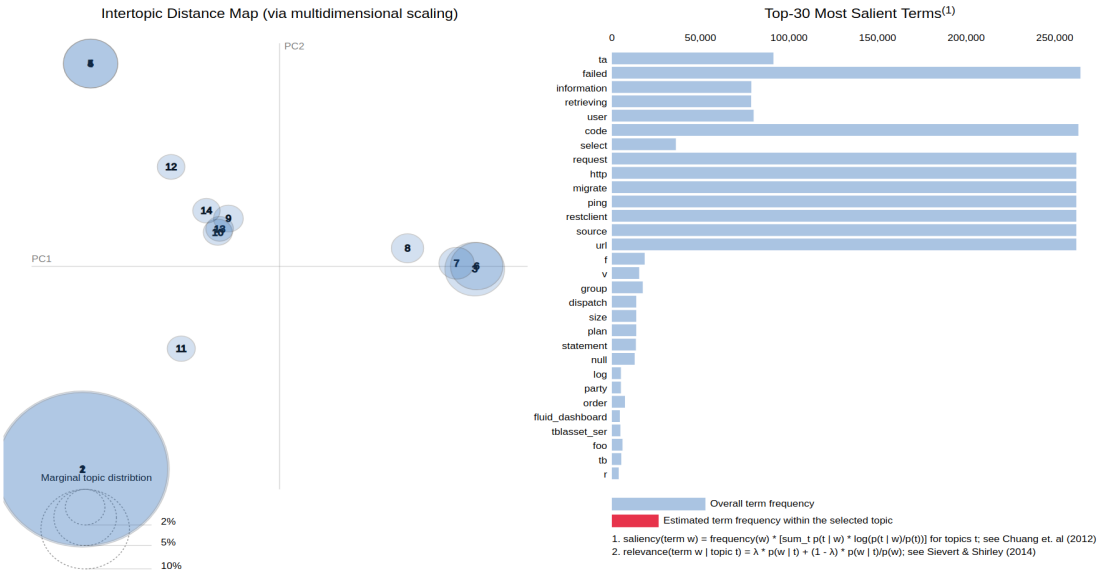


Figure 7.13: PyLdavis topic visualisation with 14 topics

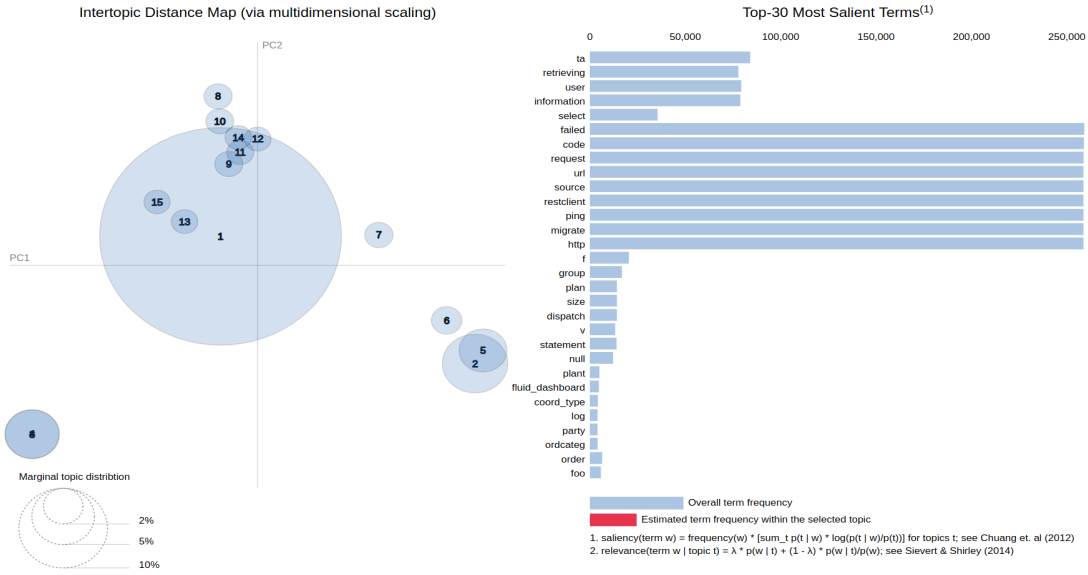


Figure 7.14: PyLdavis topic visualisation with 15 topics

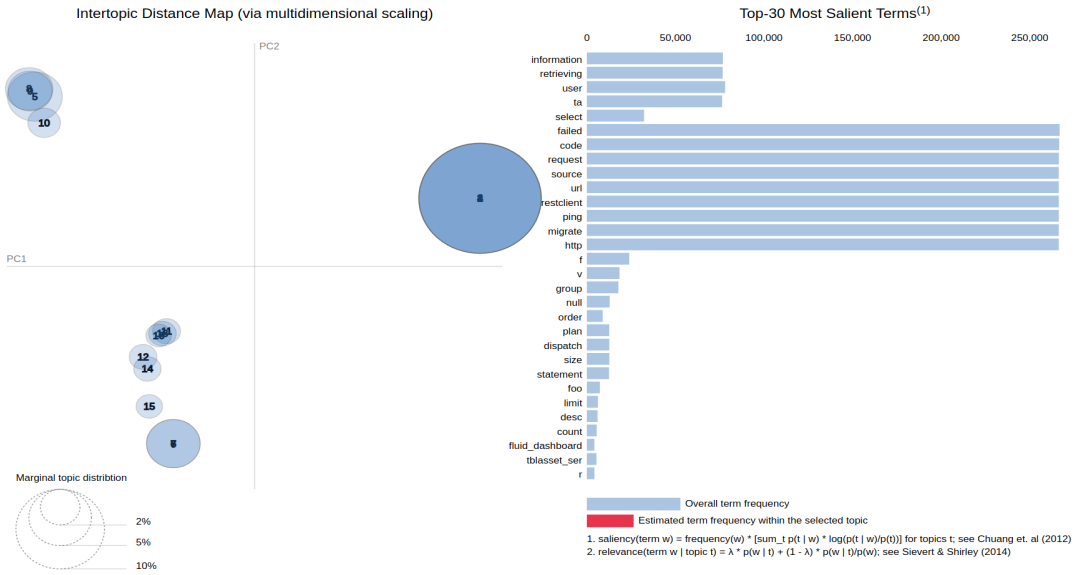


Figure 7.15: PyLdavis topic visualisation with 16 topics

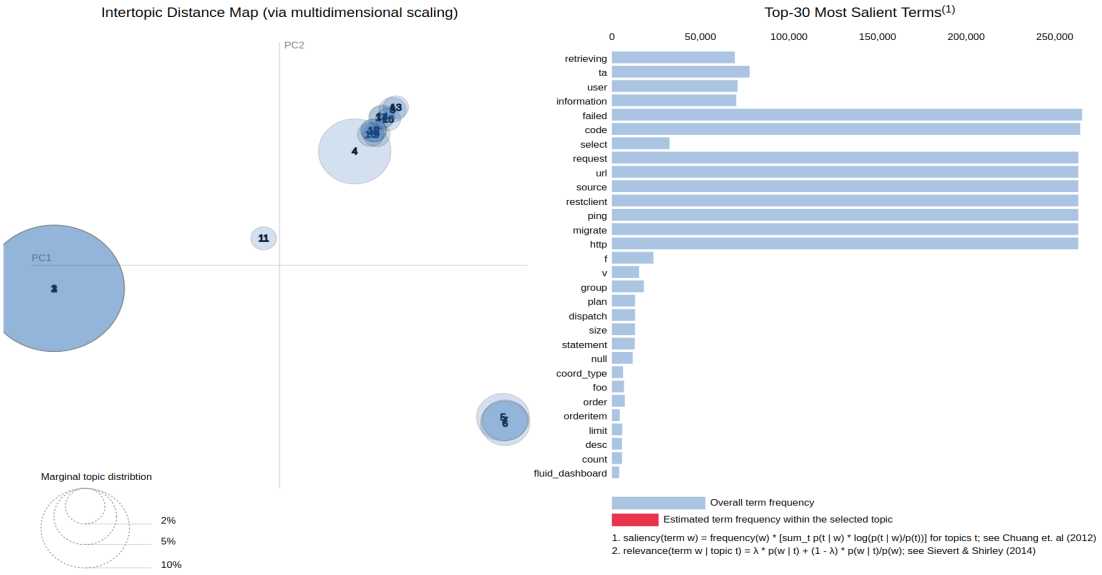


Figure 7.16: PyLdavis topic visualisation with 17 topics

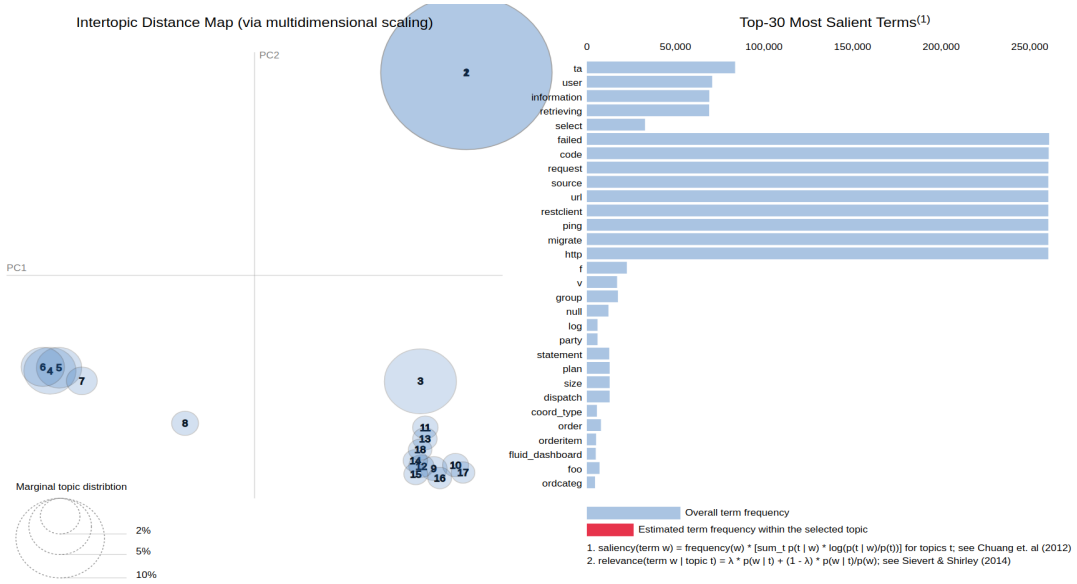


Figure 7.17: PyLdavis topic visualisation with 18 topics

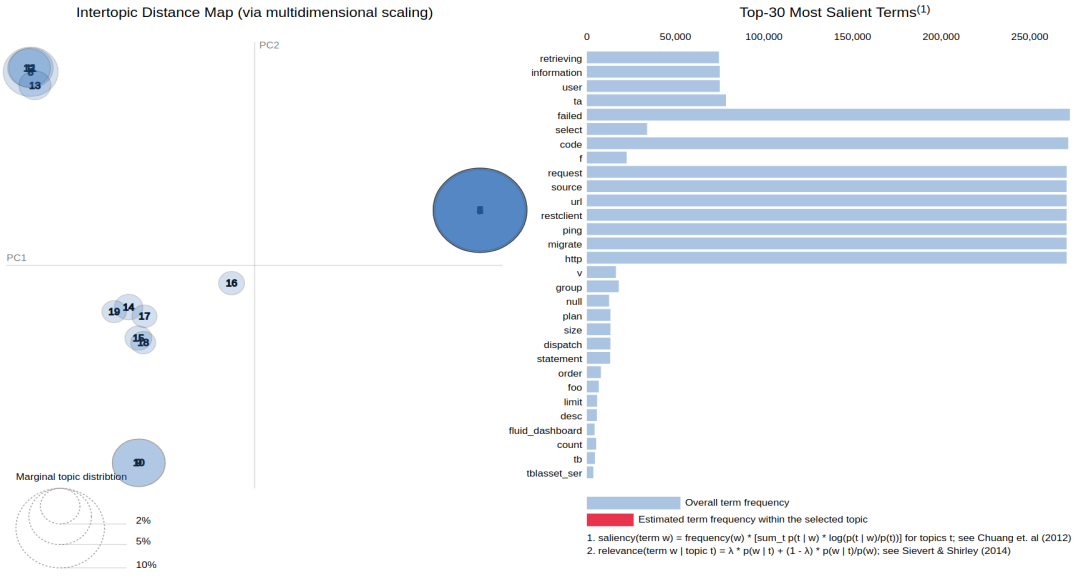


Figure 7.18: PyLdavis topic visualisation with 19 topics

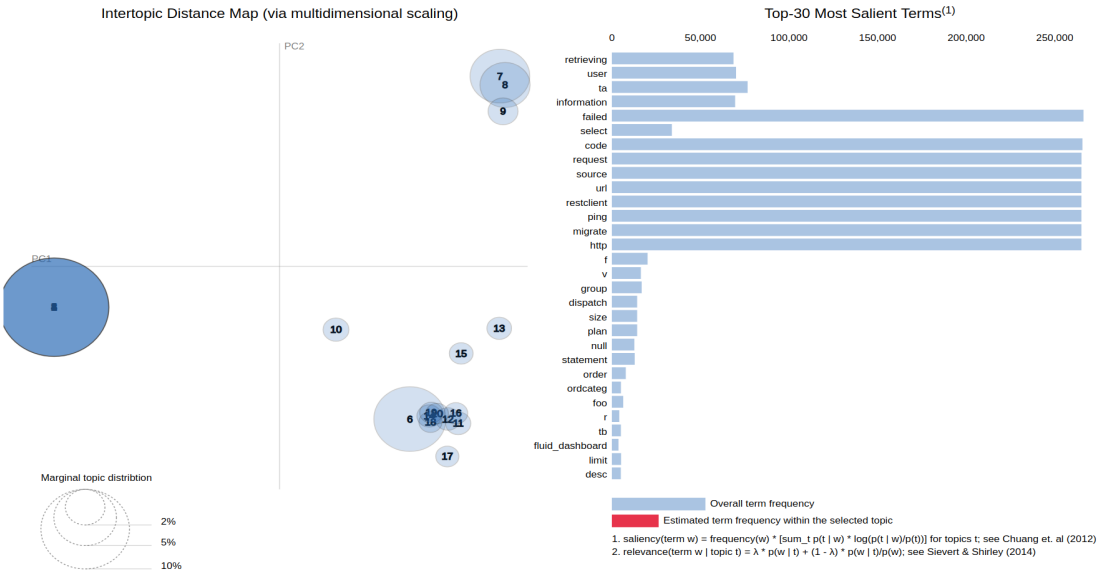


