Instituto Universitário de Lisboa

University Institute of Lisbon

Department of Information Science and Technology

# Automatization of Incident Categorization

Sara Alexandra Teixeira da Silva

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

**Master in Computer Engineering**

**Supervisor**

Rúben Filipe de Sousa Pereira, Assistant Professor, PhD

ISCTE-IUL

**Co-Supervisor**

Ricardo Daniel Santos Faro Marques Ribeiro, Assistant Professor, PhD

ISCTE-IUL

September 2018

# *Resumo*

De forma a acompanhar o crescimento da quantidade de incidentes criados no dia-a-dia de uma organização, houve a necessidade de aumentar a quantidade de recursos, de maneira a assegurar a gestão de todos os incidentes. A gestão de incidentes é composta por várias atividades, sendo uma delas, a categorização de incidentes. Através da junção de técnicas de Linguagem Natural e Processamento de Texto e de Algoritmos de Aprendizagem Automática propomos melhorar esta atividade, especificamente o Processo de Gestão de Incidentes. Para tal, propomos a substituição do subprocesso manual de Categorização inerente ao Processo de Gestão de Incidentes por um subprocesso automatizado, sem qualquer interação humana.

A dissertação tem como objetivo propor uma solução para categorizar corretamente e automaticamente incidentes. Para tal, temos dados reais de uma organização, que devido a questões de privacidade não será mencionada ao longo da dissertação. Os datasets são compostos por incidentes corretamente categorizados o que nos leva a aplicar algoritmos de aprendizagem supervisionada. Pretendemos ter como resultado final um método desenvolvido através da junção das diferentes técnicas de Linguagem Natural e dos algoritmos com melhor performance para classificar os dados. No final será avaliado o método proposto comparativamente à categorização que é realizada atualmente, de modo a concluir se a nossa proposta realmente melhora o Processo de Gestão de Incidentes e quais são as vantagens trazidas pela automatização.

**Palavras-chave**: Aprendizagem Automática, Classificação Automática de Incidentes, Classificação de Incidentes, Linguagem Natural, Processo de Gestão de Incidentes, Processamento de Texto.

# *Abstract*

To be able to keep up with the grow of the created incidents quantity in an organization nowadays, there was the need to increase the resources to ensure the management of all incidents. Incident Management is composed by several activities, being one of them, Incident Categorization. Merging Natural Language and Text Mining techniques and Machine Learning algorithms, we propose improve this activity, specifically the Incident Management Process. For that, we propose replace the manual sub-process of Categorization inherent to the Incident Management Process by an automatic sub-process, without any human interaction.

The goal of this dissertation is to propose a solution to categorize correctly and automatically the incidents. For that, there are real data provided by a company, which due to privacy questions will not be mention along dissertation. The datasets are composed by incidents correctly categorized, which leverage us to apply supervised learning algorithms. It is supposed to obtain as output a developed method through the merge of Natural Language Processing techniques and classification algorithms with better performance on the data. At the end, the proposed method is assessed comparatively with the current categorization done to conclude if our proposal really improves the Incident Management Process and which are the advantages brought by the automation.

**Key-words:** Automated Incident Categorization, Incident Categorization, Incident Management Process, Machine Learning, Natural Language, Text Mining.

# *Acknowledgements*

I would like to express my sincere gratitude to my Professors Rúben Pereira and Ricardo Ribeiro for all the support that they gave me and for everything I learned along this journey. Finally, I am grateful to my parents, sister and friends, without them my whole academic path wouldn't have been possible.

# Contents

# *List of Tables*

# *List of Figures*

# *Acronyms*

| | |
|---|---|
| **DSRM** | **D**esign **S**cience **R**esearch **M**ethodology |
| **DT** | **D**ecision **T**rees |
| **IC** | **I**ncident **C**ategorization |
| **ID3** | **I**terative **D**ichotomiser **3** |
| **IDF** | **I**nverse **D**ocument **F**requencies |
| **IM** | **I**ncident **M**anagement |
| **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 |
| **KNN** | K-**N**earest **N**eighbors |
| **ML** | **M**achine **L**earning |
| **NB** | **N**aïve **B**ayes |
| **NER** | **N**amed **E**ntity **R**ecognition |
| **NL** | **N**atural **L**anguage |
| **NLP** | **N**atural **L**anguage **P**rocessing |
| **RQ** | **R**esearch **Q**uestion |
| **SVM** | **S**upport **V**ector **M**achines |
| **TF** | **T**erm **F**requencies |
| **TFxIDF** | **T**erm **F**requency–**I**nverse **D**ocument **F**requency |
| **TM** | **T**ext **M**ining |
| **TC** | **T**ext **C**ategorization |
| **TP** | **T**rue **P**ositives |
| **FP** | **F**alse **P**ositive |
| **FN** | **F**alse **N**egative |

# 1 *Introduction*

Information Technology Service Management (ITSM) is a discipline for managing Information Technology (IT) operations (Galup, Dattero, Quan, & Conger, 2009) which provides a framework that deals with alignment and management of IT services, in conformity with the needs of the business, and which aim is to improve business performance through the best IT service delivery. Thus, ITSM focuses on the development of methodologies and tools to provide an efficient and high quality service (Marcu et al., 2009), which includes optimizing IT services and business operations and increasing employees' productivity and costumers' satisfaction.

