Instituto Universitário de Lisboa

University Institute of Lisbon
Department of Information Science and Technology

Automatization of Incident Resolution

Jorge Tafarel Morais Costa

A Dissertation presented in partial fulfillment of the Requirements for the Degree of
Master in Computer Engineering – Information Systems and Knowledge Management

Supervisor:
Dr. Rúben Filipe de Sousa Pereira, Assistant Professor
ISCTE-IUL

Co-Supervisor:
Dr. Ricardo Daniel Santos Faro Marques Ribeiro, Assistant Professor
ISCTE-IUL

October 2019

Resumo

A Gestão de incidentes é um subprocesso chave da Gestão de Serviços de TI em todas as organizações como uma forma de lidar com o volume atual de tickets criados todos os anos. Atualmente, o processo de resolução ainda exige muito trabalho humano. Um grande número de incidentes não são de um problema novo, nunca visto antes, eles já foram resolvidos no passado e sua respetiva resolução foi previamente armazenada em um Sistema de Ticket de Incidentes. A automação de tarefas repetíveis em TI é um elemento importante do Gestão de Serviços e pode ter um impacto considerável em uma organização.

Usando um grande conjunto de dados reais de tickets de incidentes, esta dissertação explora um método para propor automaticamente uma resolução adequada para um novo ticket usando textos de resolução de tickets anteriores. Em sua essência, o método usa aprendizado de máquina, análise de linguagem natural, recuperação de informações e mineração. O método proposto explora modelos de aprendizagem automática como SVM, Regressão Logística, arquitetura de algumas redes neurais e mais, para prever uma categoria de resolução de incidentes para um novo ticket e um módulo para extrair automaticamente ações de resolução de tickets usando padrões de classes gramaticais.

Nas experiências realizados, 31% a 41% dos tickets de um conjunto de testes foram considerados como resolvidos pelo método proposto, que considerando o volume anual de tickets representa uma quantidade significativa de mão de obra e recursos que poderiam ser economizados.

Palavras-chave: Gestão de Incidentes, Aprendizagem Automática, Processamento da Língua Natural, Processamento de Texto, Recuperação de Informação

Abstract

Incident management is a key IT Service Management sub process in every organization as a way to deal with the current volume of tickets created every year. Currently, the resolution process is still extremely human labor intensive. A large number of incidents are not from a new, never seen before problem, they have already been solved in the past and their respective resolution have been previously stored in an Incident Ticket System. Automation of repeatable tasks in IT is an important element of service management and can have a considerable impact in an organization.

Using a large real-world database of incident tickets, this dissertation explores a method to automatically propose a suitable resolution for a new ticket using previous tickets' resolution texts. At its core, the method uses machine learning, natural language parsing, information retrieval and mining. The proposed method explores machine learning models like SVM, Logistic Regression, some neural networks architecture and more, to predict an incident resolution category for a new ticket and a module to automatically retrieve resolution action phrases from tickets using part-of-speech pattern matching.

In the experiments performed, 31% to 41% of the tickets from a test set was considered as solved by the proposed method, which considering the yearly volume of tickets represents a significant amount of manpower and resources that could be saved.

Keywords: Incident Management, Machine Leaning, Natural Language Processing, Text Mining, Information Retrieval

Acknowledgements

I would like to express my deepest appreciation to my supervisors Rúben Pereira and Ricardo Ribeiro for their patience and amazing support and guidance during this last year. Also, I would like to thank my family, my dog, friends and colleagues that directly or indirectly emotionally supported and gave me encouragement throughout this journey.

Contents

Resumo ... ii
Abstract ... iv
Acknowledgements ... vi
Contents ... viii
List of Tables ... x
List of Figures ... xii
Abbreviations ... xiv
Chapter 1 – Introduction ... 1
1.1. Motivation ... 2
1.2. Research methodology ... 3
1.2.1. Problem Identification and Motivation ... 4
1.2.2. Define the Objective for a Solution ... 4
1.2.3. Design ... 4
1.2.3.1. The ticket fields ... 4
1.2.3.2. Short and Full Description ... 5
1.2.3.3. Ticket Language ... 6
1.2.3.4. Incident and Resolution Category ... 6
1.2.3.5. Ticket Resolution ... 8
1.2.4. Development and Demonstration ... 8
1.2.5. Evaluation and Communication ... 9
1.3. Structure ... 9
Chapter 2 – Theoretical Background ... 11
2.1. IT Service Management ... 11
2.2. Incident Management ... 12
2.3. Data and Text Mining ... 13
2.4. Text Classification ... 14
Chapter 3 – Related work ... 17
3.1. Text Categorization in Incident Management ... 17
3.2. Resolution Recommendation in Incident Management ... 18
Chapter 4 – Incident Resolution Category Prediction ... 25
4.1. Design ... 25
4.1.1. Data Selection and Preparation ... 25
4.1.1.1. Language and Ticket Selection ... 25
4.1.1.2. Entries with Empty Fields ... 26
4.1.1.3. Text Preprocessing and Transformation ... 26

4.1.1.4. Document Representation ... 27
4.1.2. Categorization Proposal ... 27
4.2. Development and Demonstration ... 29
Chapter 5 – Incident Resolution Recommendation ... 33
5.1. Design ... 33
5.2. Development ... 34
5.2.1. Step 2 - Incident Tickets Similarity ... 34
5.2.2. Step 3 - Resolution Actions Extraction ... 35
5.3. Demonstration ... 36
Chapter 6 – Evaluation ... 39
Chapter 7 – Conclusion ... 41
7.1. Research Limitation ... 42
7.2. Future Work ... 43
Bibliography ... 45

List of Tables

Table 1 - Dataset ticket fields 5
Table 2 - Word and character count for short and full description.................................. 5
Table 3 – Dataset language distribution 6
Table 4 - Incident Category distribution... 7
Table 5 - Resolution Category distribution 7
Table 6 - Word and character count for the resolution field 8
Table 7 - IM processes by ITIL 13
Table 8 - Studies Index 19
Table 9 - Studies Concept Matrix.. 20
Table 10 - Prediction results between nominal attribute vs short vs full description..... 30
Table 11 - Prediction results with stop-words removal.. 31
Table 12 - Prediction results with application of lemmatization.................................... 31
Table 13 - Prediction results with stop-words removal and application of lemmatization 32
Table 14 - Sample of actions POS patterns 36
Table 15 - Experiment accuracy results for solving tickets with the actions extraction module 37
Table 16 - Resolution time comparation between manual and proposed method by accuracy 39

List of Figures

Figure 1 - Design science research process (DSRP) model example (Peffers et al., 2006) 3
Figure 2 - Applied DSR guidelines (adapted from Figure 1) 3
Figure 3 - Occurrence correlation heatmap between the Incident Category and Incident Resolution Category 7
Figure 4 - Proposed approach workflow for automating the IM process 9
Figure 5 - Flowchart of the text classification process with the state-of-the-art elements (Mirończuk & Protasiewicz, 2018). 14
Figure 6 - Data Selection and Preparation Workflow 25
Figure 7 - Categorization Proposal Workflow 28
Figure 8 - Resolution Actions Extraction WorkFlow 33
Figure 9 - POS pattern mining process workflow 35

Abbreviations

CBR – **C**ase-**B**ased **R**easoning

CI – **C**onfiguration **I**tem

CNN – **C**onvolutional **N**eural **N**etwork

DM – **D**ata **M**ining

DSR – **D**esign **S**cience **R**esearch

IE – **I**nformation **E**xtraction

IM – **I**ncident **M**anagement

IS – **I**nformation **S**ystems

IT **–** **I**nformation **T**echnology

ITIL – **I**nformation **T**echnology **I**nfrastructure **L**ibrary

ITS – **I**ncident **T**icket **S**ystem

ITSM – **I**nformation **T**echnology **S**ervice **M**anagement

LSTM – **L**ong **S**hort-**T**erm **M**emory

ML – **M**achine **L**earning

MNN **M**ultinomial **N**aïve **B**ayes

NLP – **I**nformation **T**echnology **I**nfrastructure **L**ibrary

NN – **N**eural **N**etwork

POS – **P**art-**O**f-**S**peech

SNN – **S**oftmax **N**eural **N**etwork

TF-IDF – **T**erm **F**requency **–** **I**nverse **D**ocument **F**requency

TM – **T**ext **M**ining

Chapter 1 – Introduction

Currently, organizations spend a great amount of resources to keep their IT resources incident free and running, since there are very few areas that do not depend directly or indirectly on software systems. Therefore, they rely on IT service management (ITSM) processes to quickly detect, process and resolve incoming incidents to achieve that goal.

Implementing a set of ITSM uniform processes (such as Incident Management (IM), Change Management, etc.) allows the delivery of IT services consistently within a single IT organization and also across many IT organizations (Huttermann, 2012)

Several IT frameworks exist to guide and support organizations during ITSM implementation. The Information Technology Infrastructure Library (ITIL), one of the most widely adopted (Iden & Eikebrokk, 2013; Jan & Li, 2016), is the standard for best practices in managing IT services that provides infrastructure, development, and operations for identifying, planning, delivering, and supporting the IT services to a business, and IM is one of the main process it provides (Salah, Maciá-Fernández, Díaz-Verdejo, & Sánchez-Casado, 2016).

The IM process is responsible for restoring the normal service and operations as quickly and effective as possible and minimize the impact of incidents on the corporation as a whole (Yun, Lan, & Han, 2017). Organizations are adopting software tools called Incident Ticket System (ITS) to support the teams responsible for the IM process. An ITS that follows the ITIL framework practices provides a positive effect on the efficiency of the IM process, which in turn improves and increases companies revenue (Silva, Pereira, & Ribeiro, 2018).

Timely resolution of incoming tickets is essential to achieve availability objectives (Gupta, Prasad, & Mohania, 2008b). This has created the desire for the IT industries to automate their processes and workloads, and the application of machine learning (ML) algorithms allows the automation of the repeated tasks performed in the IT industry, and by adopting it, it is possible to benefit from reduced costs and resources in delivering the IT services (Krishnan & Ravindran, 2017).

1.1. Motivation

Problem resolution is a key issue in the IT service industry (Chen, Tao, Yan, Anerousis, & Shao, 2010) and it is still difficult for large enterprises to guarantee the service quality of IM process because of the difficulty in handling frequent incidents timely, even though ITSM standard process have already been established (Zhao & Yang, 2013).

Conventionally, the IM process is largely manual, error prone and time consuming, especially in the resolution step (Gupta, Prasad, & Mohania, 2008a). For every ticket generated the process is to have someone analyze and try to resolve the incident through personal expertise and it is the same to the next incoming tickets. However, in many cases this process is not entirely systematic and may be incoherent and inefficient (Salah et al., 2016).

It is critical for effective IM to identify tickets which are redundant or potentially have the same root cause (Ghiţ, Iosup, & Epema, 2013). Many incidents are not new – they involve dealing with something that has happened before and may as well happen again or might appear in groups addressing the same problem with the creation of multiple tickets that are related to the same incident (Salah et al., 2016).

Usually a database of historic incidents, stored as tickets on the ITS, and their corresponding resolutions actions are maintained. The search for the correlation between tickets to find the common solution for an issue is an expensive process in terms of manual labor and productivity (Ghiţ et al., 2013).

The objective of this work to is attempt to move to an automatic system that generates a resolution using the similarity of the incoming incident to other tickets by analyzing the data associated with the tickets. The nature of the tickets containing unstructured data in the form of free text such as the incident and resolution description demands the use of appropriate technologies to achieve the goal of automation. Information Extraction (IE), Text Mining (TM) and Natural Language Processing (NLP) are the tools that allows the processing, analysis and knowledge extraction of the tickets textual data (C. C. Aggarwal & Zhai, 2012), and coupled with machine learning (ML) techniques, that knowledge can be further used for pattern recognition (“Pattern Recognition and Machine Learning,” 2007) and to implement an incident prediction model that helps achieve the optimization of the resolution process.

