Application of AI on IT Service Management

Arun Singh SIVAPRAKASH
Pramod Kumar NAGARAJ
Van Tien NGUYEN
Alexander POPPE

EPITA Graduate School of Computer Science

Advisor
Olivier BERTHET

Master of Science in Artificial Intelligence Systems

16-07-2021

Contents

Abstract

Large enterprise Information technology infrastructure components generate large volume of alerts and incident tickets. These are manually screened, but it is otherwise difficult to extract information automatically from them to gain insight in order to improve operational efficiency. We propose a framework to cluster alters and incident tickets based on the text in them, using unsupervised machine learning. This would be a step towards eliminating manual classification of the alters and incidents, which is very labour intense and costly.

The challenges that arrive at analysing huge amount text data starts with the data quality. Data quality metrics concerning the interpretation of text are too difficult to compute automatically and too pragmatic because they heavily depend on the linguistic content and the specific situation in which the text is being used. Several statistical approaches have been used to determine some level of data quality in this project.

The visual representation of text data helps to bring more insights from the data. Building dashboard with various filters will help the users to customize the search. We have implemented state of art visualization dashboards with different open-source tools.

We proposed the techniques to visualize the clusters that make human readable in an easy and understandable way compared to the traditional methods, we find a simple way to define prototype of cluster for easy interpretation. This framework for clustering and visualization will enable enterprises to prioritize the issue in their IT infrastructure and improve the reliability and availability of their services.

Chapter 1

Introduction

1.1 Motivations

Organization of all sizes struggles with a common problem in Infrastructure and Operational Management. Thousands of automated alerts with semi- structure text are generated every day from hundreds of infrastructure tools. There are thousands of incidents tickets with manually entered unstructured text are created daily by support personnel. After collecting all the details of the incidents maintaining of those tickets is very hard and costly and difficult to analysis each incident without grouping them. This is not only a frustrating situation for the company but also for clients.

Many of us have been forced to take time out of our schedule to e-mail or call a helpdesk with an IT related question. Redirection of the calls in order to get the right person on the line is not very rare. This can cause some frustration and time which is not beneficial for both parties.

IT problems exist and they will always exist. We can try to prevent them as much as possible but that is not the only solution. Dealing with them in an appropriate and efficient manner is the goal of this research. By looking into the similarity between tickets, tickets can be directly assigned to a dedicated team. This should improve the efficiency and reduce any unnecessary redirection. This research also looks into the quality of a ticket. This provides more insight in how problems are documented and where to improve the service. A good documentation of a problem will always help to find a solution.

1.2 Context of the Project

Organization operating in IT enables business segment over a decade. On an average, the company receives more than 25k IT incidence or tickets. Which are handled to best practice by certain framework which are done by researchers with incident management, problem management etc.

We are developing the framework which help the IT tech companies to visualize the data quality and data clustering. With this, the company can understand the problem of the user in systematic manner within no time, and the companies auto robots can answer for the user problem with more effectively.

1.3 Objectives and Contributions

Objective of this project is to provide the enterprise various tools and mediums that leverages the latest State of Art NLP and machine learning techniques to analyse their incident management system. These tools will bring out the hidden insights in the ticket data which will enable the users to take data driven decisions for the business improvements.

Data quality of the raw text data has been identified with statistical tests. Insight's dashboard can help the company to improve the performance of the service teams and to track which sections need more attention. This will also help in the resource allocation among various sections and also in effective cost management. Clustering application will bring out the similar ticket records based on the input description, this will help the teams to get the details of tickets which were similar to the entered description and how they were resolved before. This will help in reducing time ticket resolve time.

1.4 Overview of thesis

In this thesis we have implemented various ML and NLP techniques to generate the business insights from the raw text data. We discussed in detail about the clustering of similar tickets and its implementation using various technologies in real time.

Chapter 2

State of Art

Natural language processing (NLP)[1] is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of ”understanding” the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves. In our project, NLP is used to extract information tickets and evaluate the quality of data.

2.1 NLP pipeline

In general, NLP contains three main stages: text processing, feature extraction, modeling[2].