The Incident Management (IM) process, one of the most important components of ITSM (Salah, Maciá-Fernández, Díaz-Verdejo, & Sánchez-Casado, 2016), focuses on tracking and managing all incidents, from opening until closure. Its goal is to resolve incidents as quickly as possible, ensuring the less impact for costumers and the correct operation of IT's services organizations (Gupta, Prasad, & Mohania, 2008b). IM is also the process of ITSM that provides visible gains to service quality most directly as well as cost reduction (Gupta, Prasad, & Mohania, 2008a).

In companies, incidents are created every day, which brings the need to record many events that negatively impacts the system operation. The growth of incidents brings difficulties to the responsible users included in the IM process, reducing the support performance.

## *1.1 Motivation and Research Context*

With the exponential usage of IT in companies, a lack in costumer support service has been verified (Dias Freire De Mello & Lopes, 2015). In companies, a lot of incident tickets are created every day, and specific IT teams exist to resolve them. However, in many cases this process is not entirely systematic and may be incoherent and inefficient (Salah et al., 2016), resulting in a waste of several resources which increases companies' costs (Song, Sailer, & Shaikh, 2009). Therefore, to be competitive, companies need an efficient and cost-effective service and support delivery (Zhou, Xue, Wang, & Shwartz, 2017). Consequently, many companies started to adopt tools to help and support teams that are responsible for the IM process (Marcu et al., 2009). Such tools are software systems used in organizations to register and track all incidents and typically refer to an Incident Ticket System (ITS).

A coordinated ITS provides a positive effect on the efficiency of the IM process, which in turn improves and increases companies' revenue. Most ITS follow the Information Technology Infrastructure Library (ITIL) (Cannon & Wheeldon, 2007), the most adopted ITSM framework to facilitate and help the decision-making process (Salah et al., 2016). ITIL delineates best practices and standards to IM, helping companies to improve their processes. ITS represents a significant contribution to an efficient IM process, to obtain lower costs and an increased organization growth.

To answer organization's needs, this work proposes an automatic categorization on ITS. Manual classification originates error prone, and consequently time consuming, which in large organizations is not feasible (Altintas & Tantug, 2014). With an automatic categorization, we intend to automate incident classification and in parallel reduce classification errors and useless time spent in the IM process.

Machine Learning (ML) turns possible the IM process automation, categorizing automatically incidents and assigning them to a resolution group. Text mining (TM) and Natural Language Processing (NLP) contributes to a correct categorization, highlighting the key words most relevant to a certain category.

## 1.2 Research Methodology

Design Science is a subject that creates and evaluates IT artefacts in order to solve organizational problems (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). To develop this dissertation, we adopt the Design Science Research Methodology (DSRM), which is composed by six activities that are presented in Figure 1.1 and are described below.

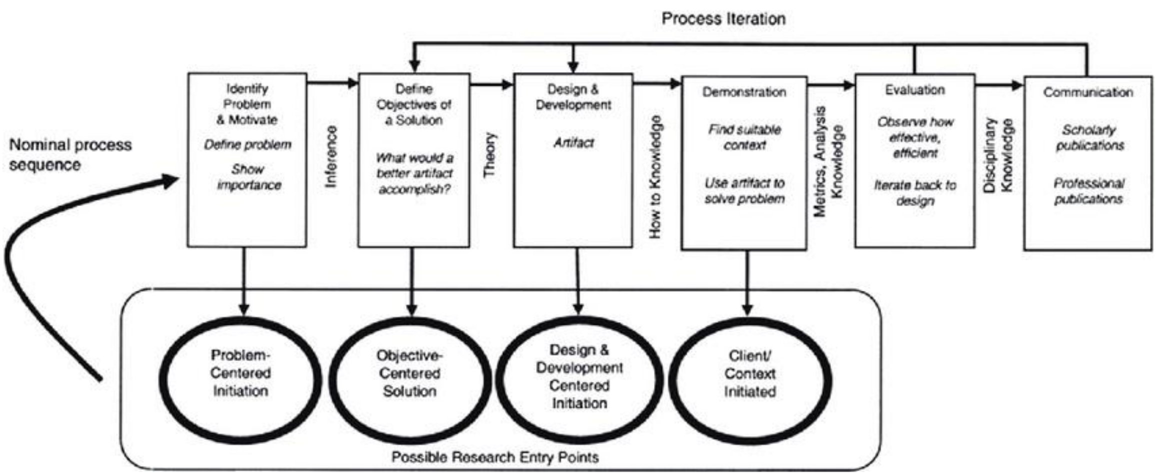


*Figure 1.1: Design Science Research Methodology Process Model* (Peffers et al., 2007)