Figure 7.19: PyLdavis topic visualisation with 20 topics

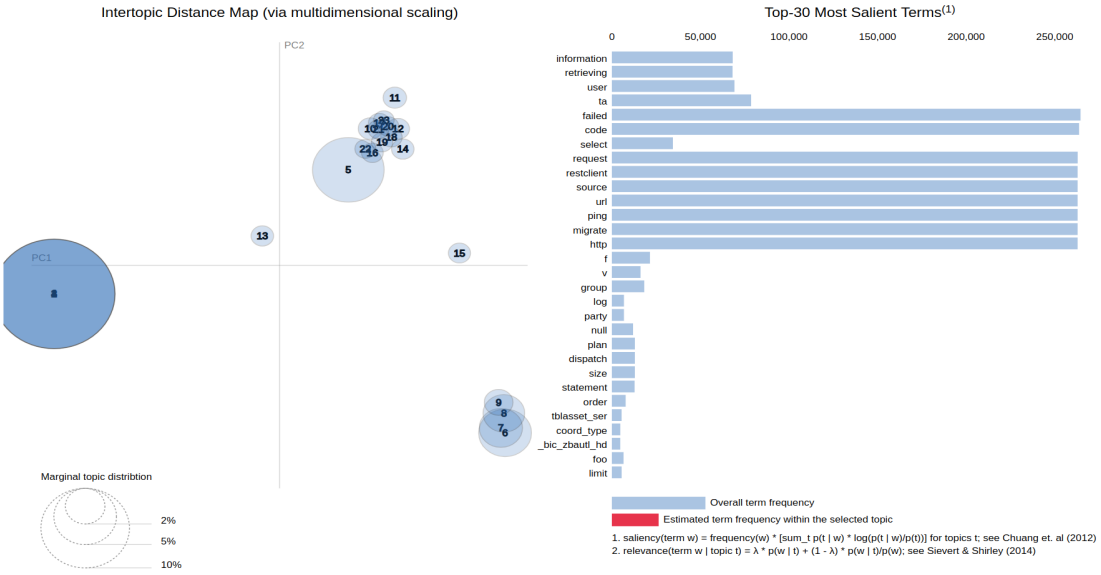


Figure 7.20: PyLdavis topic visualisation with 23 topics

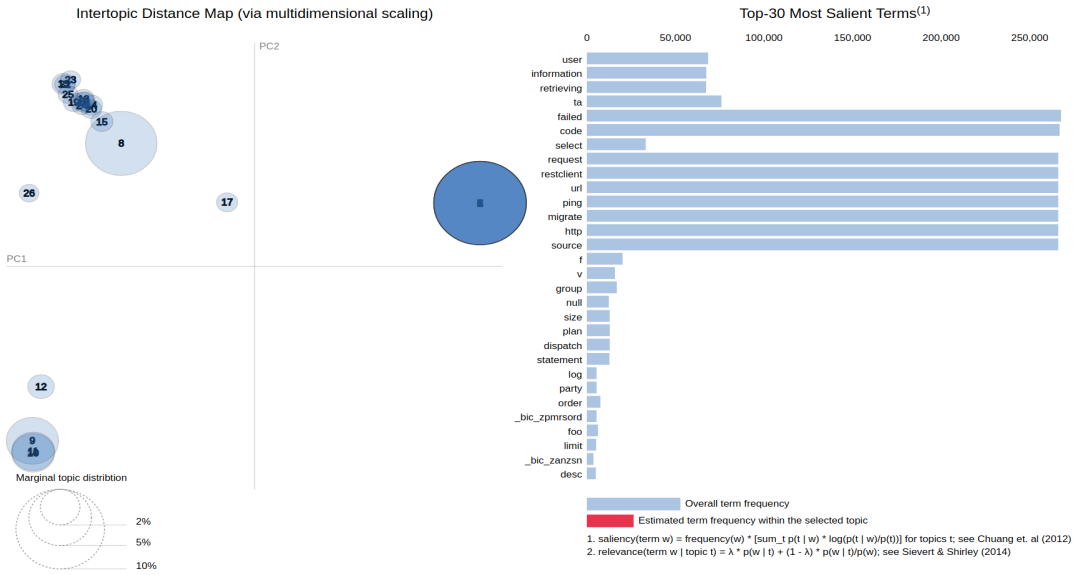


Figure 7.21: PyLdavis topic visualisation with 26 topics

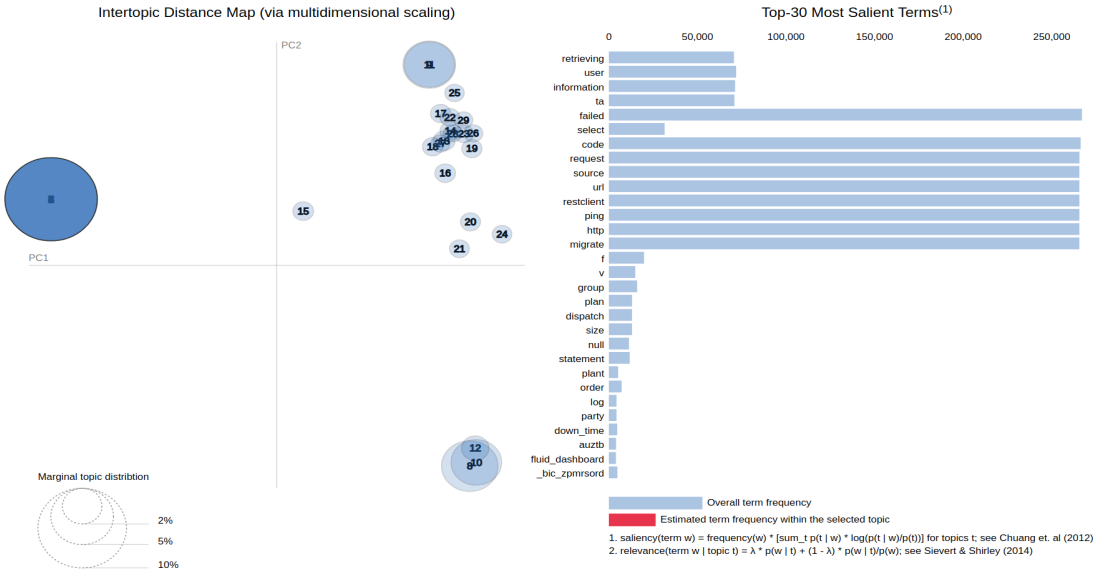


Figure 7.22: PyLdavis topic visualisation with 29 topics

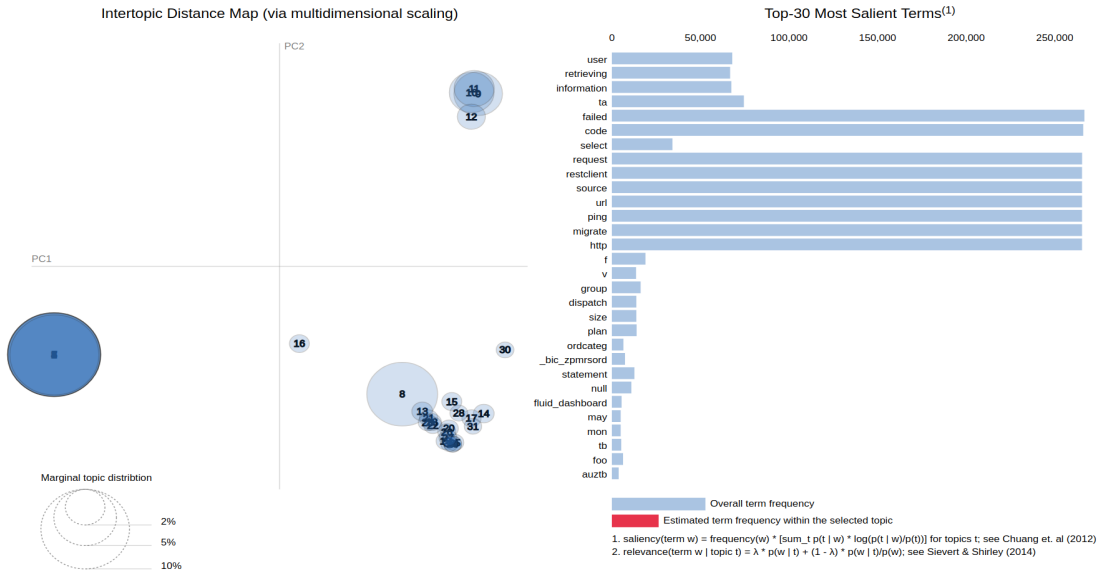


Figure 7.23: PyLdavis topic visualisation with 32 topics

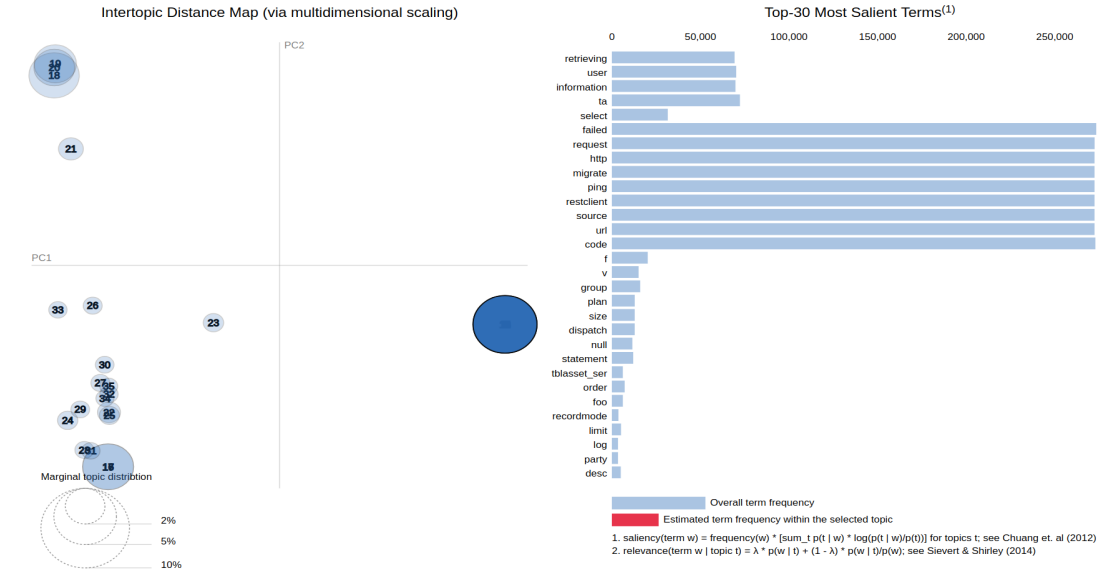


Figure 7.24: PyLdavis topic visualisation with 35 topics

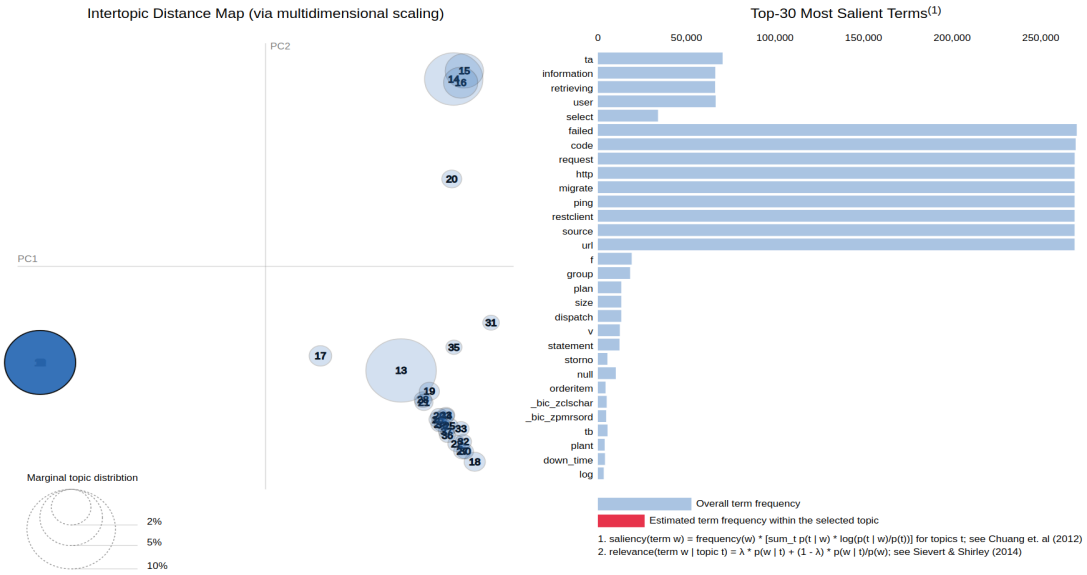


Figure 7.25: PyLdavis topic visualisation with 38 topics