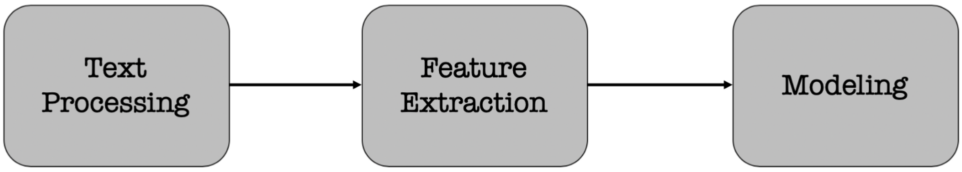


Figure 2.1: NLP pipeline

2.1.1 Text Processing

In this step, we take text raw as an input. Then we process the input into a form that is the best for next step: feature extraction. More specifically, raw data needs to be cleaned that means removing special characters such as HTML tags, etc. In other words, input data does not contain any info for the model to lean and are irrelevant or noisy data. After that, we do the data normalization which involes the case normalization, punctuation removal so that the text is in a single format for the machine to learn. In case normalization, we convert all capitalization to lower to bring a common case, and in punctuation removal, we replace all punctuations with space. The next process is breaking up text documents into individual words called tokens and remove the stop words. Stop word removal means removing the non-important words like "a", "is", "the", "and", "an", etc.

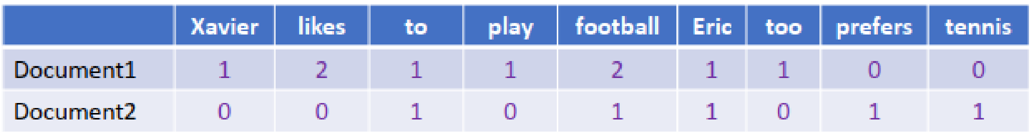
After that, we need to reduce words to its normalized form, and this process can be done by stemming or lemmatization technique. Stemming is a process of reducing a word to its root form, for example: "branching", "branched" and "branches" can all be reduced to "branch", meanwhile lemmatization is the algorithmic process of determining the canonical form (lemma) of a word. Unlike stemming, it is not only a word reduction but depends on the meaning of the word in a sentence and needs to consider a language's full vocabulary, for instance: "was" will be transformed into "be", "meeting" will be "meet" or "meeting" depending on the context.

2.1.2 Feature Extraction

Feature Extraction is a way of extracting feature vectors from the text after the text processing step so that it can be used in the machine learning model as input. Word Embedding is one such technique where we can represent the text using vectors. The more popular forms of word embeddings are: Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF).

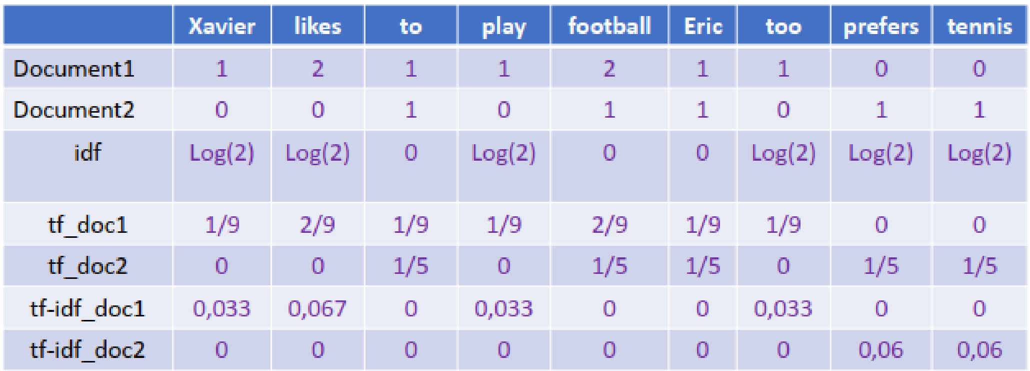
Bow is a way of extracting features from the text for use in modeling, it treats each document as a collection or bag of word. This means a representation of text that describes the occurrence of words within a document. It involes two things: a vocabulary of known words in the corpus (set of documents) and a measure of the presence of known words. Term Frequency-Inverse Document Frequency is a ponderation method uused in information retrieval. This statistical measure gives an evaluation of how important is a word to a document, depending on the corpus considered. TF-IDF is calculated as: $tfidf(t,d,D) = tf(t,d) * idf(t,D)$. Inverse document frequency: $idf(t,D) = \log(\frac{N}{|\{d \in D : t \in d\}|})$ with N the number of documents in corpus D.