1.2. Research methodology

For this project, the Design Science Research (DSR) was adopted as research methodology (Peffers et al., 2006). DSR methodology is an outcome-based IT research methodology and according to Vaishnavi et al. (2004) it provides a set of synthetic, analytical techniques and perspectives (complementing positivist, interpretive, and critical perspectives) for performing research in IS. It is used to create and evaluate IT artifacts intended to solve identified organizational problems (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). Peffers et al. (2006) described the DSR process model consisting of six activities in a nominal sequence described and presented in Figure 1.

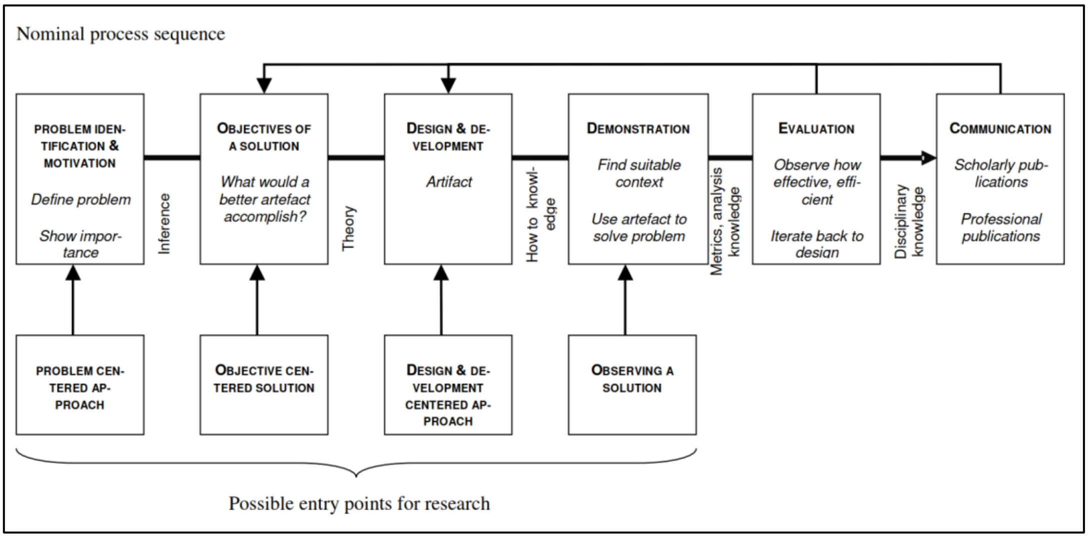

Figure 1 - Design science research process (DSRP) model example (Peffers et al., 2006)

Figure 2 presents the steps to follow along this dissertation.

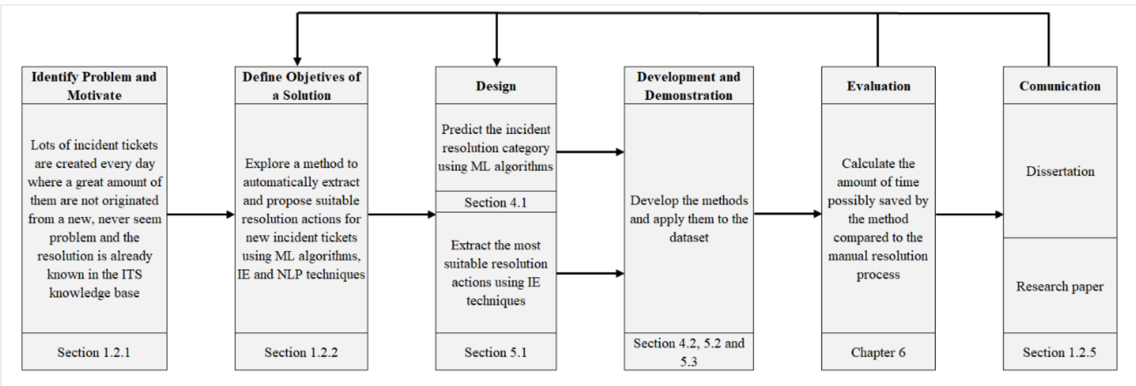

Figure 2 - Applied DSR guidelines (adapted from Figure 1)

1.2.1. Problem Identification and Motivation

The response to incidents are mostly operated manually which is time consuming, and it is of too much dependency when relying only on the subjective personal experience of a service agent (Yun et al., 2017). It should also be considered the high volume of incoming incidents, and the incorrect assignments to a resolver group, which affects the incident route negatively that demands more resources, and leads to wasted time (Shao, Chen, Tao, Yan, & Anerousis, 2008).

In addition, a lot of incidents are not originated from a new, never seen problem, making a resolver spend time investigating on a repeating problem which solution is already known in the ITS knowledge base.

Considering all this together, the incident resolution process is time expensive and can be optimized and adding some automation to these processes results in reduced waste of time and resources.

1.2.2. Define the Objective for a Solution

Grounded on the previous statements, this research aims to explore a method to automatically extract and propose suitable resolution actions for new incident tickets using ML algorithms, IE and NLP techniques.

1.2.3. Design

For this study it is used a real dataset provided by an IT department from a real multinational company that due to privacy questions cannot be mentioned along the work. The IT department is responsible for supporting and managing the global IT resources of the entire organization that are used by more than 350k employee and their incident management system is responsible for the handling of all types of tickets from all the departments inside of the organization so the data contains a diverse range of knowledge domains.

1.2.3.1. The ticket fields

The dataset under study contains 1.2 million incident tickets covering the 2018 year. It contains a large number of fields, with 24 attributes associated such as incident category, subcategory, short description and description, type, assignment group, contact source, department, location, resolution category, etc. The incident short description, full description and resolution stand out as the only ones that are in the form of unstructured text and the rest being categorical. From the categorical ones, most are redundant and remain unused in the IM process. In Table 1 it is possible to find all the fields in the dataset.

Table 1 - Dataset ticket fields

Tickets fields	
Unstructured	
Short Description	Resolution Notes
Full Description	
Structured	
Number	User Category
State	Business Duration
Created By	User Scope
Created On	Department
Assigned To	Location
Assignment Group	Severity
Incident Category	Contact Source
Incident Sub-Category	User Impact
Incident Type	Resolution Category
Resolved By	Resolved At
Resolver Group	

1.2.3.2. Short and Full Description

The short description and full description are two of the most important fields in the tickets. Essentially, the full description is a longer and more detailed description about the incident containing a more throughout explanation about the problem, whereas the short description is a short summary about the incident. Table 2 shows the minimum, maximum and average word count and character count for the short and full description. There are tickets with only one word in one of the descriptions. Concerning the maximum word count for the full description, a few tickets had whole emails conversations or other forms of external reports that were pasted to the field.

Table 2 - Word and character count for short and full description

	Short Description		Full Description	
	Characters	**Words**	**Characters**	**Words**
Minimum	1	1	1	1
Maximum	160	35	31953	5020
Average	42.94	6.92	323.80	45.28

1.2.3.3. Ticket Language

Since the company is a multinational, the dataset contains tickets in a variety of languages, such as English, Portuguese, German, Spanish, French, etc., and since there is no with the language of the ticket so it was necessary to use an additional tool to automatically detect the ticket language. Table 3 presents the dataset language distribution results from the automatic language detection process.

Table 3 – Dataset language distribution

Language	
English	82.5%
German	8.8%
Portuguese	2.9%
Spanish	1.04%
French	0.5%
Others	4.26%

1.2.3.4. Incident and Resolution Category

The incident category refers to the problem domain inside of the organization, is selected by the user or the support staff in the ticket creation process and it is used for routing, to deliver the ticket to the service agent with the correct knowledge expertise for the ticket problem.

Once a service agent has solved the incident, he selects the resolution category. In the IM, process this category is mostly used for reporting and statistics. Table 4 and Table 5 shows the categories distribution in the dataset, and as it is possible to see, the data is heavily imbalanced. Figure 3 shows the occurrence correlation between the two categories. The presented values are normalized. The correlation is not strong and does not show a clear separation between the categories, and, as expected, the most common resolution categories ("Information/Advice", "Request" and "Configuration") are the ones with a stronger presence across the dataset.

The "Security and Access" incident category and the "Request" resolution category presents the stronger correlation in the dataset which is not surprising since the request for access of any type, especially in the IT environment (request for password, access to a system or software, etc.), is a common occurrence inside of an organization. The "Information/Advice", the most frequent resolution category, not surprisingly share a relatively stronger correlation with most frequent incident category which can be credited to the simple fact that it has more samples.

The rest of the categories have a fairly uniform distribution.

Table 4 - Incident Category distribution

Incident Category (%)	
Application	20.1
Workplace	16.8
Collaboration	15.4
Security and Access	13.1
Software	11.0
Support	8.5
Network	4.9
Hosting Services	4.5
Output Management	2.8
IT HR Services	2.4

Table 5 - Resolution Category distribution

Resolution category (%)	
Information/Advice given	35.5
Request	29.2
Configuration	17.4
Installation	5.1
Other	3.8
Security	2.3
Complaint	1.9
Hardware	1.6
Software	1.6
Data	1.2

Figure 3 - Occurrence correlation heatmap between the Incident Category and Incident Resolution Category

Silva et al. (2018) has already approached the incident category by creating a system predict the incident category for tickets routing assignments. This research will focus only on the incident resolution category since it is the one covering the resolution domain.

1.2.3.5. Ticket Resolution

The resolution field is also a natural language text field registered by the service agent in the end of the incident lifecycle containing information about the incident resolution such as the action's steps taken, additional information about the problem or both. This field presents a high amount of noise, since it is the last to field to be filled by the agents, they mostly do it in a rush and carelessly. Table 6 presents the minimum, maximum and average word count and character count for the resolution field. Few tickets from the dataset did not contain the resolution but not in a significant number. Similar to the short and full description, some tickets present entire email conversations or reports that were pasted by the agent to the field.

Table 6 - Word and character count for the resolution field

	Resolution	
	Characters	**Words**
Minimum	1	0
Maximum	20214	4010
Average	190.78	33.81

1.2.4. Development and Demonstration

To achieve the goal of this study, this research proposes the development of a method capable to automatically propose a suitable resolution for new incoming incidents. To design the artifact, this research follows the two steps bellows:

- First, it is necessary to predict the resolution category. For this, text pre-processing techniques and feature selection should be applied to the dataset to then obtain the best classifier for the resolution category. Here, different ML and TM techniques combinations will be tested and presented, and the one that best predicts the resolution category will be selected. This research will not focus on the prediction of the incident category since it was previously approached by Silva et al. (2018) on a similar dataset.

- Next, this research will then explore the use of Information Extraction (IE) techniques to extract the most suitable resolution actions from the dataset for the new incident ticket.

As presented in Figure 4, with these two steps, and with the research from Silva et al. (2018), the IM is covered as whole.

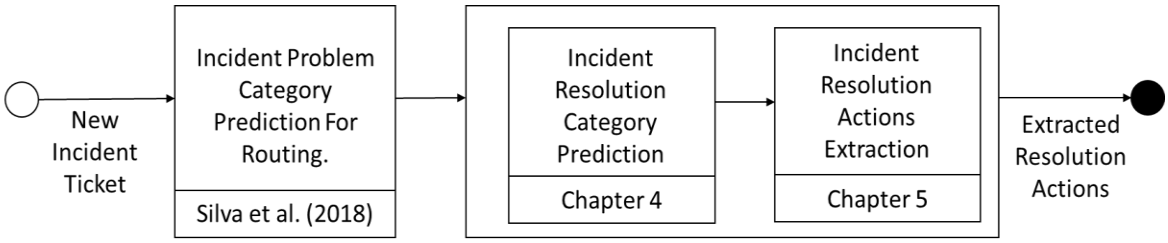

Figure 4 - Proposed approach workflow for automating the IM process

1.2.5. Evaluation and Communication

Regarding research communication, part of this research is presented in one published paper and the whole research is represented by this document.