7.3 Document distributions per amount of topics

7.3.1 Train test

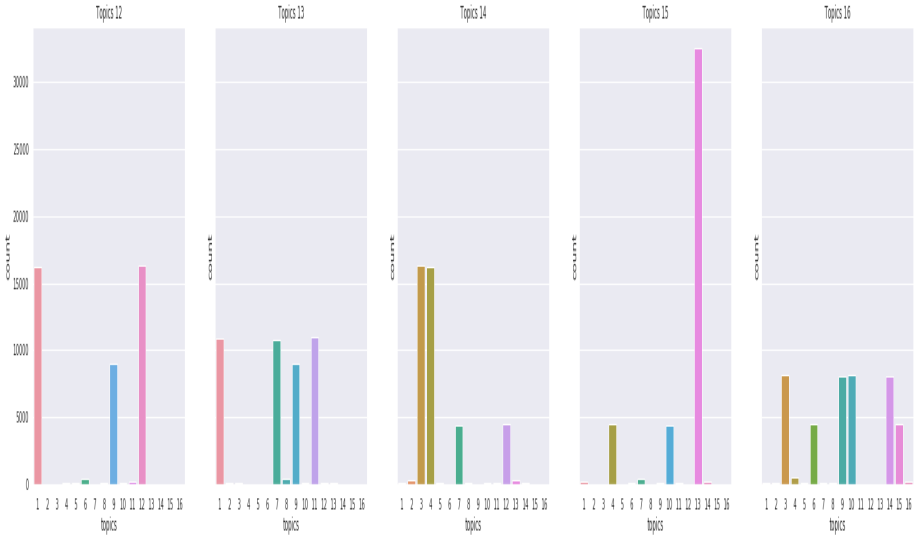


Figure 7.26: Document distribution with 12-16 topics

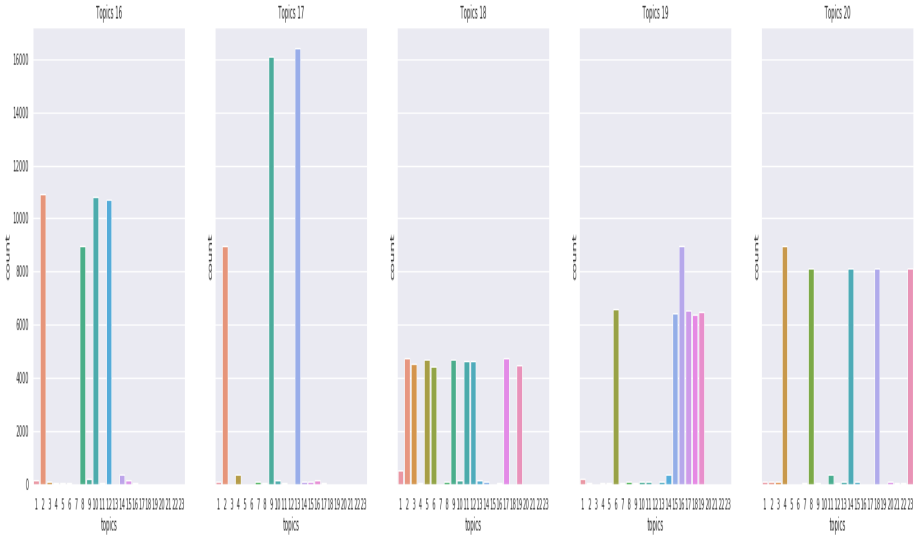


Figure 7.27: Document distribution with 17-23 topics

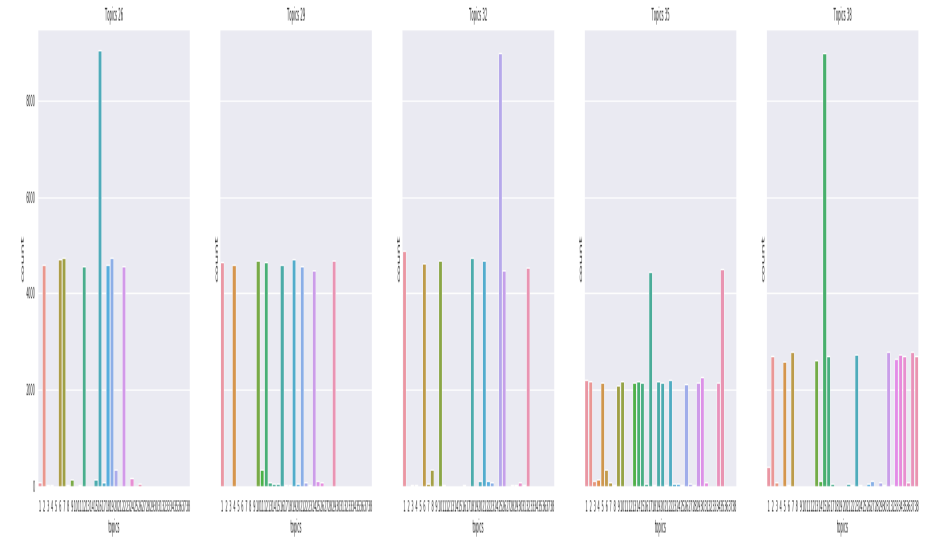


Figure 7.28: Document distribution with 26-38 topics

7.3.2 Held out

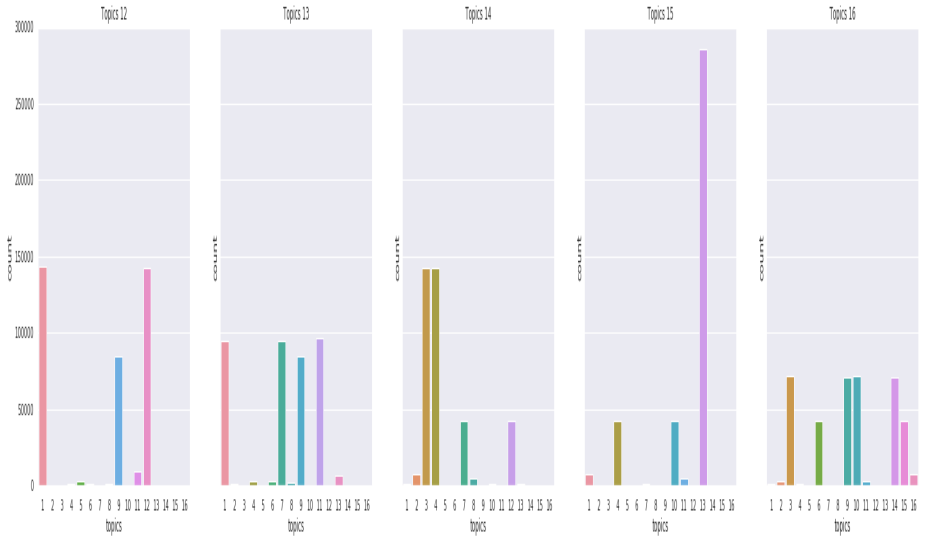


Figure 7.29: Document distribution with 12-16 topics

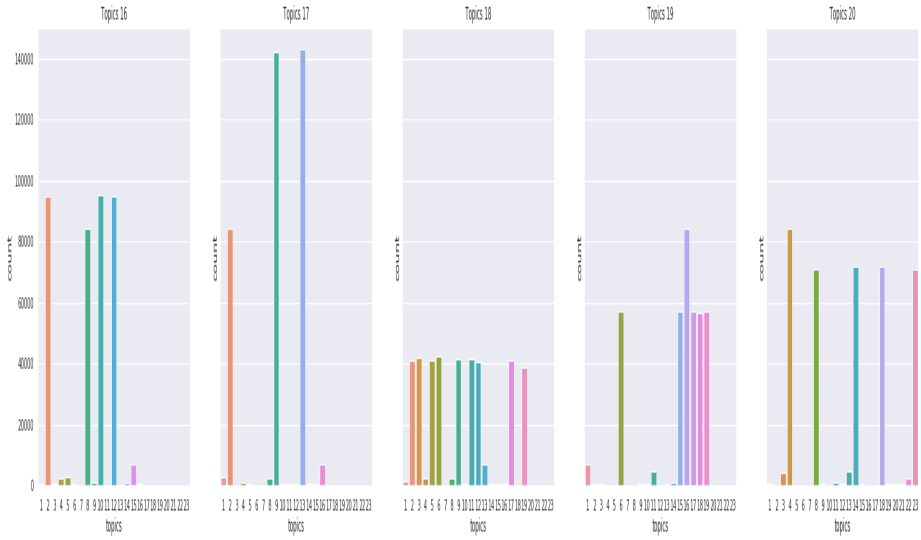


Figure 7.30: Document distribution with 17-23 topics

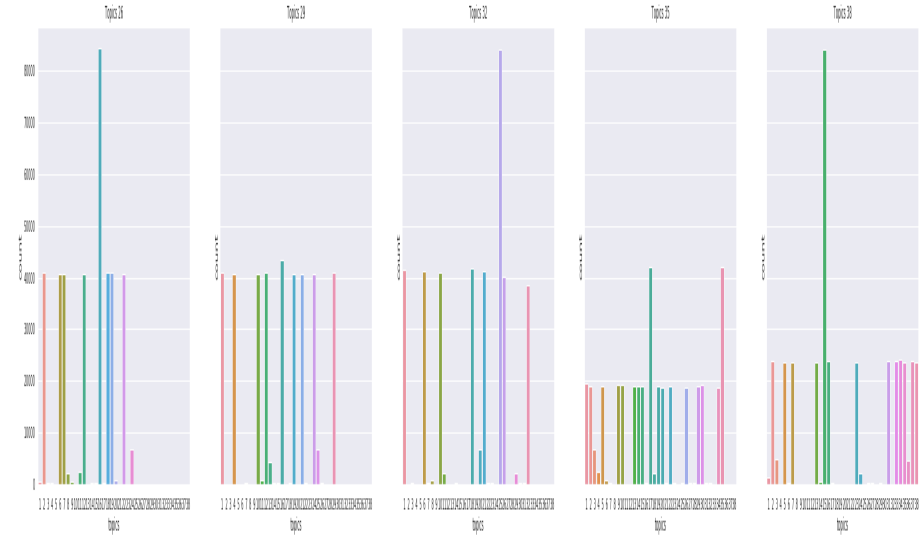


Figure 7.31: Document distribution with 26-38 topics