For example: we have two documents which are Document1 "Xavier likes to play football. Eric likes football too." and Document2 "Eric prefers tennis to football", we can have the representation below:



	Xavier	likes	to	play	football	Eric	too	prefers	tennis
Document1	1	2	1	1	2	1	1	0	0
Document2	0	0	1	0	1	1	0	1	1

Figure 2.2: Example of BoW



	Xavier	likes	to	play	football	Eric	too	prefers	tennis
Document1	1	2	1	1	2	1	1	0	0
Document2	0	0	1	0	1	1	0	1	1
idf	Log(2)	Log(2)	0	Log(2)	0	0	Log(2)	Log(2)	Log(2)
tf_doc1	1/9	2/9	1/9	1/9	2/9	1/9	1/9	0	0
tf_doc2	0	0	1/5	0	1/5	1/5	0	1/5	1/5
tf-idf_doc1	0,033	0,067	0	0,033	0	0	0,033	0	0
tf-idf_doc2	0	0	0	0	0	0	0	0,06	0,06

Figure 2.3: Example of TF-IDF calculation

2.1.3 Modeling

The final stage of the NLP pipeline is modeling, which includes designing a statistical or machine learning model, fitting its parameters to training data, using an optimization procedure, and then using it to make predictions about unseen data. In our project, we use clustering machine learning algorithms.

Clustering is an unsupervised machine learning task. It involves automatically discovering natural grouping in data. Unlike supervised learning, clustering algorithms only interpret the input data and find natural groups or clusters in features in feature space. A cluster is often an area of density in the feature space where examples from the domain (observations or rows of data) are closer to the cluster than other clusters. The cluster may have a center (the centroid) that is a sample or a point feature space and may have a boundary or extent. Clustering can be helpful as a data analysis activity in order to learn more about the problem domain. It also can be useful as a type of feature engineering, where existing and new examples can be mapped and labeled as belonging to one of the identified clusters in the data.

Evaluation of identified clusters is subjective and my require a domain expert, although many clustering-specific quantitative measures do exist. Typically, clustering algorithms are compared academically on synthetic datasets with pre-defined clusters which an algorithm is expected to discover. There are two common clustering algorithms: K-Means and Gaussian Mixture Model.

K-Means:

This is one of the simplest and frequently used unsupervised learning algorithms, especially in data mining and statistics. Being a portioning algorithm, its goal is to form groups of data points based on the number of clusters, represented by the variable k. K needs to be predefined before the execution. K-means uses an iterative refinement method to produce its final clustering based on the number of clusters defined by the user and the dataset. Initially, k-means randomly chooses k as the mean values of k clusters, called centroids, and find the nearest data points of the chosen centroids to form k clusters. Then, it iteratively recalculates the new centroids for each cluster until the algorithm converges to one optimum value. K-means clustering would be suited with the numerical data with a low dimensionality because numerical data is used to compute the mean value. The type of data best suited for K-means clustering would be numerical data with a relatively lower number of dimensions.

Problem description[3]: A set of observations $(x_1, x_2, \ldots, x_n)$ where each observation is a d-dimensional real vector, this algorithm aims to partition the n observations into $k(<= n)$ sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within cluster sum of squares.

Algorithm[3]: initial randomly set of k means $(m_1, m_2, \ldots, m_k$, the algorithm proceeds by altering between two steps. The first step is assignment step: assign each observation to the cluster with the nearest mean that with the least squared Euclidean distance: $S_i = \arg\min_k ||x_i - m_k||^2$. The second step is update step which recalculates the means (centroids) for observation assigned to each cluster: $m_k = \frac{\Sigma_{i:k_i=k} x_i}{|i:k_i=k|}$. The algorithm has converged when the assignments no longer change.

K-means has some advantages[4] which are simple clustering algorithm so that it can be implemeneted easily. It also faster computationally than other clustering algorithms because K-means has only a few comupations. Additionally, this algorithm can scale up to large dataset and easily adapt to new data samples. Besides, there are some disavantages: K-means has difficulty with clustering data set of varying sizes and density. Its result can also vary depending on inital values and the number is clusters has to be specified manually.