1.3. Structure

The remainder of this dissertation consists of 6 chapters that are structured as follows.

The second chapter presents the related work, studies related with automating, improving and optimizing the IM process. It describes which approaches and ML algorithms are implemented, and which are the results obtained with the respective approaches.

In the third chapter, is presented the related work with studies about categorization in incident management and the resolution recommendation approaches.

In the fourth chapter, it is presented the design, development and demonstration phase for the incident resolution category prediction.

In the fifth chapter, it is presented the design, development and demonstration phase for the incident resolution recommendation.

In the sixth chapter, it is presented the evaluation of the proposed method.

The seventh chapter is the conclusion of this dissertation document.

Chapter 2 –Theoretical Background

This chapter presents relevant concepts and definitions related to this research. The literature here discussed represents and serves as a guideline to the research development and in order to achieve the defined goals.

2.1. IT Service Management

ITSM is a subset of Service Science that focuses on IT operations such as service delivery and service support (Kang & Zaslavsky, 2010). It refers to the entirety of activities: directed by policies, organized and structured in processes and supporting procedures – that are performed by an organization to design, plan, deliver, operate and control IT services offered to customers and meets the needs of the business. It must be carried out effectively and efficiently and from the business perspective enables organizational high performance and value creation (Kosasi, Prabowo, & Budiastuti, 2017).

Strategic values given by ITSM can be in the form of service deliveries and support effectiveness and efficiency. IT service providers should prioritize IT service quality based on the consumers' expectations. This quality is, therefore, the agreement between IT service providers and the consumers (Lepmets, Cater-Steel, Gacenga, & Ras, 2012). It has two important elements: services support and deliveries. The former refers to ways to gain access of information service availability quickly and completely, while the latter presents services to fulfill managerial needs of the stakeholders in decision making (J. Wan, Zhang, & Wan, 2011).

Referring to ITIL, service supports consist of IM, problem management, change management, exemption management, and configuration management. Service deliveries, on the other hand, include management service levels, finance management, IT continuance management, capacity management, and availability management(S. H. C. Wan & Chan, 2007).

A well-defined ITSM framework results in a better monitoring of processes so that organizations can reach a higher maturity levels enabling a global understanding and a better vision of processes. When the efficiency and productivity of process activities are improved, the organization can develop, maintain and deliver higher quality services, meet business objectives and obtain a higher customer satisfaction (Orta & Ruiz, 2019).

By establishing a set of uniform processes (IM, Change Management, etc.), ITSM enables the delivery of IT services consistently and optimize tactical and strategic IT asset use within a sole IT organization as well as across many IT organizations (multi-nationals, outsourcers, etc.) (Galup, Dattero, Quan, & Conger, 2009).

2.2. Incident Management

The IM process is responsible for managing the lifecycle of all incidents, including any event that disrupts, or could disrupt a service. This includes events which are communicated directly by users, either through the Service Desk or through an interface from Event Management to IM tools (Marcu et al., 2009) .

An incident is defined by ITIL as an unplanned interruption to an IT service or a reduction in the quality of an IT service and is reported by humans or automatically detected and generated by a monitoring system (IBM Tivoli Enterprise Console). The recording of an incident and its nature is saved as a ticket in an ITS, which is a primary tool used by management for tracking and report of ongoing and resolved incidents (Salah, Maciá-Fernández, & Díaz-Verdejo, 2019).

As a key process, IM provides data record of each step-in incident resolution process, verifies resource configuration, management process and its operation quality to achieve service objectives, and provides data for developing service report, service plan, cost accounting as well as service workload assessment (OGC, 2007). Thereafter, IM is involved in the whole lifecycle of ITSM.

Once established, effective IM value is highly visible to the business, and it is therefore easier to demonstrate its value than most areas in Service Operation (OGC, 2007). It has the ability to optimize costs and expenders reducing unplanned labor due to the incidents detection and resolution, resulting in lower downtime for the business. It also aligns the IT activity to the business priorities by highlighting other areas that need attention and potential improvements to services and dynamically allocate resources to provide for to the business needs (Marcu et al., 2009).

The ITIL divides the IM into several steps as shown in Table 7:

Table 7 - IM processes by ITIL

Activity	Description
Incident detection and recording	All incidents must be promptly fully logged with all the information about its nature before they have an impact on users
Classification and initial report	The incident must be assigned to the correct category, so the exact type of incident is documented with the appropriate priority (high, medium, low).
Investigation and diagnosis	An initial evaluation and diagnosis are done to the incident and the proper escalation is applied until the incident is routed to the right analyst able to solve it.
Resolution and recovery	When a potential resolution has been identified, it should be applied and tested.
Closure and tracking	At this point, the resolution is completed and confirmed, the incident is considered closed and the incident process ends.

This study focuses on the resolution process of IM included in the fourth activity.

2.3. Data and Text Mining

The term Data Mining (DM) is a diverse and broadly diversified. Gorunescu (2011) describes it as the science of extracting useful data from large databases. Additionally, DM also deals with analyzing and structuring these data, i.e. preparing them in a meaningful way. It relies on the application of machine learning, pattern recognition, statistics, databases and visualization to solve the problem of information extraction from the databases (Sharma, Sharma, & Dwivedi, 2017). The data can be available in different forms, for example numbers, time or date information or in text form.

One variant and important sub-area of DM is called Text Mining (TM). TM is the discovery and extraction of interesting, non-trivial and high-quality knowledge from text data (Zhong, Li, & Wu, 2012). Typical TM involves tasks and processes such as information retrieval, text classification, clustering, entity relation and event extraction (Kao & Poteet, 2007). Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics. The main objective is, basically, to turn text into data for analysis, via application of Natural Language Processing (NLP) and analytical methods (Agnihotri, Verma, & Tripathi, 2014; Han, Kamber, & Pei, 2012). The TM tasks on which this work mainly focuses are text classification and information retrieval.

2.4. Text Classification

As previously stated, text classification is an important task in TM and predictive analysis with many applications. It the process of automatically classify natural language texts or documents that are unlabeled into a predefined set of semantic categories or classes (Pang & Lee, 2008) by using classifiers. After the text of said documents has been transformed, text classification is done by feeding that data into machine learning algorithms. With exponential growth in the volume of the unstructured data in the Internet, automatic text classification has become more and more important as it helps categorize and organize various mixed documents into different labels of interest with known properties, therefore making the search and retrieval of information a much more efficient process.

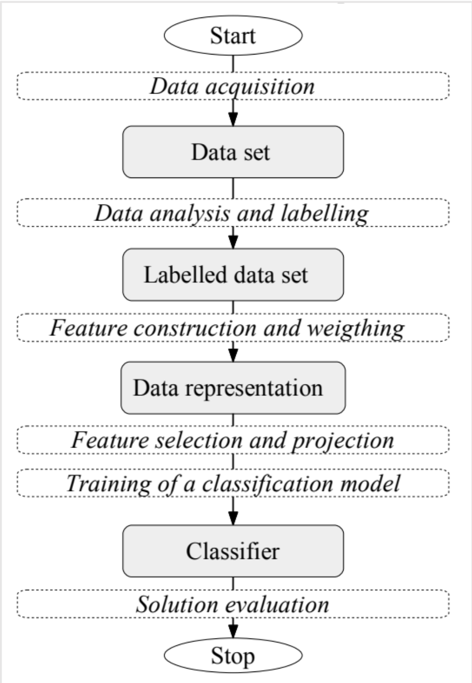

Figure 5 - Flowchart of the text classification process with the state-of-the-art elements (Mirończuk & Protasiewicz, 2018).

Text classification can utilize two groups of learning algorithms: supervised learning and unsupervised learning. In supervised learning, the ML algorithm learns from data comprising of examples (that have already been correctly labeled) that have both input and output values (Bari, Chaouchi, & Jung, 2014), and unsupervised learning in which, by contrast, the data used does not contain any information regarding the output.

Since the dataset is composed of labelled ticket data, this research will only approach supervised learning classification.

Figure 5 presents the six elements included in the classification baseline process identified by Mirończuk and Protasiewicz (2018).

After developing a classifier, it is always necessary to measure the classifier performance by using classification evaluation metrics. In general, an evaluation metric can be described as the tool that measures the performance of classifier. “Different metrics evaluate different characteristics of the classifier induced by the classification algorithm” (Hossin & Sulaiman, 2015). The performance measure mostly used for classification problems is the overall accuracy or, equivalently, the total error.

The four most common metrics are the following (Hossin & Sulaiman, 2015):

- Accuracy: It measures the ratio of correct predictions over the total number of instances evaluated.

$$(1)\ Accuracy = \frac{True\ Positives\ (TP) + True\ Negatives\ (TN)}{TP + TN + False\ Positives (FP) + False\ Negatives\ (FN)}$$

- Precision: It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$(2)\ Precision\ = \frac{TP}{TP+FP}$$

- Recall: It is the number of correct positive results divided by the number of all relevant samples. It is used to measure the fraction of positive patterns that are correctly classified.

$$(3)\ Recall\ = \frac{TP}{TP+FN}$$

- F-Score: It is the harmonic average of precision and recall.

$$(4)\ F-score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

One aspect to consider when choosing a metric is the data balance nature. Classification problems with imbalanced distributions between classes can raise issues and are considered difficult problems (Daskalaki, Kopanas, & Avouris, 2006), however, they are not uncommon in data mining projects. Problems like fraud detection (Ngai, Hu, Wong, Chen, & Sun, 2011) and the diagnosis of rare diseases (Laurikkala, 2001) are typical examples. In these cases, is not wise to rely strictly on accuracy (Hossin & Sulaiman, 2015).

It is also important to considerate the training approach. Previous researches (Weiss & Provost, 2001) have shown that training with the "natural" distribution does not always result in the best classifier; hence, new training datasets should be constructed to assist the training process. Especially when the minority cases are rare, a training set "enriched" with minority examples is needed in order for the induction algorithms to create classifiers with strong predictive capabilities for all classes by either eliminating cases in the majority classes until the desired distribution in achieved (under-sampling) or by artificially duplicating the cases in the minority distribution (oversampling) (Daskalaki et al., 2006).

Chapter 3 – Related work

This chapter details some of the approaches found among the literature that relates to this study. It is divided in two sub sections related to the two previously identified tasks in this study: Text Categorization in Incident Management, related to the prediction of the resolution category and the literature approaching incident resolution recommendation.

3.1. Text Categorization in Incident Management

There have been some studies about the use of classification in the domain of incident tickets. Although these studies are using machine learning to perform classification, all of them are related to the prediction of the incident category of domain used for routing purposes. The most common used algorithms were Naïve Bayes, SVM, K-Nearest Neighbours (KNN), and logistic regression (logit). Following, is a more detailed analysis of some of these studies. No studies were found with the intention to predict a category related to the ticket resolution similar to this research.

Silva et al. (2018) introduces a module to automatically categorize incident tickets turning the responsible teams for incident management more productive and reducing the time wasted on incident ticket route and reducing the amount of errors on incident categorization. The authors compared results between SVM and KNN classifiers applied to the ticket description that were represented by using term frequency-inverse document frequency (TF-IDF). The accuracy results obtained was of 89%, approximately, on a dataset similar to the one used by this study.