7.4 Model topic overview

Topic	Terms				
0	user	information	retrieving	ta	select
1	failed	code	request	restclient	ping

Table 7.2: Topic 1..2 with top 5 terms

Topic	Terms				
0	failed	code	request	source	ping
1	ta	select	f	group	plan
2	user	information	retrieving	text	log

Table 7.3: Topic 1..3 with top 5 terms

Topic	Terms				
0	fluid_dashboard	text	tblasset_ser	recordmode	log
1	ta	select	f	group	plan
2	user	information	retrieving	cancel	token
3	failed	code	request	source	restclient

Table 7.4: Topic 1..4 with top 5 terms

Topic	Terms				
0	ta	select	f	group	plan
1	fluid_dashboard	text	tblasset_ser	tbljob_sow	group
2	v	r	order	null	tb
3	failed	code	request	source	restclient
4	user	information	retrieving	log	party

Table 7.5: Topic 1..5 with top 5 terms

Topic	Terms				
0	failed	code	request	source	url
1	fluid_dashboard	text	tblasset_ser	tbljob_sow	coord_type
2	user	information	retrieving	controller	could
3	count	plant	redo	customer	record
4	ta	select	f	group	plan
5	v	order	r	null	tb

Table 7.6: Topic 1..6 with top 5 terms

Topic	Terms				
0	ta	select	size	dispatch	plan
1	notificatn	coord_type	rowcnt	tblasset_ser	_bic_zvorgang
2	user	information	retrieving	terminated	slice_id
3	f	v	group	limit	desc
4	fluid_dashboard	text	tblasset_ser	tbljob_sow	orderitem
5	v	r	order	tb	null
6	failed	code	request	url	source