Gaussian Mixture Model (GMM):

Probabilistic models and use the soft clustering approach for distributing the points in different clusters, it assumes that there are a certain number of Gaussian distributions and each of these distributions represent a cluster. therefore, a GMM tends to group the data points belonging to a single distribution together.

Gaussian distribution (or Normal distribution)[5]: is a type of continuous probability distribution for a real-valued random variable. The general form of its probability density function: $f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, with parameters: μ is mean of the distribution and σ is standard deviation.

To implement GMM, we use Expectation-Maximization algorithm (EM)[5]. EM is a statistical algorithm for finding the right model parameters. It typically used for handling missing values or in other words latent variables. This algorithm has two steps: E-step, the available data is used to estimate the values of the missing variables and M-step, based on the estimated values generated in the e-step, the complete data is used to update the parameters.

EM in GMM[5]: We need to assign k number of clusters. This means that there are k Gaussian distributions, with the mean and covariance values to be $\mu_1, \mu_2, ..., \mu_k$ and $\sigma_1, sigma_2, ..., sigma_k$. Additionally, there is another parameter for the distribution that defines the number of points for the distribution. Or in other words, the density of the distribution is represented with Π

2.2 Text Data Visualization

Data visualization is essential to assist businesses in quickly identifying data trends, which would otherwise be a hassle. The pictorial representation of data sets allows analysts to visualize concepts and new patterns. With the increasing surge in data every day, making sense of the quintillion bytes of data is impossible without Data Proliferation, which includes data visualization.

Every professional industry benefits from understanding their data, so data visualization [6] is branching out to all fields where data exists. For every business, information is their most significant leverage. Through visualization, one can prolifically convey their points and take advantage of that information

2.2.1 Importance of Data Visualization

A dashboard, graph, infographics, map, chart, video, slide, etc. all these mediums can be used for visualizing and understanding data. Visualizing the data enable decision-makers to interrelate the data to find better insights and reap the importance of data visualization[6], which are:

Analysing the Data in a Better Way

Analysing reports helps business stakeholders focus on the areas that require attention. The visual mediums help analysts understand the key points needed for their business. Whether it is a sales report or a marketing strategy, a visual representation of data helps companies increase their profits through better analysis and better business decisions.

Faster Decision Making

Humans process visuals better than any tedious tabular forms or reports. If the data communicates well, decision-makers can quickly take action based on the new data insights, accelerating decision-making, and business growth simultaneously.

Making Sense of Complicated Data

Data visualization allows business users to gain insight into their vast amounts of data. It benefits them to recognize new patterns and errors in the data. Making sense of these patterns helps the users pay attention to areas that indicate red flags or progress. This process, in turn, drives the business ahead.

2.2.2 Tools used for Text Data Visualization

Data visualization tool provides users with an easier way to create visual representations of large data sets. When dealing with data sets that include hundreds of thousands or millions of data points, automating the process of creating a visualization, at least in part, makes a designer's job significantly easier.

ELK STACK

The ELK[7] stack is an acronym used to describe a stack that comprises of three popular open-source projects: Elasticsearch, Logstash, and Kibana. ELK stack gives us the ability to aggregate logs from all systems and applications, analyze these logs, and create visualizations for application and infrastructure monitoring, faster troubleshooting, security analytics, and more.

E = Elasticsearch

Elasticsearch is an open-source, RESTful, distributed search and analytics engine built on Apache Lucene. Support for various languages, high performance, and schema-free JSON documents makes Elasticsearch an ideal choice for various log analytics and search use cases.

L = Logstash

Logstash is an open-source data ingestion tool that allows you to collect data from a variety of sources, transform it, and send it to your desired destination. With pre-built filters and support for over 200 plugins, Logstash allows users to easily ingest data regardless of the data source or type.

K = Kibana

Kibana is an open-source data visualization and exploration tool for reviewing logs and events. Kibana offers easy-to-use, interactive charts, pre-built aggregations and filters, and geospatial support and making it the preferred choice for visualizing data stored in Elasticsearch.
We have implemented the dashboard with ELK to visualize the ticket data and generate as much insights from it.

Chapter 3

System's Architecture

3.1 Insights Dashboard Architecture

Raw Text data can be sent into ELK through Elasticsearch or Logstash. Logstash parses and the transforms the data and push it into Elasticsearch where the data is indexed. Kibana is used to create charts, graphs and dashboards. ELK is scalable and easier to deploy in both cloud and on-premise architecture. Below is the pictorial representation of implementation.