Son et al. (2014) also approached the incident ticket category classification by resorting to two algorithms such as Multinomial Naïve Bayes (MNB) and Softmax Regression Neural Network (SNN). On both approaches, ticket subjects were used to create an input word list along with a manual word group list to enhance accuracy. The text mining algorithms used the input word list to select input words in the tickets. The implementations were tested with a dataset composed of 7042 for training tickets and 717 reserved for testing. MNB achieved slightly better results with the best overall accuracy at 85.8% than SNN at 84%.

Altintas & Tantug (2014) proposes an extension to ITS for auto-addressing the incident ticket to the relevant expert in support team consisting of a two phase classification process. The first phase intent is to detect the related category of the ticket which is directly related to the department of the issue, whereas the second phase tries to determine the related subcategory or unit under the specified category based on a pre-defined threshold. In the feature extraction step, the authors adopted the bag of word approach using TF-IDF. The experiments were conducted on a dataset consisting of approximately ten thousand issue tickets in Turkish by exploring four algorithms, namely SVM, KNN, decision trees (DT) and Naïve Bayes. The performance varied directly related to the machine learning algorithm, the weighting method and the dataset, achieving the best accuracy of 86% with the SVM.

Al-Hawari & Barham (2019) introduce a help desk system that acts as a single point of contact between users and the IT staff that utilizes an accurate ticket classification machine learning model to associate a help desk ticket with its correct service from the start. The experimental results showed that including the ticket comments and description in the training data was one of the main factors that enhanced the model prediction accuracy. The J48 (Tree-based), Decision Table (Rule-based), NaiveBayes (Bayes-based) and SMO (SVM-based) algorithm were used. Applying the TF-IDF feature vectorization, the best results were achieved by SMO reaching an accuracy value of 81.4%.

3.2. Resolution Recommendation in Incident Management

Over the years, there have been different attempts to automate the IM resolution process, with different techniques applied. Table 8 and

Table *9* presents a summary about of the approaches followed by a more detailed explanation about each study. Most of the attempts were based on clustering and correlation algorithms to provide the closest ticket based on similarity.

In the task of analyzing the raw textual ticket data written in natural language, some studies attempt to create and process the knowledge domain base using ontology extraction (S8, S2, S6). Some (S1, S9) have used TF-IDF to create the knowledge domain by extracting the word frequency score for each domain. The frequency of specific words may offer precious information about a specific domain, so mapping and using the weight of the most frequent terms might help the discovery of important keywords for the individual domains. Some authors applied PoS tagging to extract the nouns in the ticket. It was identified that the most important words that contain the most information about a ticket were in the form of nouns (e.g. disk, card, laptop, etc.). The use of normalization and n-grams (S1, S5, S6) to tackle the ambiguity problem showed positive results in the extraction of semantic domain knowledge.

Table 8 - Studies Index

Studies	Index	Quadrille	Country	Year
(Gupta et al., 2008a)	S1	B - ERA A2 - Qualis	India	2008
(Kang & Zaslavsky, 2010)	S2	B - ERA A1 - Qualis	China	2010
(Li & Zhan, 2012)	S3	A - ERA B1 - Qualis	China	2012
(Tang, Li, Shwartz, & Grabarnik, 2013)	S4	B1 - Qualis	USA	2013
(V. Aggarwal, Agarwal, Dasgupta, Sridhara, & Vijay, 2016)	S5	A - ERA B1 - Qualis	Australia/ USA	2016
(Roy, Yan, Budhiraja, & Lim, 2016)	S6	A - ERA B1 - Qualis	India/South Korea	2016
(Zhou et al., 2016)	S7	Q1 - Scimagojr	USA	2016
(Wang et al., 2017)	S8	A - ERA B1 - Qualis	USA	2017
(Yun et al., 2017)	S9	-	China	2017

The KNN algorithm was mostly used to obtain the correlation of new incoming tickets to their closest possible solution (S4, S1, S6, S7). In (S4) the authors compare the result from tradition KNN, weighted KNN and their divide and fusion algorithms based on KNN. Their algorithms achieved a higher accuracy than traditional KNN with a lower average penalty for false resolution. They tested 3 datasets where they reached accuracy up to 83%.

Other studies experimented Neural Networks for the task. In (S9) a back propagation neural network algorithm (BP NN) is used to develop a classifier. It uses the learning mechanism of the backward propagation to correct the weights in the neural network, and finally achieve the goal of output the correct result. By varying the number of results presented as possible solution they were able to achieve 93% accuracy. The authors in (S3) compare their BP NN to the Mixed-Integer Programming (MIP) approach. The classification accuracy of their NN is fluctuant but it gradually increases with the growing training set and achieve a higher accuracy than the MIP approach, achieving up to 88% accuracy.

Table 9 - Studies Concept Matrix

Concepts vs studies		**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**
ML algorithms	Ontology		**X**				**X**		**X**	
	KNN/K-means	**X**			**X**		**X**	**X**		
	NN			**X**						**X**
NLP Techniques	Stop-Words					**X**	**X**			**X**
	N-gram					**X**	**X**			
	PoS	**X**		**X**		**X**	**X**		**X**	**X**
	Tokenization	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**
	Normalization	**X**	**X**	**X**		**X**	**X**			
	NER	**X**					**X**			
	Tf-Idf	**X**		**X**						**X**
Similarity Algorithms	Edit distance			**X**		**X**				
	Dice’s coefficient		**X**							
	Jaccard		**X**		**X**		**X**	**X**	**X**	
	Cosine						**X**			**X**
	Pearson correlation coefficient									**X**
Topic Model	LDA							**X**		

Wang et al (2017) proposed a framework where the knowledge base is modeled using an ontology to attempt problem inference by processing ticketing information. It outlines an integrated solution that uses obtained knowledge to optimize problem resolution in a learning-loop system and addresses the problem with unstructured fields and the ambiguity brought by the free-form text on the tickets. The framework is based on a phrase extraction stage (information retrieval, NLP and TM), knowledge construction using an ontology model and a recommendation/ticket resolution stage where incoming ticket is first processed by a Class Tagger module of an information inference component. By inputting the tagged ticket, the recommendation component provides the list of the most relevant resolutions. The ticket is then archived into the historical ticket database, and the newly obtained domain expertise can be used to enrich the knowledge base. They tested the framework on 22,423 tickets from IBM Global Services that consists of both structured fields and unstructured free-form text field covering a 3-month period. The test set was manually tagged by experts to build the ground truth that was then compared to the framework solution. Precision, recall, F1 score, and accuracy were close to 1 but it was due to the small number of instances of the test set.

Tang et al. (2013) studies the possibility to run a modern service infrastructure management in a fully automated operation environment with automatic monitoring software systems that capture and generate incident tickets working together with automated problem resolution. The authors suggest a recommendation systems approach to the resolution of event tickets and propose two resolution recommendation algorithms ("Divide" and "Fusion") based on the weighted KNN algorithm framework with additional penalty incorporated to avoid misleading resolutions. The authors experimented with tickets captured by IBM Tivoli Monitoring and their algorithms outperform the traditional KNN and weighted KNN in all their case studies.

Gupta et al. (2008) presented a technique to identify a failing component by integrating text specified in the problem ticket with structured data stored in CMDB database along with incident classification. The scope was reduced to customer reported incidents only. They implemented a component with an automatic identification of keywords with part-of-speech filtering and word normalization, search over CMDB using search context, and limiting the search scope using directed navigation. The dataset used consisted of 192,000 objects and more than 150,000 relationships among them in an IBM network.

Yun et al. (2017) aim to use data mining technology to build an automatic decision-making model in order to automate the IM process. The model created is composed by two parts: a machine learning classifier used to predict the classification of the input incident and provide context to search similar historical incident; a search engine used to search the historical incidents similar to the input incidents using similarity algorithms and NN. When receiving an incident request, the framework can identify the possible failing CIs based on historical data, and predict the incident classification, and then retrieve the knowledge base of incidents to return the results of reference value. They used the Symian (Bartolini, Stefanelli, & Tortonesi, 2008) tool to construct a training dataset of 2000 incidents and test dataset of 1000 incidents in the simulation process, based on an enterprise real historical data. Their proposed method improved the accuracy rate by 67% achieving up to 93% accuracy in one of the tests.

Kang et al. (2010) proposes a knowledge-rich similarity measure for improving the ITSM resolution process. Based on their similarity measure, the most k-top similar tickets are retrieved for a new one, and then the solutions contained in the retrieved tickets can be used to help to generate the appropriate solution for the new incident. The measure considers the classification, workgroup and description of the ticket exploiting semantic knowledge described in taxonomies that represent the relationships among the entries. The final measure value is a union of the similarity measure of those three components similarity. For the study, they compare four similarity measures using different taxonomies combinations. Evaluation was performed using a real dataset based on off-line analysis using precision and recall focusing on empirical evaluation on the returns of relevant and irrelevant incident cases. Three PhD students with strong background of IT were used for assessment and relevance judgments by comparing solution descriptions between a query ticket and each of k-top incident cases retrieved and the best results were obtained by using all three taxonomies.

Aggarwal et al. (2016) presented a system called ReAct which help the service agents to identify set of possible actions and resolve the ticket issue and uses visualization to help user choose the most suitable option. The authors implemented a NLP engine for action extraction, semantic similarity calculations, summarizing verbose resolution text into brief action phrases and sequence mining on action sequences. The resulting data was used to build a knowledge database. The next step was a series of prediction services based on action-set, next action and attribute prediction. They also built a UI layer for interactive navigation through the whole pipeline in a step by step manner and displays the constructed action sequence, recommended set of actions and attributes.

Roy et el. (2016) propose an automated method based on historic incident knowledge for recovering resolutions for new incident based on ontology-driven clustering of tickets, unsupervised learning and KNN search. They used the incident description to extract and build an ontology augmented with concepts from WordNet. This ontology drives the clustering of incidents using unsupervised learning in which a ticket is modeled as a feature vector of keywords/concepts. Once a new ticket appears, it is parsed and mapped to the set of concepts in the ontology created beforehand, and thus generating the feature vector for it based on these concepts. The authors then compute the similarity distance of the new incident to the dataset and then place it with its closest cluster. By using KNN, a couple of nearest tickets are chosen, and their corresponding resolutions are published as the recovered resolutions for the new ticket. The approach was tested on tickets for and application maintenance system of Infosys, Ltd on the retail business. In evaluation they compared each of the proposed resolutions with the actual resolution and compute the semantic similarity between them and if most similar resolution exceeds a predefined threshold then it is considered a matching resolution. Overall there was an average of 48% between the suggestions and the actual resolution.

Li et al. (2012), propose a complete solution to automate incident management process with the incident description as the start and measuring the most similar incidents as the end. The system is divided into five components: Keyword Extraction and Normalization, CI Indexing, Deploy Architecture Calculation, Training Element Signature, and Machine Learning. The keyword extractor automatically extracts the normalized keywords from the new ticket description that are then used to search the CI by the CI indexing and the deploy architecture calculation component is responsible for extracting the architecture influenced by incident. All the results will be fed to the ML component after signing. The learning process is based on BP NN. For evaluation, the authors chose 1000 incident pairs as test set, and after training the NN, they compared the results with manmade feedback to obtain accuracy.

Zhou et al. (2016), tried to improve the similarity measure used in KNN by utilizing both the incident and resolution information in historical tickets via a topic-level feature extraction using the LDA (Latent Dirichlet Allocation) model, which can extract hidden topics and then encode monitoring tickets using topic levels features. When resolution categories are available, they propose to learn a more effective similarity measure using metric learning. They authors used Inference feature vectors using the trained LDA model for both incoming events and historical monitoring tickets and then monitoring tickets can be encoded as feature vectors and the cosine similarity can then be applied to measure their similarities. For evaluation they used 1000 labeled tickets with resolution categories from IBM Global Services and their algorithm achieved better results compared to the KNN and weight KNN approach.