Table 7.7: Topic 1..7 with top 5 terms

Topic	Terms				
0	request	ping	source	restclient	url
1	log	party	_bic_zanzsn	orderitem	_bic_zvorgang
2	request	ping	migrate	source	http
3	coord_type	ordcateg	cancel	token	_bic_zpmrsord
4	user	information	retrieving	may	mon
5	ta	select	f	group	plan
6	failed	plant	group	code	controller
7	fluid_dashboard	text	tblasset_ser	tbljob_sow	rowcnt

Table 7.8: Topic 1..8 with top 5 terms

Topic	Terms				
0	ta	f	group	limit	desc
1	fluid_dashboard	text	tblasset_ser	tbljob_sow	may
2	log	party	coord_type	storno	employee
3	ta	select	size	dispatch	plan
4	failed	code	request	source	url
5	ordcateg	unit_day	auztb	ops	hawqstatus
6	user	information	retrieving	terminated	stage
7	cancel	rowcnt	redo	token	record
8	v	r	select	null	order

Table 7.9: Topic 1..9 with top 5 terms

Topic	Terms				
0	request	migrate	restclient	url	ping
1	fluid_dashboard	text	tblasset_ser	tbljob_sow	coord_type
2	ordcateg	cancel	token	postgres	_bic_zpmrsord
3	orderitem	redo	record	unit_day	length
4	f	foo	count	ta	v
5	ops	hawqstatus	down_indic	set	_bic_zlongit
6	request	ping	migrate	url	http
7	failed	material	recordmode	notificatn	group
8	ta	select	group	plan	dispatch
9	user	information	retrieving	tblasset_ser	tbljob_sow

Table 7.10: Topic 1..10 with top 5 terms

Topic	Terms				
0	request	ping	source	restclient	url
1	failed	group	controller	code	could
2	ta	select	plan	dispatch	size
3	_bic_zanzsn	orderitem	_bic_zvorgang	_bic_zpmrsord	set
4	coord_type	storno	unit_day	division	employee
5	ta	f	group	v	select
6	request	source	url	restclient	code
7	user	information	retrieving	tblasset_ser	tbljob_sow
8	ta	v	select	null	r
9	plant	redo	customer	record	length
10	fluid_dashboard	text	tblasset_ser	log	party

Table 7.11: Topic 1..11 with top 5 terms

Topic	Terms				
0	request	ping	migrate	source	http
1	plant	redo	customer	record	length
2	ordcateg	unit_day	notif_orgn	down_time	auztv
3	failed	controller	group	code	could
4	ta	f	order	v	null
5	fluid_dashboard	text	log	party	tblasset_ser
6	ta	material	recordmode	notificatn	createdon
7	tblasset_ser	cancel	tbljob_sow	orderitem	token
8	information	user	retrieving	_bic_zpmrsord	_bic_zobjvw
9	_bic_zanzsn	coord_type	may	mon	_bic_zvorgang
10	ta	select	group	plan	dispatch
11	request	ping	source	restclient	url

Table 7.12: Topic 1..12 with top 5 terms

Topic	Terms				
0	request	ping	url	source	restclient
1	failed	group	controller	code	could
2	cancel	token	postgres	auztb	user
3	ta	f	v	order	null
4	coord_type	rowcnt	storno	unit_day	division
5	ta	f	select	v	group
6	request	ping	url	source	restclient
7	fluid_dashboard	text	tblasset_ser	log	party
8	information	retrieving	user	_bic_zpmrsord	_bic_zobjvw
9	_bic_zanzsn	_bic_zvorgang	_bic_zqmdat	_bic_zbautl_hd	set
10	request	code	failed	source	restclient
11	redo	record	length	checkpoint	restart
12	ta	select	plan	dispatch	size

Table 7.13: Topic 1..13 with top 5 terms

Topic	Terms				
0	failed	group	controller	code	could
1	ta	select	plan	dispatch	size
2	request	restclient	failed	source	code
3	request	restclient	url	source	failed
4	fluid_dashboard	text	recordmode	_bic_zanzsn	_bic_zvorgang
5	v	ta	select	r	null
6	information	retrieving	user	_bic_zpmrsord	_bic_zobjvw
7	ta	f	select	group	v
8	ta	unit_day	auztb	_bic_zpmrsord	division
9	cancel	rowcnt	token	postgres	user
10	tblasset_ser	coord_type	tbljob_sow	storno	down_indic
11	information	user	retrieving	_bic_zpmrsord	_bic_zobjvw
12	log	party	mat_plant	p_plant	_bic_zpsttr
13	ordcateg	redo	record	length	checkpoint

Table 7.14: Topic 1..14 with top 5 terms

Topic	Terms				
0	ta	select	plan	dispatch	size
1	_bic_zanzsn	_bic_zvorgang	record	length	checkpoint
2	ta	material	recordmode	notificatn	_bic_zobzae
3	information	retrieving	user	_bic_zpmrsord	down_time
4	ordcateg	class_num	order_quan	class_type	_bic_zangeb
5	orderitem	not_type	notif_orgn	_bic_zbautl_hd	ausbs
6	log	party	cancel	token	postgres
7	plant	customer	wbs_elemt	redo	costcenter
8	fluid_dashboard	text	tblasset_ser	tbljob_sow	auztb
9	user	retrieving	information	_bic_zpmrsord	proxy
10	ta	f	group	select	v
11	set	search_path	saperrorcode	saperrormessage	job_guid
12	failed	code	request	source	url
13	may	mon	employee	quantity	amountfx
14	coord_type	ta	down_time	storno	_bic_zpmrsord

Table 7.15: Topic 1..15 with top 5 terms

Topic	Terms				
0	v	r	select	order	null
1	ta	f	select	foo	v
2	failed	code	source	restclient	url
3	log	party	coord_type	rowcnt	may
4	unit_day	employee	quantity	amount	amountvr
5	information	retrieving	user	_bic_zpmrsord	down_time
6	cancel	token	tbljob_sow	postgres	tblasset_ser
7	fluid_dashboard	text	_bic_zanzsn	tblasset_ser	_bic_zvorgang
8	failed	code	request	source	url
9	failed	code	request	http	ping
10	ta	f	limit	v	desc
11	orderitem	tblasset_ser	_bic_zpmrsord	class_type	class_num
12	_bic_zqmdat	_bic_zbautl_hd	set	_bic_znummanf	_bic_znumsgpw
13	failed	code	request	url	source
14	information	user	retrieving	side	extension
15	ta	select	size	dispatch	plan

Table 7.16: Topic 1..16 with top 5 terms

Topic	Terms				
0	ordcateg	cancel	token	postgres	hdfs
1	request	url	failed	source	restclient
2	fluid_dashboard	text	tblasset_ser	tbljob_sow	down_indic
3	ta	f	limit	v	desc
4	ta	f	select	v	foo
5	plant	redo	record	length	checkpoint
6	coord_type	down_time	storno	_bic_zclschar	_bic_zsystatus
7	user	retrieving	information	first	without
8	rowcnt	may	mon	ops	hawqstatus
9	request	ping	source	code	failed
10	_bic_zanzsn	_bic_zvorgang	_bic_znumsgpw	_bic_zlatit	_bic_zlongit
11	restclient	ping	source	failed	code
12	unit_day	division	_bic_zgsmng	_bic_zangeb	_bic_zbedarf
13	log	party	not_type	notif_orgn	ausvn
14	ta	select	plan	dispatch	size
15	failed	controller	code	could	details
16	orderitem	_bic_zpmrsord	class_num	class_type	partno

Table 7.17: Topic 1..17 with top 5 terms

Topic	Terms				
0	ta	f	select	v	foo
1	information	user	retrieving	_bic_zpmrsord	proxy
2	ta	not_type	notif_orgn	unit_day	ausbs
3	log	party	ordcateg	po_unit	ord_typ
4	v	select	r	ta	null
5	orderitem	division	zzwbs	zzdber	equnr
6	may	mon	class_num	class_type	e
7	ta	f	group	limit	desc
8	failed	code	request	url	restclient
9	auztb	client	eof	sales_unit	n
10	plant	redo	customer	record	length
11	_bic_zanzsn	_bic_zvorgang	down_indic	_bic_zlongit	_bic_znumsgpw
12	failed	code	request	source	restclient
13	fluid_dashboard	text	rowcnt	tblasset_ser	time
14	cancel	token	postgres	user	hdfs
15	ta	select	plan	dispatch	size
16	tblasset_ser	tbljob_sow	ops	hawqstatus	saperrorcode
17	coord_type	_bic_zpmrsord	storno	down_time	_bic_zclschar