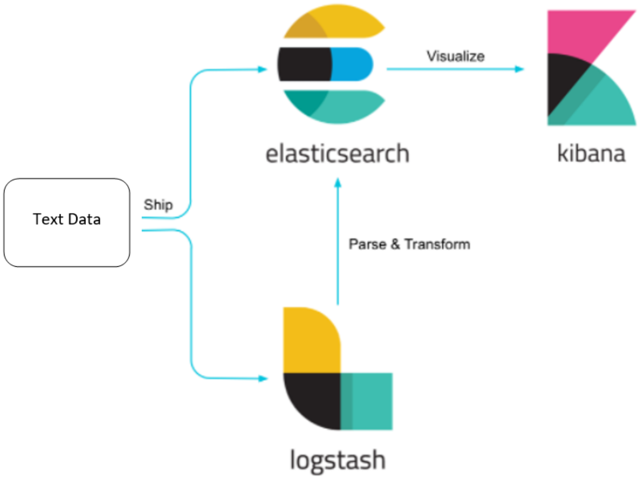


Figure 3.1: Dashboard Artchitecture

3.2 Clustering System Artchitecture

We processed to have a system architecture which is built on the streamlit and FASTAPI framework. We will make use of different tools to build our web application for IT ticket clustering.

Initially, we select the raw data given by the company. We do data pre-processing and this data will be sent to model either KNN or GMM. The model is trained with the pre-processed data. Later, the trained model is called through FASTAPI and post the request to streamlit as a frontend. After deploying the application, the user will be able to rise his incidence ticket and the data will be pre-processed and the model will use this pre-processed text to cluster similar tickets and return the output to frontend.

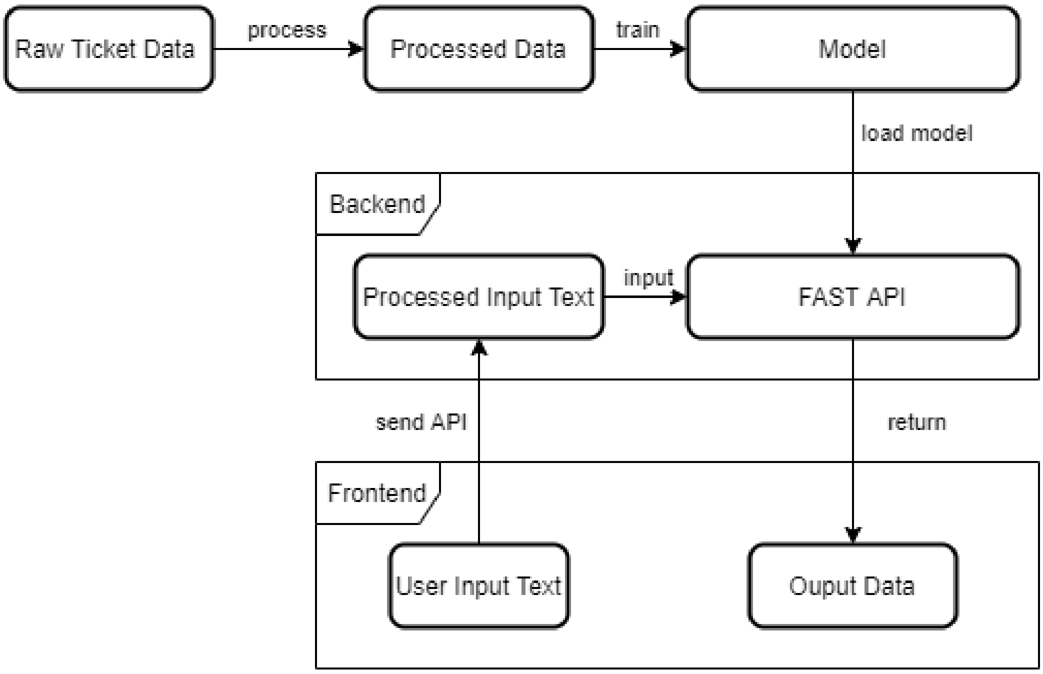


Figure 3.2: Clustering System Artchitecture

Chapter 4

Methodology

4.1 Data Visualization – ELK Dashboard

We have implemented our visualization architecture in a Ubuntu Machine. Single-Node ELK Stack[8] is installed in Ubuntu 20.04. Our text data is fairy small and we are not connected into a live stream of tickets. Hence Single Node ELK Stack is sufficient of our analysis. Data is sent into ELK through Logstash, indexed in elastic search and visualized in KIBANA. Custom filters are created that enables the users to make advanced searches and find insights.
Dash Board can be shared to anyone in the organization's WLAN and also can be hosted online. But accruing SSH certificates and domain registration are beyond the scope of this project.