Chapter 4 – Incident Resolution Category Prediction

This chapter presents the design, development and demonstration sections to create the artifact for the prediction of the incident resolution category.

4.1. Design

This section is divided as follows:

- **Section 4.1.1:** Describes the combinations of all the necessary steps to prepare and clean the data to build the final dataset, including the text preprocessing methods applied.

- **Section 4.1.2:** Describes the proposed categorization approach

4.1.1. Data Selection and Preparation

Because of the size and nature of the dataset and in order to achieve better classification results, it is necessary to apply preprocessing and data selection. The process workflow is presented in Figure 6.

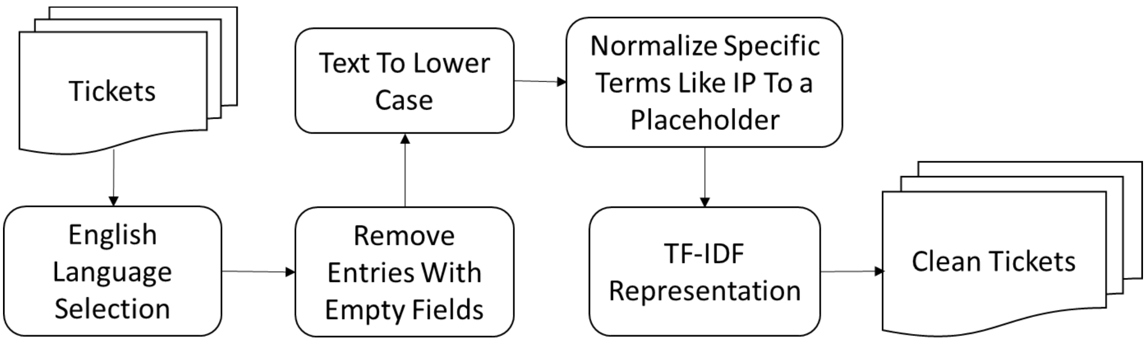

Figure 6 - Data Selection and Preparation Workflow

4.1.1.1. Language and Ticket Selection

As previously stated, several languages were identified in the dataset ticket records. It was decided that only tickets in the English language will be used since it is the most common language composing 82,5% of the dataset.

This step was also used to filter out the tickets created and closed by the automatic monitoring system. This system is responsible to create tickets when it detects failures and problems in another supervised system, like a failure in the network, for example.

These types of tickets have no contribution for this study, so they are discarded. Their resolutions are generally a highly technical, automatic generated report placed by the automatic system.

After this step, the dataset ends up with 850 thousand incident tickets in the dataset.

4.1.1.2. Entries with Empty Fields

In an initial analysis, several tickets presented empty values in some important fields especially the incident category and the resolution category as previously stated. For the incident category it was identified that 6,416 tickets did not have an incident category assigned, and for the resolution category, 463,034 tickets did not contain the field, which represents more than half of the dataset. Some employees of the organization were enquired about it, but they could not provide a specific explanation other than the possible neglect and hurry of the agent that closed the ticket, since assigning the resolution category is the last step of the resolution process. After also discarding the tickets with one of the categories missing and the ones where the short or full description were only one word, the resulting dataset contained 350,080 ticket records.

4.1.1.3. Text Preprocessing and Transformation

Text transformation is the conversion of the content of a text so that it can be recognized by a computer, allowing the machine to process and classify it (Man Lan, Chew Lim Tan, Jian Su, & Yue Lu, 2009). It is essential that all data are cleaned in the way that unnecessary complexity and corpus noise in the data does not have a negative influence on the final classification results. It is important to obtain a clean word stream which is better for the learning algorithms (Uysal & Gunal, 2014).

First, case information is removed by transforming all text into its lowercase form. The goal is to reduce the complexity in the data and the number of features that will be created later, since the vocabulary size is reduced in this process.

All IP address, MAC address, email and website URLs are replaced to a matching placeholder, the following are some examples:

192.142.23.12 → ipaddress

00:0a:95:9d:68:16 → macaddress

example@email.com → emailurl

example.com → websiteurl

Additional cleaning is necessary to contractions in the texts by replacing them with their full form, for example: *what's→what is*, *'ve→have*, *n't→ not*, *'re→are*, etc. Any whitespace beyond one space between words and all non-alphanumeric characters including punctuation is removed. Tokenization is also applied, which means splitting the text into a sequence of tokens. This is necessary for the process of feature extraction.

4.1.1.4. Document Representation

After the data preprocessing process an important, step is document representation by representing each document as a feature vector, selecting the terms or tokens that are most relevant to identify a document and removing features that are irrelevant to the classification task, leading to dimensionality reduction of the dataset. Reducing the dimensionality can reduce the noise in the original text collection and thus provide better patterns. The bag of word model is widely used in text mining and information retrieval (Baeza-Yates & Ribeiro-Neto, 1999). Words order is not important, and each word corresponds to a dimension in the resulting data space represented by their frequency. Each document is then transformed into a vector consisting of non-negative values on each dimension.

One way of represent those words that is popular in information retrieval, data mining and in the related work as presented in Chapter 3 –is with term frequency-inverse document frequency (TF-IDF) (Kotu & Deshpande, 2015). TF-IDF weights the frequency of a word or term t in a document d with a factor that discounts its importance with its appearance in the whole document collection (Huang, 2008). The weight increases with the number of times the term occurs but is counterbalance by the frequency of the term in the corpus. The TF-IDF formula is the following:

$$(5)\ tf - idf(d,t) = tf(d,t) \times log\,(\frac{|D|}{df(t)})$$

Where $df(t)$, is the total number of documents in which the term t appears. To create the input data for the classifiers, this work mainly uses TF-IDF vector space and embedding layers for the NN as the standard approach for feature extraction and dimensionality reduction.

4.1.2. Categorization Proposal

For the categorization, this work uses two algorithms. First, using algorithms presented in the related work, more specifically Multinomial Naïve Bayes, Logistic Regression

(LR), SVM and KNN, and second, a neural based approach by using neural networks models such as Long Short Term Memory (LSTM), Bidirectional LSTM (BiLSTM) and Convolutional Neural Networks (CNN) for the classification task. The purpose is to compare the algorithms and models' performance using different TM techniques and verifying which one produces better results. The process workflow is presented in Figure 7.

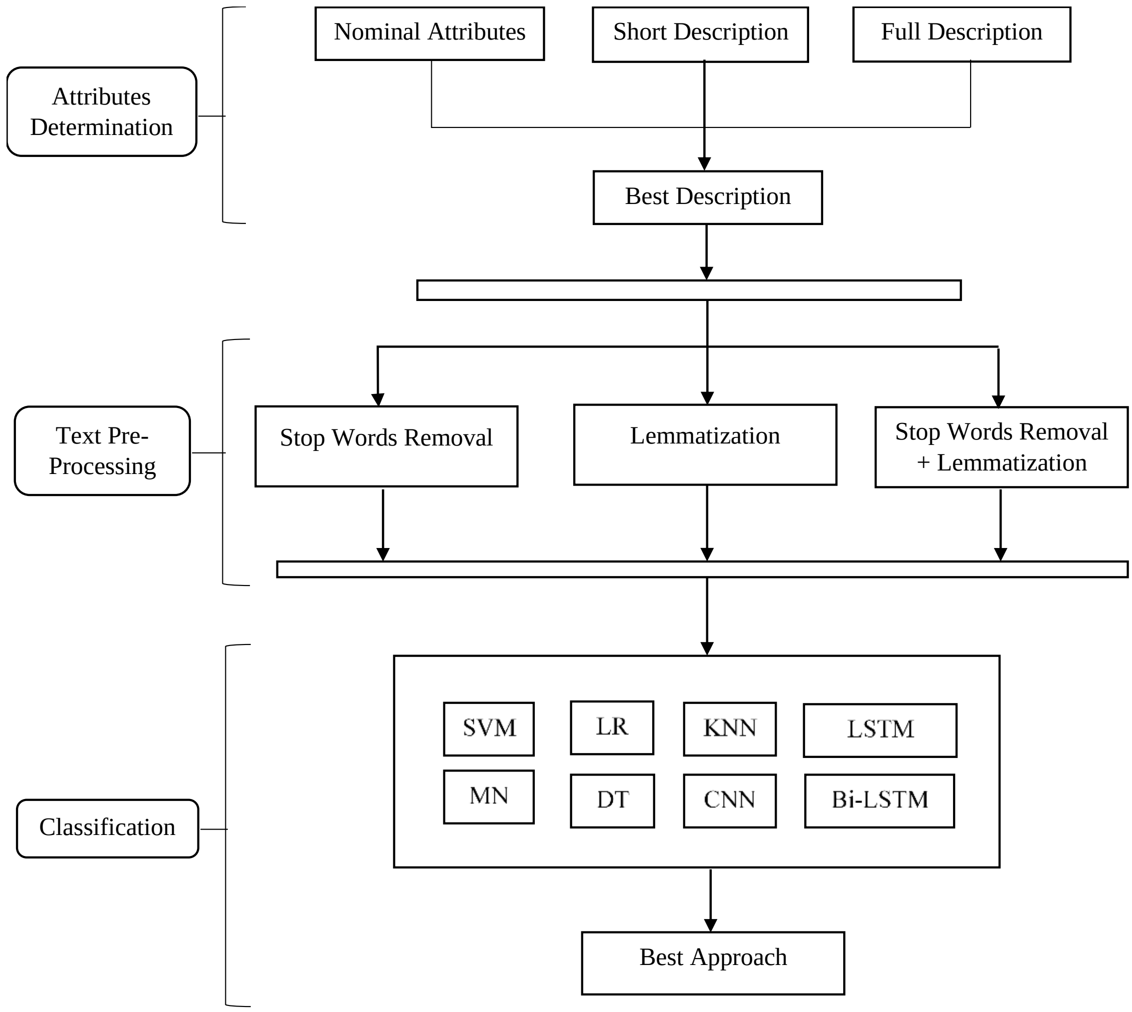

Figure 7 - Categorization Proposal Workflow

The research starts by evaluating which attributes are the most relevant for the prediction of the resolution category, by comparing the performance between considering only the nominal and text categories.

After assessing the importance of the textual data, the authors also evaluate the use of the short description versus full description to confirm which description provides better prediction results. Since one cannot be sure if the preprocessing steps actually improve or harm the performance of the ML models, some preprocessing techniques were investigated. This research explores the impact of the removal of stop-words and normalization, specifically lemmatization to the classification results.

As previously shown, the dataset is highly imbalanced, so for both approaches the dataset was under sampled to 6000 incidents for each resolution category in an attempt to reduce the risk of bias in the classification. For the same reason, since accuracy might not be enough to evaluate the models' performance as previously stated, this research also uses the F1-Score as evaluation metric. The training process is based on 5-Fold cross-validation dedicating 80% for training and 20% for testing. The goal of cross-validation is to test a model's ability to predict new data that was not used in the training process, in order to flag problems like overfitting or selection bias (Cawley & Talbot, 2010).

In the neural-based approach, for the networks first layer, an embedding layer is used that takes sequence of tokens as input to generate their representations. For this, the ticket text is divided in tokens in a tokenization process at word level transforming in each text entry into a token vector. Each vector is then transformed to a sequence of integers where each word is represented by a unique integer number and it is ensured that the sequences have the same size. This is achieved by adding padding until each sequence has the same length as a predefined selected length. The sequences are then used as input to the networks. The last layer of each NN architecture is the output layer, that maps the previous layer output to the corresponding classification label. The process is achieved by a densely connected layer with the number of neurons equivalent to the number of possible categories. This dense layer uses a softmax activation function to identify the class with the highest probability and categorical cross entropy as the loss function to obtain the classification result.