Table 7.18: Topic 1..18 with top 5 terms

Topic	Terms				
0	fluid_dashboard	text	log	party	rowcnt
1	request	restclient	failed	source	code
2	information	retrieving	user	_bic_zpmrsord	down_time
3	ta	f	select	v	foo
4	restclient	ping	source	failed	code
5	information	retrieving	user	_bic_zpmrsord	down_time
6	_bic_zanzsn	_bic_zvorgang	down_indic	_bic_zlatit	_bic_znummanf
7	ta	f	limit	desc	order
8	code	ping	source	url	restclient
9	unit_day	_bic_zpmrsord	client	eof	n
10	request	ping	source	code	failed
11	request	ping	source	restclient	failed
12	ta	select	plan	size	dispatch
13	failed	controller	group	code	could
14	v	ta	select	r	null
15	redo	record	length	checkpoint	restart
16	restclient	ping	source	failed	code
17	tblasset_ser	createdon	assembly	tbljob_sow	orderitem
18	request	failed	url	source	restclient

Table 7.19: Topic 1..19 with top 5 terms

Topic	Terms				
0	ta	select	statement	plan	size
1	redo	record	length	checkpoint	restart
2	v	r	select	order	null
3	fluid_dashboard	text	tblasset_ser	tbljob_sow	down_indic
4	failed	group	material	equipment	recordmode
5	request	ping	source	code	failed
6	terminated	stage	search	cest	stack_trace
7	may	mon	_bic_zqmdat	_bic_znumsgpw	_bic_zlongit
8	orderitem	_bic_zpmrsord	ops	hawqstatus	dispatch
9	ordcateg	unit_day	auztb	client	eof
10	ta	f	select	v	group
11	not_type	notif_orgn	storno	ausbs	ausvn
12	cancel	token	postgres	delegation	hdfs
13	log	party	_bic_zanzsn	rowcnt	_bic_zvorgang
14	request	restclient	url	source	code
15	user	information	retrieving	simpleajpservice	e
16	request	ping	source	code	failed
17	code	ping	source	failed	restclient
18	request	ping	source	code	failed
19	coord_type	without	employee	quantity	quantityfx

Table 7.20: Topic 1..20 with top 5 terms

Topic	Terms				
0	redo	record	checkpoint	length	restart
1	_bic_zobknr	assembly	rowcnt	may	mon
2	ta	select	group	f	foo
3	information	user	retrieving	without	first
4	not_type	notif_orgn	storno	oi_ebelp	mat_plant
5	auztb	class_type	class_num	job_guid	_bic_zperidint
6	fluid_dashboard	text	_bic_zanzsn	_bic_zvorgang	down_indic
7	request	url	code	source	restclient
8	failed	controller	could	code	policy
9	_bic_zqmdat	saperrormessage	saperrorcode	sapstatus	partno
10	log	party	po_unit	_bic_zcslngtxt	_bic_zrev_lvl
11	_bic_zbautl_hd	client	eof	sales_unit	ord_typ
12	ta	select	plan	dispatch	size
13	request	restclient	url	source	code
14	tblasset_ser	tbljob_sow	serial_guid	groupnumber	downloadtoscope
15	coord_type	unit_day	division	down_time	_bic_zsystatus
16	equipment	_bic_zobzae	ta	ausvn	ausbs
17	request	source	code	url	restclient
18	v	select	r	ta	order
19	ordcateg	orderitem	cancel	token	postgres
20	ops	hawqstatus	_bic_zpmrsord	set	search_path
21	ta	f	limit	v	group
22	code	ping	source	failed	restclient

Table 7.21: Topic 1..23 with top 5 terms

Topic	Terms				
0	ordcateg	ops	hawqstatus	saperrormessage	saperrorcode
1	request	ping	source	code	failed
2	auztb	_bic_zbautl_hd	set	search_path	unnamed
3	failed	controller	could	code	details
4	down_indic	_bic_zlatit	_bic_zlongit	_bic_znumsgpw	_bic_znumoiw
5	request	restclient	url	source	code
6	failed	ping	url	source	code
7	ta	f	limit	order	desc
8	_bic_zpmrsord	down_time	client	eof	n
9	notificatn	_bic_zobzae	_bic_zobjvw	_bic_zqmdat	ta
10	ta	f	select	group	v
11	code	ping	source	failed	restclient
12	class_type	class_num	partno	equnr	zzwbs
13	v	r	select	order	null
14	rowcnt	may	mon	employee	quantity
15	information	user	retrieving	terminated	slice_id
16	cancel	orderitem	token	postgres	authorized
17	request	ping	source	code	failed
18	request	ping	source	code	failed
19	log	party	po_unit	_bic_zperidint	p_plant
20	record	checkpoint	without	first	starting
21	code	ping	source	failed	restclient
22	not_type	notif_orgn	unit_day	ausbs	auztv
23	ta	select	plan	size	dispatch
24	_bic_zanzsn	coord_type	_bic_zvorgang	storno	division
25	fluid_dashboard	text	tblasset_ser	recordmode	tbljob_sow

Table 7.22: Topic 1..26 with top 5 terms

Topic	Terms				
0	request	ping	source	code	failed
1	select	not_type	notif_orgn	auztv	ausbs
2	tblasset_ser	tbljob_sow	saperrorcode	saperrormessage	sapstatus
3	source	ping	url	failed	restclient
4	plant	record	redo	starting	checkpoint
5	notificatn	_bic_zobzae	_bic_zobknr	coord_type	_bic_zobjvw
6	v	select	r	ta	order
7	unit_day	employee	quantity	amountfx	quantityfx
8	_bic_zanzsn	ordcateg	_bic_zvorgang	ta	division
9	request	ping	source	code	failed
10	log	party	without	first	sales_unit
11	request	ping	source	code	failed
12	ta	f	v	group	select
13	rowcnt	terminated	stack_trace	cest	stage
14	ops	hawqstatus	plan	dispatch	size
15	information	retrieving	user	user_guid	access
16	_bic_zpmrsord	zzdber	equnr	zzwbs	mat_plant
17	failed	controller	could	code	policy
18	failed	ping	url	source	code
19	fluid_dashboard	text	down_indic	time	timestamp
20	code	ping	source	failed	restclient
21	orderitem	_bic_zlongit	_bic_znummanf	_bic_znumoiw	_bic_zlatit
22	down_time	client	eof	salesorg	distr_chan
23	user	retrieving	information	may	mon
24	ta	select	statement	size	dispatch
25	cancel	token	postgres	fmcprod	user
26	customer	length	redo	wbs_elemt	restart
27	auztb	_bic_zclschar	_bic_zpmrsord	po_unit	ch_on
28	request	restclient	url	source	code