4.2 Clustering

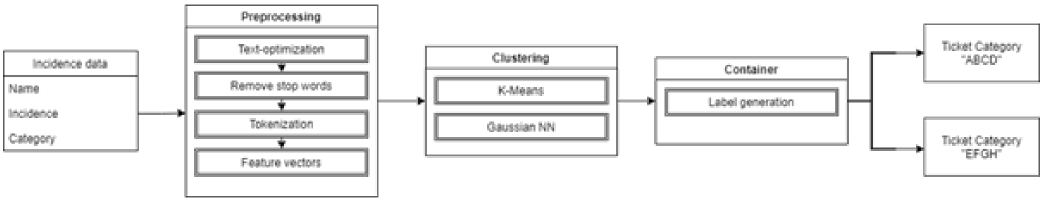


Figure 4.1: Clustering Pipeline

In order to categorize incident tickets, the tickets are pre-processed and converted into feature vectors, and cluster algorithms are applied for categorizations. In the last step, the incidence is meaningfully assigned to a specific category. After initial clustering, the trained algorithm can be saved to get fixed categories and to predict the category of new incidence tickets later. Now we can read about each component in details.

4.3 Data exploratory

In order to have a better understanding of the data collected and integrated for clustering, we performed extensive exploratory data analysis on various features of the data. This step is highly significant in helping us to understand the distribution of target fields and different categories. After pre-processing, we removed the words which had no information using NLTK framework and extract feature vector.

4.4 Algorithm

4.4.1 K-Means

We use K-means because it is a simple baseline algorithm. K-means represent a simple algorithm to classify data into cluster. K-means works in three steps, first it will initiate the random cluster centroids. Second, each data point is assigned to the cluster with the smallest distance. On the last step, the steps 2 are repeated until the cluster centroids unchanged.

4.4.2 Gaussian Mixture Model

We use Gaussian mixture model because it presents a powerful tool for clustering. It assumes that each data group can be reasonably described by one mixture model component. This establishes a one-to-one relationship between mixture component and cluster. The idea for using this algorithm to alleviate the error occurs due to collection of records in different ways and give more efficient results. The working of this model is explained in the state of the art.

4.5 Label generation

Without descriptive cluster names, it is difficult for a human to gather information from the grouped document. Getting the understandable description to human is the most difficultly task. For the algorithm form the field of supervised learning, this is easily possible to form the training data. However, those are not present during clustering. The Cluster labelling is done in many ways. The most popular methods are human labelling, automatic labelling, and semi-automatic labelling. In the human labelling, it's manually labelling the cluster by humans how creates a name after visual inspection of same documents of cluster. Automatic methods is to generate the cluster names from most powerful features within a cluster. Semi-clustering means customizing the automatic cluster names by humans or list of cluster names predefined, subsequently the documents are automatically sorted into categorize. For K-means algorithm, the information can be obtained from the cluster centroids. The cluster centroids are located in the features space and therefore have the same dimension as the tf-idf feature vectors. In order to obtain a list of words sorted by relevance for K-means, the values of cluster centroids must be sorted.

Chapter 5

Results

This document is made in preparation of the actual Action week where our main focus will be the coding and producing actual results. During this preparation and literature review, we are not producing definite results yet. However, we have an idea about what might be possible, and which results previous work yielded. For this chapter, the basis will be previous work and results that they obtained. This will be our goal for next week to reach. We should find adapt existing research to our problem at hand and try to also provide accurate result for this use case.

There has been a lot of research that studies the similarity between tickets. This gives us some interesting information which can be adapted to our use case. At the moment, ServiceNow is a service that is often used to process tickets. Although they are providing a good service, some improvements can be made. Article[9] explores an alternative model to what ServiceNow is providing. They improve the accuracy of ServiceNow from 40% to 85%. This is a very significant improvement and also something we should be aiming for during the implementation of our model.