To implement the algorithms and NN models, this work used the Python programming language (Rossum & Drake, 1995) with the libraries *Scikit-learn* (Pedregosa et al., 2011) for the linear algorithms and *Keras* (Chollet, 2015) for the NN.

4.2. Development and Demonstration

This section presents all the results after applying the steps described in the Design section.

As previously stated, first the performance between using the nominal attributes versus the textual attributes will be assessed. The TM techniques applied to the textual data in this step were only the ones presented in the data preparation section.

The nominal attributes to use are the are the following: Department, Assignment Group, Incident Category, Incident Sub-category, Severity, and Location. Table 11 presents the achieved results with the application of the nominal attributes against the text attributes. It is possible to observe the importance of the textual attributes for the prediction. In all models, using the short or full description provides a better performance compared to the nominal attributes. It is also possible to observe that the full description presents better results than the short description. This might be related to the difference in size between them. The richer vocabulary and bigger word dimension of the full description provides better patterns and representation of the resolution domain compared to the short description. Overall the neural networks presented a better performance over the rest of the models, with the CNN achieving the best results with an accuracy of 56.45% and f1-score of 56.8%. Out of the neural networks the best model is SVM with an accuracy of 54.38% and F1-score of 53.79%.

Table 10 - Prediction results between nominal attribute vs short vs full description

	Nominal attributes		Short description		Full description	
	Accuracy (%)	**F1-Score (%)**	**Accuracy (%)**	**F1-Score (%)**	**Accuracy (%)**	**F1-Score (%)**
MNB	30.05	29.96	39.64	38.2	44.02	44.11
DT	30.59	29.80	39.87	39.7	41.72	40.66
LR	32.31	31.80	41.91	40.48	50.28	50.59
KNN	31.94	30.02	40.45	39.48	40.46	39.77
SVM	31.96	32.59	44.09	43.38	54.38	53.79
LSTM	34.18	33.59	47.41	47.95	56.03	56.28
Bi-LSTM	34.35	33.97	47.50	47.81	55.84	55.96
CNN	36.19	35.81	48.06	47.20	**56.45**	**56.80**

As next step this research analyses the impact of the removal of stop-words from the description’s corpus. Stop words removal consists on the elimination of common words , called stop-words, that are not meaningful and are commonly used to connect speech like propositions (Kotu, Deshpande, Kotu, & Deshpande, 2015). The results are presented in Table 11. Again, the neural networks models presented overall better results. The removal of stop-words offered no improvements in most of the models’ performance except for the CNN model that showed a 1% accuracy increase over the baseline performance.

Table 11 - Prediction results with stop-words removal

	Short description		Full description	
	Accuracy (%)	**F1-Score (%)**	**Accuracy (%)**	**F1-Score (%)**
MNB	39.30	38.76	44.19	43.43
DT	40.06	39.21	40.77	40.63
LR	41.65	40.02	49.01	48.46
KNN	40.03	39.51	41.89	41.21
SVM	44.51	43.01	53.40	53.03
LSTM	47.20	46.72	56.09	55.78
Bi-LSTM	47.24	46.21	56.20	55.65
CNN	47.16	47.12	**57.20**	**57.56**

This work also explores the use of normalization which attempts to identify the root of a term and reduces words to their base form (Son et al., 2014). Similar to the removal of stop-words, it is applied in an attempt to reduce the number of features and noise. Especially, lemmatization is used which according to Balakrishnan and Ethel (2014) "is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item". Table 12 shows the achieved prediction results with the application of lemmatization to the tickets descriptions field and at last, Table 13 presents the results of the combination of stop-words removal and lemmatization. Again, the impact was minimal across most of the models and presented similar values to the one of previous experiments. The CNN presented another performance increase, the best results were 58.63% accuracy and 58.51% F1-score with the combination of stop-words removal and lemmatization.

Table 12 - Prediction results with application of lemmatization

	Short description		Full description	
	Accuracy (%)	**F1-Score (%)**	**Accuracy (%)**	**F1-Score (%)**
MNB	39.94	39.04	43.45	42.88
DT	39.73	39.41	40.22	40.21
LR	42.37	40.90	49.38	48.90
KNN	41.77	40.82	39.66	39.31
SVM	45.36	44.39	54.42	54.10
LSTM	46.74	45.85	56.58	56.24
Bi-LSTM	47.14	46.80	55.22	55.40
CNN	47.60	47.86	**58.18**	**58.00**

Table 13 - Prediction results with stop-words removal and application of lemmatization

	Short description		Full description	
	Accuracy (%)	**F1-Score (%)**	**Accuracy (%)**	**F1-Score (%)**
MNB	39.66	38.22	43.42	42.46
DT	40.81	40.13	40.58	40.40
LR	41.92	40.50	48.36	47.68
KNN	40.82	39.90	42.46	41.66
SVM	44.97	43.66	53.02	52.59
LSTM	47.30%	46.34	56.37	56,02
Bi-LSTM	47.61%	46.12	56.62	56.17
CNN	48.06%	47.64	**58.63**	**58.51**

Overall the results were considerably similar across the models and experiments but taking into account the volume of the data of a million ticket even a 2% difference in the accuracy has a big impact. Considering 1 million tickets, it might be the difference in correctly or incorrectly predicting 20,000 tickets.

Chapter 5 – Incident Resolution Recommendation

This chapter presents the design, development and demonstration sections to create the artifact for the incident resolution actions recommendation.

5.1. Design

The proposed method for the incident resolution action recommendation follows the steps bellow and the process workflow is presented in Figure 8.

Step 1: With a new incoming ticket, predict the incident resolution category, filter the database and find tickets with the same incident category and incident resolution category as the new ticket. This is done based on the premise that tickets with a similar problem and resolution have the same incident category and resolution category. It also decreases the search dimension for the possible tickets to where the resolution should be extracted.

Step 2: Use a similarity metric to find the top *K* most similar tickets to the new ticket.

Step 3: After having the top *K* most similar tickets, the next step is to extract the resolution actions from those tickets' resolution field.

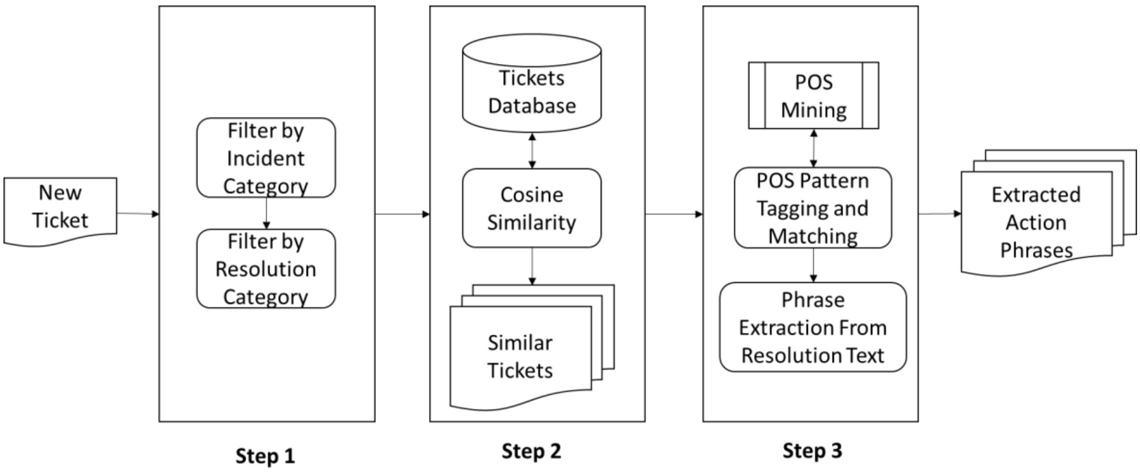

Figure 8 - Resolution Actions Extraction WorkFlow

5.2. Development

In this section, it is presented the development of the steps presented in the previous Design section. The first step is a straightforward one so the development presented in this section will only be related to the step two and three.

5.2.1. Step 2 - Incident Tickets Similarity

As previously stated, this step is used to find the top *K* most similar tickets in the dataset to a new incoming ticket by using a similarity metric. In the task of finding similar documents, a variety of similarity or distance measures have been proposed such as Euclidean distance, cosine similarity and Jaccard index (Huang, 2008). For this work, the cosine similarity was chosen being one of the most popular similarity metrics applied to text documents, such as in several information retrieval applications (Baeza-Yates & Ribeiro-Neto, 1999) and also clustering (Larsen & Aone, 1999).

The cosine similarity consists in the cosine of the angle between two vectors and captures a scale invariant understanding of similarity (Strehl, Strehl, Strehl, Ghosh, & Mooney, 2000). A strong property is that it does not depend on the length of the documents, what allows documents with the same composition but different totals to be treated identically. Because of this, it is considered an effective measure for documents similarity.

Given two documents $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ their cosine similarity is:

$$(6)\ SIM_C\left(\overrightarrow{t_a}, \overrightarrow{t_b}\right) = \frac{\overrightarrow{t_a} \times \overrightarrow{t_b}}{|\overrightarrow{t_a}| \times |\overrightarrow{t_b}|},$$

Where $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ are *m*-dimensional vectors over the term set $T = \{t_1, ..., t_m\}$. Each dimension contains a non-negative weight value that represents a term in the document. As result, the cosine similarity is non-negative and limited between [0,1].

In the previous chapter, the use of the full description presented the best results in predicting the resolution domain compared to the short description, so the full description TF-IDF vector representation is used to calculate the similarity between tickets. Tickets are only considered as similar if their cosine similarity value is higher than a certain threshold *h*, e.g. (0.5). This is done for the rare possible case where the number of tickets in the dataset of a certain incident and resolution category combination is less than or closer to *K*.

5.2.2. Step 3 - Resolution Actions Extraction

After extracting the top K most similar tickets to the new incoming ticket, the next step is to extract the resolution actions from those tickets' resolution field.

This work explores the use of dependency parsing and part-of-speech tagging for the action extraction. The key insight for extracting phrases from the resolution text that denote actions phrases in the tickets is that they usually contain a verb representing an operation or action. The extraction is based on a set of defined part-of-speech patterns. To create the patterns the authors analyzed the resolution corpus and manually identified some patterns, and also, derived from the work done by Agarwal et. al. (2017), the authors also applied a POS mining process to identify the most relevant POS patterns in the corpus.

The POS mining process follows the workflow shown in Figure 9. The resolution corpus is tokenized into POS tags using the open-source software library SpaCy (2017) with stop words removal applied. To define the most relevant POS patterns, n-grams of the POS tags are extracted from the tagged text with n ranging from 2 to 5 and a frequency count analysis is applied to the extracted n-grams considering only the ones containing at least one verb. Some examples of the obtained patterns are presented in Table 14.

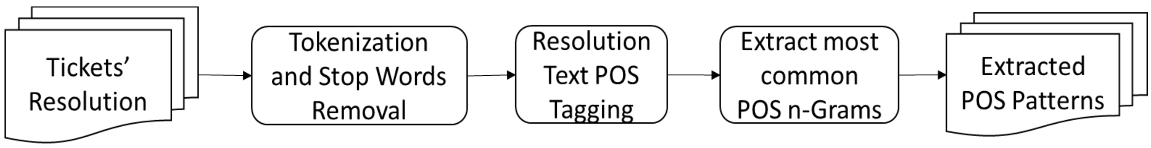

Figure 9 - POS pattern mining process workflow

More patterns variation can be added to the list by replacing the noun tag to an adjective or pronoun tag. The pattern matching is applied to the POS tagged resolution text from the similar tickets and the output is a list of short phrases representing the actions taken similar to the examples demonstrated in Table 14.

In an attempt to reduce the noise and decrease redundancy, stemming is applied to the extracted phrases. This helps handling with terms that are similar like "installed", "install", etc. Afterwards, the phrases are much cleaner but still have some redundancies.