Table 7.23: Topic 1..29 with top 5 terms

Topic	Terms				
0	code	ping	source	failed	restclient
1	partno	zzdber	equnr	zzwbs	datapakid
2	failed	controller	could	code	details
3	record	length	checkpoint	redo	master
4	notif_orgn	unit_day	ausbs	auztv	ausvn
5	request	restclient	url	source	code
6	ops	hawqstatus	plan	size	dispatch
7	log	party	rowcnt	_bic_zdlv_date	_bic_znet
8	wbs_elemt	redo	order_quan	location	starting
9	request	ping	source	code	failed
10	ta	f	select	foo	count
11	ordcateg	sales_unit	invalid	_bic_zartpr	pldreldate
12	coord_type	storno	employee	quantity	quantityfx
13	tb	v	select	ta	r
14	_bic_zlatit	_bic_znumoiw	_bic_znummanf	_bic_zlongit	_bic_znumsgpw
15	text	tblasset_ser	_bic_zanzsn	_bic_zvorgang	tbljob_sow
16	not_type	oi_ebeln	oi_ebelp	mat_plant	mrp_contrl
17	request	ping	source	code	failed
18	_bic_zobknr	assembly	saperrorcode	saperrormessage	_bic_zbearb
19	ta	select	statement	dispatch	size
20	code	ping	source	failed	restclient
21	may	mon	_bic_zclschar	_bic_zauffx	actstartdt
22	_bic_zpmrsord	n	_bic_zclschar	down_time	select
23	orderitem	division	calday	objnr	plgrp
24	user	information	retrieving	user_guid	access
25	restclient	url	code	source	failed
26	auztb	_bic_zpoolk	_bic_zstatusw	_bic_zawvst	salesorg
27	fluid_dashboard	down_indic	class_type	class_num	addedby_guid
28	ta	f	group	v	null
29	cancel	token	postgres	hdfs	authorized
30	_bic_zbautl_hd	_bic_zqmdat	down_time	client	eof
31	url	migrate	restclient	code	ping

Table 7.24: Topic 1..32 with top 5 terms

Topic	Terms				
0	code	ping	source	failed	restclient
1	request	ping	source	code	failed
2	ta	select	statement	plan	size
3	ta	f	select	foo	v
4	code	ping	source	failed	restclient
5	log	party	_bic_zqmdat	_bic_zbautl_hd	_bic_zntfcompl
6	orderitem	cancel	token	postgres	hdfs
7	recordmode	rowcnt	mrp_contrl	_bic_zdlv_date	plnd_delry
8	code	ping	source	failed	restclient
9	code	ping	source	failed	restclient
10	down_indic	_bic_zlongit	_bic_znummanf	_bic_zlatit	_bic_znumoiw
11	not_type	notif_orgn	unit_day	ausbs	auztv
12	code	ping	source	failed	restclient
13	code	ping	source	failed	restclient
14	restclient	ping	code	failed	source
15	tbljob_sow	tblasset_ser	auztb	saperrorcode	saperrormessage
16	information	retrieving	user	user_guid	access
17	ta	f	limit	desc	v
18	request	ping	source	code	failed
19	code	ping	source	failed	restclient
20	failed	notificatn	group	createdon	_bic_zehistty
21	request	ping	source	code	failed
22	storno	ops	hawqstatus	plan	dispatch
23	fluid_dashboard	text	_bic_zanzsn	coord_type	_bic_zvorgang
24	redo	customer	record	checkpoint	length
25	code	ping	source	failed	restclient
26	ordcateg	terminated	search	stack_trace	stage
27	_bic_zpmrsord	job_guid	down_time	client	eof
28	request	ping	source	code	failed
29	request	restclient	url	source	code
30	tblasset_ser	tbljob_sow	select	n	serial_guid
31	v	r	order	null	tb
32	_bic_zobzae	_bic_zobjvw	class_num	class_type	ta
33	source	failed	url	restclient	code
34	user	information	retrieving	user_guid	access

Table 7.25: Topic 1..35 with top 5 terms

Topic	Terms				
0	log	party	without	order_quan	first
1	code	ping	source	failed	restclient
2	ta	f	v	select	group
3	answt	zssupclass	_sapcem_bdpo	zzret_dat	zzcoarea
4	request	ping	source	code	failed
5	redo	record	length	checkpoint	restart
6	request	ping	source	code	failed
7	_bic_zobknr	ordcateg	coord_type	_bic_zpmrsord	sales_unit
8	_bic_znumsgpw	_bic_zlongit	_bic_zlatit	_bic_znummanf	_bic_znumoiw
9	not_type	notif_orgn	unit_day	auztv	ausbs
10	ta	material	recordmode	notificatn	_bic_zobzae
11	mat_plant	cust_desc_guid	mrp_contrl	base_uom	_bic_zeisbe
12	request	ping	source	code	failed
13	r	failed	group	controller	could
14	information	user	retrieving	user_guid	access
15	code	ping	source	failed	restclient
16	ops	hawqstatus	plan	size	dispatch
17	po_unit	_bic_zcomb	_bic_zbedarf	_bic_zkapartxt	opr_plant
18	rowcnt	saperrormessage	saperrorcode	job_guid	sapstatus
19	down_time	calday	_bic_zstatuv	ch_on	_bic_zotype
20	fluid_dashboard	text	tblasset_ser	tbljob_sow	time
21	_bic_zpmrsord	client	eof	txtmd	resp_cctr
22	code	ping	source	failed	restclient
23	select	ta	group	tb	size
24	storno	_bic_zsystatus	finishdate	priority	schedfindt
25	terminated	cest	stage	slice_id	search
26	may	mon	down_indic	_bic_zclschar	_bic_znot_cat
27	plant	customer	wbs_elemt	costcenter	bus_area
28	_bic_zclschar	n	distr_chan	notes	exttointtblerrorrowcount
29	orderitem	partno	zzdber	zzwbs	equnr
30	request	restclient	url	source	code
31	_bic_zbautl_hd	_bic_zqmdat	set	search_path	unnamed
32	code	ping	source	failed	restclient
33	code	ping	source	failed	restclient
34	failed	ping	url	source	code
35	ta	select	statement	plan	dispatch
36	code	ping	source	failed	restclient
37	code	ping	source	failed	restclient

Table 7.26: Topic 1..38 with top 5 terms

Bibliography

[Aja13] Ajay Chandramouly, Ravindra Narkhede, Vijay Mungara, Guillermo Rueda, Asoka Diggs. Reducing Client Incidents through Big Data Predictive Analytics. *Intel IT Big Data Predictive Analytics*, (December), 2013.

[CB12] Allison J.B. Chaney and David M. Blei. Visualizing Topic Models. *International AAAI Conference on Weblogs and Social Media*, pages 419–422, 2012.

[CGWB09] Jonathan Chang, Sean Gerrish, Chong Wang, and David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. *Advances in Neural Information Processing Systems 22*, pages 288–296, 2009.

[Dav03] David M. Blei, Andrew Y. Ng, Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[FT04] Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 31. IEEE, 2004.

[HKN14] Jingwei Huang, Zbigniew Kalbarczyk, and David M. Nicol. Knowledge discovery from big data for intrusion detection using lda. In *Big data (BigData Congress), 2014 IEEE international congress on*, pages 760–761. IEEE, 2014.

[HO07] John R. Hershey and Peder A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.

[Hof17] Matthew D. Hoffman. Learning Deep Latent Gaussian Models with Markov Chain Monte Carlo. *International conference on Machine Learning*, (3):1510–1519, 2017.

[Hua08] Anna Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, pages 49–56, 2008.

[LMZ11] Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. *Information Retrieval*, 14(2):178–203, 2011.

[LRU14] Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman. *Mining of massive datasets*. Cambridge university press, 2014.

[Mat10] Matthew D. Hoffman, Francis Bach, David M. Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[Mat12] Matthew D. Hoffman and David M. Blei, Chong Wang, John Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14:1303–1347, 2012.

[MBCD10] David M. Blei, Lawrence Carin, and David Dunson. Probabilistic topic models. *IEEE Signal Processing Magazine*, 55(4):77–84, 2010.

[RBH15] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM, 2015.

[Rou87] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[Sam59] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[Set94] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

[SFMW14] Ruben Sipos, Dmitriy Fradkin, Fabian Moerchen, and Zhuang Wang. Log-based predictive maintenance. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1867–1876. ACM, 2014.

[SS14] Carson Sievert and Kenneth Shirley. Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70, 2014.

[TRH16] Ben Towne, Carolyn P. Rosé, and James D. Herbsleb. Measuring similarity similarly: Lda and human perception. *ACM Transactions on Intelligent Systems and Technology*, 8(1), 2016.

[Wik18] Wikipedia contributors. Text normalization — Wikipedia, the free encyclopedia, 2018. [Online; accessed 22-August-2018].

[ZJW+11] Wayne X. Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *European conference on information retrieval*, pages 338–349. Springer, 2011.