5.1 Development Anticipated

We have planned perform several statistical tests to analyse the quality of the text data. As a part of insights generation from data we have planned to implement a dashboard with ELK stack. In Clustering Application, we intent to group the cluster the tickets and build a model that can interact with the Streamlit UI and FastAPI to return details of the similar tickets.

5.2 Thesis Development Updated

The quality calculation is based on four metrics: the noun ratio, verb ratio, readability and number of words. This can still be expanded by including more metrics in the formulas. The ratios of punctuation signs such as exclamation marks, points, comma's, etc. could be added. We have used two part of speech (verb and noun), others could also be looked into such as adjectives, adverbs, etc. In our results, we saw that very technical information could result in a low readability score. This is normal since very technical information is often not easy to understand unless you have the specific knowledge. In some cases, this could lead to a higher drop in quality than necessary. Although our metric gives a general good result, more thorough research in the metrics can lead to improvement of the exceptions.

In the clustering, we anticipated using Kmeans and GMM so that we could compare the performance of the two models. However, in real development, we faced some problems with these models. That was the quality of clustering and the performance when training models. Although we used google-colab which runs on GPU, it still took hours to fit the data. Therefore, we changed to use the Latent Dirichlet Allocation (LDA) which returns a better result. We also use CountVectorizer to transform data to get the embedded data. This model not only gives better clustering but also gives a good performance which was about one minute to get vector data and three minutes to train data.

Chapter 6

Conclusion

In this paper, we explained the background of our challenge which is the need for an improved IT System Management (ITSM). This includes two main parts, mapping similarity between tickets and visualising the quality of a ticket. In our literature review, we found state-of-the-art implementations of ITSM and NLP implementations that can help us to reach our goal. The remaining time of this project, we will focus on the coding part and adapting all research to our specific use case. The key in this research is to present our results with to-the-point visualisations for which we will use a dashboard as described in chapter 3 on page 10.

Bibliography

[1] "Natural language processing - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing

[2] "What is Natural Language Processing? Intro to NLP in Machine Learning." [Online]. Available: https://www.gyansetu.in/what-is-natural-language-processing/

[3] "k-means clustering - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/K-means_clustering

[4] N. Minh Le Kazuhiro Ogata, "Master's Thesis Applying Clustering Techniques for Refining Large Data Set (Case Study on Malware) 1710443 Yoon Myet Thwe Supervisor Mizuhito Ogawa Main Examiner Mizuhito Ogawa Examiners Nao Hirokawa," 2019.

[5] "Gaussian Mixture Models — Clustering Algorithm Python." [Online]. Available: https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/

[6] "Importance, Purpose, and Benefit of Data Visualization Tools!" [Online]. Available: https://splashbi.com/importance-purpose-benefit-of-data-visualization-tools/

[7] "The ELK stack." [Online]. Available: https://aws.amazon.com/elasticsearch-service/the-elk-stack/

[8] "Sending Logs to ELK with Winlogbeat and Sysmon – Burnham Forensics." [Online]. Available: https://burnhamforensics.com/2018/11/18/sending-logs-to-elk-with-winlogbeat-and-sysmon/

[9] E. Garrett, "Predict IT Support Tickets with Machine Learning and NLP — by Garrett Eichhorn — Towards Data Science," 2020. [Online]. Available: https://towardsdatascience.com/predict-it-support-tickets-with-machine-learning-and-nlp-a87ee1cb66fc

[10] P. Marcu, G. Grabarnik, L. Luan, D. Rosu, L. Shwartz, and C. Ward, "Towards an optimized model of incident ticket correlation," *2009 IFIP/IEEE*

International Symposium on Integrated Network Management, IM 2009, pp. 569–576, 2009.