Table 14 - Sample of actions POS patterns

	Pattern	Pattern Representation	Examples
1	Verb – Noun sequence	VB – NN – … – NN	Update client network driver; Deleted cookies; Reset internet explorer
2	Noun – Verb	NN – VB	User called; Software reinstalled
3	Noun | Adjective sequence – Verb – Noun | Adjective sequence	NN – … – NN – VB – NN – … – NN	Server needs password reset
4	Verb – Verb – Noun	VB – VB – NN	Install required software
5	Verb – Noun – Verb – Noun – Noun	VB – NN – VB – NN – NN	Install software using software center, Place windows account cancellation request
6	Adjective – Noun – Verb – Verb - Verb	ADJ – NN – VB – VB – VB	Secure login has been installed; OfficeScan engine has been started
7	Noun – Verb – Verb	NN – VB – VB	Software was installed
8	Verb – Adposition – Noun	VB – IN – NN	Connected via IP; Informed via email
9	Verb – Adposition – Noun – Noun	VB – IN – NN – NN	Login via Internet explorer

5.3. Demonstration

To evaluate if a ticket is considered solved or not is necessary to compare the extracted actions phrases for the new ticket with the ones from the similar tickets.

A lot of the extracted action phrases are from the shorter patterns. The shorter a pattern is the more matches it extracts from the resolution corpus and the longer a pattern is, the more specific it becomes, and the fewer matches it gets. There is also the case of duplicate extractions since a shorter pattern might be part of a longer pattern, e.g. in Table 14, pattern *1* is a sub pattern of pattern *6*, what means that for every phrase extracted by pattern *6* there is also a subphrase extracted by pattern *1*. This means that a lot of redundant phrases are extracted. Because of this redundancy it is not possible to consider the whole set of extracted phrases for the new ticket in the comparation process. Based on the previous statements, this research considers a ticket as solved if a certain predefined percentage threshold of its extracted resolution action phrases is present in the set of action phrases extracted from the similar tickets' resolutions. This threshold is labeled as solved acceptance threshold.

For the experiment configuration, the top-*K* similar tickets considered for the actions' extraction was from 5 to 30 with intervals of 5. A ticket will only be similar if its cosine similarity score is at minimum 0.5. The minimum acceptance threshold for a ticket to be considered solved was set to a range of 30% to 60% with intervals of 10. The experiment accuracy is calculated by the following ratio:

$$(7)\ Accuracy = \frac{number\ of\ solved\ tickets}{total\ number\ of\ tickets}$$

For the test data, 1000 tickets from each incident category were selected to a total of 10000 tickets to build a test set.

Table 15 presents the accuracy results. Depending on the top-*K* and the solved acceptance percentage threshold, the experiment achieved an accuracy ranging from 31.5% to 40.5% meaning that 31% to 40% of the total number of tickets tested were considered as solved.

Table 15 - Experiment accuracy results for solving tickets with the actions extraction module

		Acceptance threshold (%)			
		30	**40**	**50**	**60**
Top-*K*	**5**	31.50	29.00	27.25	24.75
	10	36.00	32.75	30.75	27.75
	15	38.25	35.25	33.50	29.50
	20	39.50	36.75	35.00	30.50
	25	40.25	37.50	35.50	31.25
	30	40.50	37.75	35.75	31.50

As expected, accuracy goes up with a bigger top k and with a lower solved acceptance threshold. The average processing time for each ticket was 3 to 10 seconds depending on the categories, since the ones with a bigger sample size mean a bigger search space for similar tickets.

Increasing the top-*K* also increases computation time and the total number of extracted actions. A subsequent experiment was performed to calculate the average number of actions extracted for a ticket by applying the extraction module to 100,000 tickets. From those, it was possible to extract at least one action from 85,196 representing a coverage of 85%, which considering the noisy nature of the resolution field is a quite good result, and the average number of actions extracted was 6.95 with a median of 5. So, choosing a high value to K can significantly increase the number of possible actions and computation time. A real-world application of the module would be necessary to find the optimal value for the configuration variables based on application, domain and performance results by testing variation configuration and try to understand the impact in the accuracy, which unfortunately was not possible in this research.

Chapter 6 – Evaluation

In this chapter, it is presented the possible impact of the proposed method to the incident resolution process. The impact is analyzed by considering the average time allocated to the resolution step in the IM process.

As previously stated, after a new incident ticket has been created it will be assigned to a service agent to apply his knowledge expertise and attempt to solve the problem. Some high-level requests or incidents may require a bigger effort or if the incident is related to a configuration item, it would be necessary to apply a deeper investigation to the possible problem. So, depending on the possible problem that originated the incident, the workload for the service agent may vary and obviously it also depends on the agent experience. A new agent would take a longer time to understand the problem and find the correct resolution for the incident.

From the dataset used in this research it is possible to calculate how long does a ticket takes to be solved by the service agent by calculating the difference between the timestamp of when the ticket was assigned to the agent and the timestamp the agent set the ticket as resolved. For the current dataset, the average solve time for an incident ticket is 98.27 hours which is equivalent to 4.09 days. These values were confirmed by the company. This does not mean that an agent dedicated 4 days exclusively on each ticket since normally the agent is solving multiple incident at the same time when in work hours. Table 16 represents an analysis of the achieved accuracy range and their equivalent number of tickets and representation in days and years and also the equivalent time the proposed method would have taken to solve those same number of tickets. For the last one, it is considered the worst average time of 10 seconds per ticket.

Table 16 - Resolution time comparation between manual and proposed method by accuracy

Accuracy (%)	# Solved Tickets	Equivalent in Days	Equivalent in Years	Equivalent Time in Days By Proposed Method
31	558,000	2,282,220	6,248	64.58
35	630,000	2,576,700	7,054	72.92
41	738,000	3,018,420	8,264	85.41

In the last year, the organization had 1.8 million incident tickets opened. Considering the proposed system lowest accuracy achieved of 31% as was shown in the previous chapter, in the best-case scenario it means that 558,000 tickets would have been considered solved or the agent would have the resolution actions readily available. Those 558,000 incident tickets represent a summation of 2,282,220 days or 6,248 years allocated to them which, as it is possible to see, implies a significant amount of time and resources possibly saved. In the best-case scenario with the achieved of 41%, represents 738,000 tickets from the dataset and 3,018,420 days allocated to those tickets. In contrast, the proposed method in the worst-case scenario, would take 85.41 days to find the resolution for the same volume of tickets.

The amount of manpower possibly saved could be reallocated to other areas of the organization wherever is necessary and useful. Even though the presented values are based on a best case-scenario, this could represent a significant impact to the organization.

Related to the proposed method, the authors believe that it is important to refer that the method does not achieve full automation of the incident resolution process. In the current state, the method could be used as a decision support system aiding a service agent to find a suitable resolution to the incident by providing a possible set of actions to be taken.

Chapter 7 – Conclusion

Every organization allocates a considerable amount of resources towards making sure that their assets are running incident free and IM is an essential process to achieve that goal. Plenty of incidents are not new, making a service agent spend time and resource towards an incident ticket which has already been seen and its respective resolution is documented in the ITS knowledge base.

This work aimed to automate the IM resolution process and introduced a method to attempt to automatically extract and propose suitable resolution actions for new incoming incident tickets. The proposed method was developed and tested on incident data provided by the IT department of a big multinational company serving hundreds of thousands of employees.

Two major contributions were developed during this research and the creation of the proposed method:

- A ML model to predict an incident resolution category was developed by applying supervised ML algorithms. Different NLP techniques and combinations were applied to assess their impact on the models' performance. The best result was achieved by CNN with embeddings and the application of stop-words removal and lemmatization, achieving an accuracy of 58.63% and a F1-score of 58.1%. It was also discovered that, contrary to the incident category prediction, the use of the ticket full description as input to the models, provided a better prediction performance compared to the short description. This concludes that the full description contains a better representation of the resolution domain while the short description contains a better representation of the incident problem domain.
- An action phrase extraction module based on POS pattern rules matching to extract action phrases from the new ticket most similar tickets' resolution field. Those similar tickets were obtained by using the cosine similarity score with the tickets' full description as input since as previously shown, is the best one for the resolution domain. A mining process was applied to create the POS patterns by tagging the dataset resolution field and extracting the most frequent POS n-grams sequences containing a verb. This process allowed the creation of a list of patterns that are tailored to the used dataset and its domains but can also be used in other datasets. The action phrases are extracted by retrieving every phrase that match a POS pattern in the resolution field.

In the experimental results, 31% to 41% of the 10,000 tested tickets were considered as solved by the proposed method, which considering the high volume of tickets and the average resolution time spent by the a service agent on each ticket, represents a significant amount of resources that can be saved or relocated. The results indicate that there is still room to improve in the automation of IM. The quality of data is essential to achieve a good performance. It is necessary that every member of the IM process in the organization is compliant to the best practices and the process rules to ensure the quality of the data and the service.

7.1. Research Limitation

An identified limitation is that the model for the incident resolution prediction could only achieve an accuracy of 58% which is not ideal for a real-world application. This can be partially be attributed to the quality of data. According to the organization, since the resolution category is traditionally only used for reporting, the service agents are a bit careless when selecting the field which affects the quality of the data and consequently a model prediction quality.

Also, not every ITS has the resolution category field which makes the proposed prediction module not applicable, but since most, if not all possess the incident category, the module for the action extraction can still be applied. The consequence would be that the search space for the similar tickets would be bigger. It is even possible to apply the extraction module without an incident category, but for each ticket it would be necessary to search the whole dataset for the similar tickets which would be computationally heavy and would take a considerably higher time to run.

The actions' extraction module, although presented a good performance result, it is also not yet ideal for the real-world application where it is possible to unworriedly trust in the suggested resolution actions. It would still be necessary for a service agent to validate the extracted resolution. In the same note, it is necessary to run a real-world application test to establish the optimal values for the number of similar tickets to be considered and the minimum percentage threshold for a ticket to be considered as solved.

7.2. Future Work

Other than improve the prediction results for the resolution category. The authors also pretend to improve the action's extraction module.

Instead of only extraction the actions as separated entities, it would be interesting to extract sequences of connected actions where the order of the actions is considered and assured. This could be achieved by calculating the probability of a group of actions appearing on the same tickets and one after the other. Basically, a next word prediction process which is commonly used in Language Modeling but using the action phrases instead of words. This could possibly reduce the number of proposed actions and also increase the quality of those actions by provide a more semantically connected sequence of actions.

In the same note, it would also be interesting to apply a domain-based Name Entity Recognition (NER) module. The authors experimented with some commonly used libraries, but the results were very poor since those libraries were created with mainly open domain, well written texts and could not recognize most IT terms like software names. It would be necessary to manually identify and tag the corpus to create and train a NER model which is specific to the dataset domain. The application of NER to the process could also help connect actions that are mentioning the same entities like a software, hardware, etc.

Bibliography

Agarwal, S., Aggarwal, V., Akula, A. R., Dasgupta, G. B., & Sridhara, G. (2017). Automatic problem extraction and analysis from unstructured text in IT tickets. IBM Journal of Research and Development, 61(1), 4:41-4:52. https://doi.org/10.1147/JRD.2016.2629318

Aggarwal, C. C., & Zhai, C. (2012). Mining text data. Springer.

Aggarwal, V., Agarwal, S., Dasgupta, G. B., Sridhara, G., & Vijay, E. (2016). ReAct: A system for recommending actions for rapid resolution of IT service incidents. Proceedings - 2016 IEEE International Conference on Services Computing, SCC 2016, 1–8. https://doi.org/10.1109/SCC.2016.134

Agnihotri, D., Verma, K., & Tripathi, P. (2014). Pattern and cluster mining on text data. Proceedings - 2014 4th International Conference on Communication Systems and Network Technologies, CSNT 2014, 428–432. https://doi.org/10.1109/CSNT.2014.92

Al-Hawari, F., & Barham, H. (2019). A machine learning based help desk system for IT service management. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/J.JKSUCI.2019.04.001

Altintas, M., & Tantug, A. C. (2014). Machine Learning based Ticket Classification in Issue Tracking Systems. Proceedings of the International Conference on Artificial Intelligence and Computer Science (AICS 2014). https://doi.org/10.1109/DATE.2009.5090792

Baeza-Yates, R. A., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. Retrieved from https://dl.acm.org/citation.cfm?id=553876

Balakrishnan, V., & Ethel, L.-Y. (2014). Stemming and Lemmatization: A Comparison of Retrieval Performances. Lecture Notes on Software Engineering. https://doi.org/10.7763/lnse.2014.v2.134

Bari, A., Chaouchi, M., & Jung, T. (2014). Predictive analytics for dummies. In --For dummies.

Bartolini, C., Stefanelli, C., & Tortonesi, M. (2008). SYMIAN: A Simulation Tool for the Optimization of the IT Incident Management Process. In Proceedings of the 19th IFIP/IEEE international workshop on Distributed Systems: Operations and Management: Managing Large-Scale Service Deployment (pp. 83–94). https://doi.org/10.1007/978-3-540-87353-2_7

Cawley, G. C., & Talbot, N. L. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. Journal of Machine Learning Research, 11(Jul), 2079–2107. Retrieved from http://www.jmlr.org/papers/v11/cawley10a.html

Chen, Y., Tao, S., Yan, X., Anerousis, N., & Shao, Q. (2010). Assessing Expertise Awareness in Resolution Networks. 2010 International Conference on Advances in Social Networks Analysis and Mining, 128–135. https://doi.org/10.1109/ASONAM.2010.82

Chollet, F. (2015). Keras: The Python Deep Learning library. Keras.Io.

Daskalaki, S., Kopanas, I., & Avouris, N. (2006). EVALUATION OF CLASSIFIERS FOR AN UNEVEN CLASS DISTRIBUTION PROBLEM. Applied Artificial Intelligence, 20(5), 381–417. https://doi.org/10.1080/08839510500313653

Galup, S. D., Dattero, R., Quan, J. J., & Conger, S. (2009). An overview of IT service management. Communications of the ACM, 52(5), 124. https://doi.org/10.1145/1506409.1506439

Ghiţ, B., Iosup, A., & Epema, D. (2013). Towards an optimized big data processing system. Proceedings - 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2013. https://doi.org/10.1109/CCGrid.2013.53

Gorunescu, F. (2011). Introduction to Data Mining. https://doi.org/10.1007/978-3-642-19721-5_1

Gupta, R., Prasad, K. H., & Mohania, M. (2008a). Automating ITSM incident management process. 5th International Conference on Autonomic Computing, ICAC 2008, 1, 141–150. https://doi.org/10.1109/ICAC.2008.22

Gupta, R., Prasad, K. H., & Mohania, M. (2008b). Information integration techniques to automate incident management. NOMS 2008 - 2008 IEEE Network Operations and Management Symposium, 979–982. https://doi.org/10.1109/NOMS.2008.4575262

Han, J., Kamber, M., & Pei, J. (2012). Data Mining : Concepts and Techniques : Concepts and Techniques (3rd Edition). In Data Mining. https://doi.org/10.1016/B978-0-12-381479-1.00001-0

Hossin, M., & Sulaiman, M. N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process. https://doi.org/10.5121/ijdkp.2015.5201

Huang, A. (2008). Similarity Measures for Text Document Clustering. Proceedings of the New Zealand Computer Science Research Student Conference.

Huttermann, M. (2012). DevOps for Developers. In Apress. https://doi.org/10.1017/CBO9781107415324.004

Iden, J., & Eikebrokk, T. R. (2013). Implementing IT Service Management: A systematic literature review. International Journal of Information Management, 33(3), 512–523. https://doi.org/10.1016/j.ijinfomgt.2013.01.004

Jan, E., & Li, Y. (2016). Service analytics for IT service management. IBM Journal of Research and Development, 60(2), 1–17. https://doi.org/10.1147/JRD.2016.2520620

Kang, Y., & Zaslavsky, A. (2010). A knowledge-rich similarity measure for improving IT incident resolution process. Proceedings of the 2010 …, 1781–1788. https://doi.org/10.1145/1774088.1774466

Kao, A., & Poteet, S. R. (2007). Natural language processing and text mining. In Natural Language Processing and Text Mining. https://doi.org/10.1007/978-1-84628-754-1

Kosasi, S., Prabowo, H., & Budiastuti, D. (2017). Improving information performance of schools in higher education through IT service management. 2017 5th International Conference on Cyber and IT Service Management, CITSM 2017. https://doi.org/10.1109/CITSM.2017.8089246

Kotu, V., Deshpande, B., Kotu, V., & Deshpande, B. (2015). Text Mining. Predictive Analytics and Data Mining, 275–303. https://doi.org/10.1016/B978-0-12-801460-8.00009-4

Krishnan, G., & Ravindran, V. (2017). IT service management automation and its impact to IT industry. 2017 International Conference on Computational Intelligence in Data Science(ICCIDS), 1–4. https://doi.org/10.1109/ICCIDS.2017.8272633

Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’99, 16–22. https://doi.org/10.1145/312129.312186

Laurikkala, J. (2001). Improving Identification of Difficult Small Classes by Balancing Class Distribution. https://doi.org/10.1007/3-540-48229-6_9

Lepmets, M., Cater-Steel, A., Gacenga, F., & Ras, E. (2012). Extending the IT service quality measurement framework through a systematic literature review. Journal of Service Science Research. https://doi.org/10.1007/s12927-012-0001-6

Li, H., & Zhan, Z. (2012). Machine learning methodology for enhancing automated process in IT incident management. Proceedings - IEEE 11th International Symposium on Network Computing and Applications, NCA 2012, 191–194. https://doi.org/10.1109/NCA.2012.28

Marcu, P., Grabarnik, G., Luan, L., Rosu, D., Shwartz, L., & Ward, C. (2009). Towards an optimized model of incident ticket correlation. 2009 IFIP/IEEE International Symposium on Integrated Network Management, IM 2009, 569–576. https://doi.org/10.1109/INM.2009.5188863

Mirończuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2018.03.058

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems. https://doi.org/10.1016/j.dss.2010.08.006

OGC. (2007). The Official Introduction to the ITIL Service Lifecycle. In Office of Government Commerce. https://doi.org/citeulike-article-id:2420971

Orta, E., & Ruiz, M. (2019). Met4ITIL: A process management and simulation-based method for implementing ITIL. Computer Standards & Interfaces, 61, 1–19. https://doi.org/10.1016/j.csi.2018.01.006

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis: Foundations and Trends in Information Retrieval. Foundations and Trends® in Information Retrieval. https://doi.org/10.1561/1500000011

Pattern Recognition and Machine Learning. (2007). Journal of Electronic Imaging, 16(4), 049901. https://doi.org/10.1117/1.2819119

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The Design Science Research Process: A Model for Producing and Presenting Information Systems Research. The Proceedings of Design Research in Information Systems and Technology DESRIST 2006, 83–106.

https://doi.org/10.2753/MIS0742-1222240302
Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302
Rossum, G. Van, & Drake, F. L. (1995). Python Tutorial, Technical Report CS-R9526. Centrum Voor Wiskunde En Informatica (CWI). https://doi.org/10.1016/j.abb.2004.09.015
Roy, S., Yan, J. J. Y. T., Budhiraja, N., & Lim, A. (2016). Recovering resolutions for application maintenance incidents. Proceedings - 2016 IEEE International Conference on Services Computing, SCC 2016, 617–624. https://doi.org/10.1109/SCC.2016.86
Salah, S., Maciá-Fernández, G., & Díaz-Verdejo, J. E. (2019). Fusing information from tickets and alerts to improve the incident resolution process. Information Fusion, 45, 38–52. https://doi.org/10.1016/j.inffus.2018.01.011
Salah, S., Maciá-Fernández, G., Díaz-Verdejo, J. E., & Sánchez-Casado, L. (2016). A Model for Incident Tickets Correlation in Network Management. Journal of Network and Systems Management, 24(1), 57–91. https://doi.org/10.1007/s10922-014-9340-6
Shao, Q., Chen, Y., Tao, S., Yan, X., & Anerousis, N. (2008). Efficient ticket routing by resolution sequence mining. Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 08. https://doi.org/10.1145/1401890.1401964
Sharma, A., Sharma, M. K., & Dwivedi, R. K. (2017). Literature Review and Challenges of Data Mining Techniques for Social Network Analysis (Vol. 10). Retrieved from http://www.ripublication.com
Silva, S., Pereira, R., & Ribeiro, R. (2018). Machine learning in incident categorization automation. Iberian Conference on Information Systems and Technologies, CISTI, 2018-June, 1–6. https://doi.org/10.23919/CISTI.2018.8399244
Son, G., Hazlewood, V., & Peterson, G. D. (2014). On Automating XSEDE User Ticket Classification. 1–7. https://doi.org/10.1145/2616498.2616549
SpaCy. (2017). spaCy.
Strehl, A., Strehl, A., Strehl, E., Ghosh, J., & Mooney, R. (2000). Impact of Similarity Measures on Web-page Clustering. IN WORKSHOP ON ARTIFICIAL

INTELLIGENCE FOR WEB SEARCH (AAAI 2000, 58--64. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.2110

Tang, L., Li, T., Shwartz, L., & Grabarnik, G. (2013). Recommending resolutions for problems identified by monitoring. 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), 134–142.

Vaishnavi, V., Kuechler, B., & Petter, S. (2004). DESIGN SCIENCE RESEARCH IN INFORMATION SYSTEMS. Retrieved from http://desrist.org/desrist/content/design-science-research-in-information-systems.pdf

Wan, J., Zhang, H., & Wan, D. (2011). Evaluation on Information Technology Service Management Process with AHP. Technology and Investment, 02(01), 38–46. https://doi.org/10.4236/ti.2011.21005

Wan, S. H. C., & Chan, Y. H. (2007). IT Service Management for campus environment -practical concerns in implementation. 10th IFIP/IEEE International Symposium on Integrated Network Management 2007, IM ’07. https://doi.org/10.1109/INM.2007.374833

Wang, Q., Zhou, W., Zeng, C., Li, T., Shwartz, L., & Grabarnik, G. Y. (2017). Constructing the Knowledge Base for Cognitive IT Service Management. 2017 IEEE International Conference on Services Computing (SCC), 410–417. https://doi.org/10.1109/SCC.2017.59

Weiss, G., & Provost, F. (2001). The effect of class distribution on classifier learning: an empirical study. In Technical Report ML-TR-44, Department of Computer Science, Rutgers University.

Yun, M., Lan, Y., & Han, T. (2017). Automate incident management by decision-making model. 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(, 217–222. https://doi.org/10.1109/ICBDA.2017.8078811

Zhao, G., & Yang, S. (2013). IT service incident management model decision based on ELECTRE III. Proceedings of 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2013. https://doi.org/10.1109/ICIII.2013.6702987

Zhong, N., Li, Y., & Wu, S.-T. (2012). Effective {Pattern} {Discovery} for {Text} {Mining}. IEEE Transactions on Knowledge and Data Engineering. https://doi.org/10.1109/TKDE.2010.211

Zhou, W., Tang, L., Zeng, C., Li, T., Shwartz, L., & Grabarnik, G. Y. (2016).

Resolution Recommendation for Event Tickets in Service Management. IEEE Transactions on Network and Service Management, 13(4), 954–967. https://doi.org/10.1109/TNSM.2016.2587807