[11] A. Rajput, “Natural Language Processing for IT Support Incidents — by Ajit Rajput — Analytics Vidhya — Medium,” 2020. [Online]. Available: https://medium.com/analytics-vidhya/natural-language-processing-for-it-support-incident-51cb35af0735

[12] G. Eichhorn, “Predict IT Support Tickets with Machine Learning and NLP — by Garrett Eichhorn — Towards Data Science,” 2020. [Online]. Available: https://towardsdatascience.com/predict-it-support-tickets-with-machine-learning-and-nlp-a87ee1cb66fc

[13] “sklearn.cluster.KMeans — scikit-learn 0.24.2 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[14] “Natural Language Toolkit — NLTK 3.6.2 documentation.” [Online]. Available: http://www.nltk.org/

[15] K. N. Syeda, S. N. Shirazi, S. Asad, A. Naqvi, and H. J. Parkinson, “Exploiting Natural Language Processing for Analysing Railway Incident Reports,” *IGI Global Publishing*, pp. 1–18, 2017. [Online]. Available: http://www.digitalrail.co.uk/wp-content/uploads/2018/01/Exploiting-Natural-Language-Processing-for-Analysing-Railway-Incident-Reports.pdf

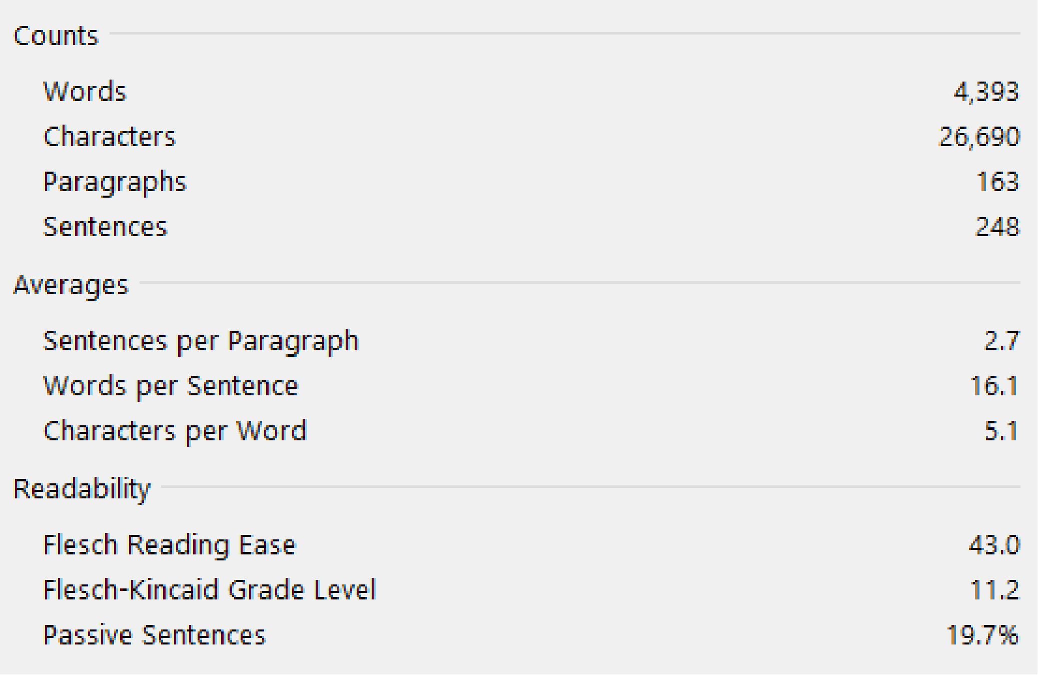
[16] D. Sonntag, “Assessing the quality of natural language text data.”

[17] O. Azeroual, G. Saake, M. Abuosba, and J. Schöpfel, “Text data mining and data quality management for research information systems in the context of open data and open science.”

[18] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: An Efficient Data Clustering Method for Very Large Databases,” *SIGMOD Record (ACM Special Interest Group on Management of Data)*, vol. 25, no. 2, pp. 103–114, 1996.

[19] TsengYuen-Hsien, “Generic title labeling for clustered documents,” *Expert Systems with Applications: An International Journal*, vol. 37, no. 3, pp. 2247–2254, mar 2010. [Online]. Available: https://dl.acm.org/doi/abs/10.1016/j.eswa.2009.07.048

List of Figures



Counts	
Words	4,393
Characters	26,690
Paragraphs	163
Sentences	248
Averages	
Sentences per Paragraph	2.7
Words per Sentence	16.1
Characters per Word	5.1
Readability	
Flesch Reading Ease	43.0
Flesch-Kincaid Grade Level	11.2
Passive Sentences	19.7%

Figure 6.1: Readability Statistics