Our research represents an industry need, since the goal is automating one of the processes of IM. Currently, the process of categorizing incidents is done manually by ITIL agents, so our proposal is to automate this process. Consequently, the approach that we adopted is an objective-centered solution, in which we are addressing the development and design of one artefact. The artefact focuses on the best method to categorize incidents automatically: first, we need to choose which are the algorithms to the best categorization possible; then, with the application of the different algorithms and TM techniques is obtained the best method. After that, with the best output method of the previous process, are categorized one more time the same incidents, but with different categories. Finally, we present the evaluation, to understand if our proposal helps to solve the problem and fits the defined objectives. Figure 1.2 presents the steps to follow along this dissertation.

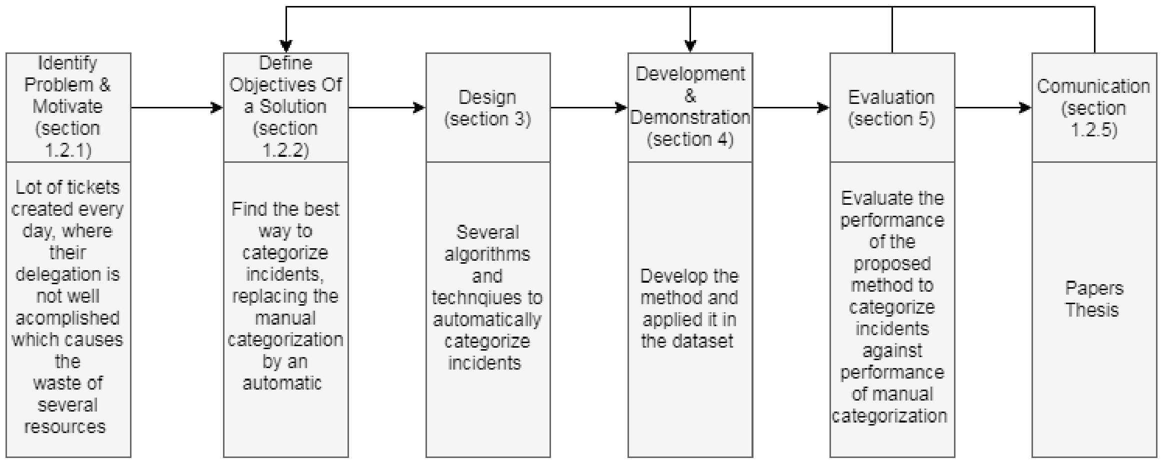


*Figure 1.2: Guideline for this research (Adapted from 1.1)*

In the next subsections, we will describe in detail each step presents in Figure 1.2.

### *1.2.1 Identify Problem and Motivate*

Delegation of incidents is not always well accomplished, and incidents are addressed to resolution groups that are not capable of solving them, causing delays in the whole process of dispatch (Agarwal, Sindhgatta, & Sengupta, 2012;Salah et al., 2016). Incidents are forwarded to finally be addressed to the right resolution group, which affects incident route negatively demanding the use of more resources, consequently, leading to wasted time generating customer dissatisfaction (Shao, Chen, Tao, Yan, & Anerousis, 2008).

In order to attain a right assignment, it is crucial to have an appropriate incident classification, process that attributes a suitable category to an incident, so they are routed more accurately (Cannon & Wheeldon, 2007).

Automating incident classification means no human error; reduced waste of resources, and no incorrect routing due to the wrong classification (Gupta, Prasad, Luan, Rosu, & Ward, 2009).

### 1.2.2 Define Objectives for a Solution

The goal of this research is to develop and propose a method that categorizes correctly and automatically an incoming incident, using ML algorithms and NLP techniques. To achieve the proposed objective there are three main research questions (RQ) that we will try to answer:

RQ1: Can we use supervised learning algorithms and NLP techniques to categorize automatically and correctly incident tickets?

RQ2: Can we find a method that correctly categorizes most of incident tickets?

RQ3: The proposed method improves the Incident Categorization (IC) process?

### 1.2.3 Design

In this thesis it is used a dataset from a company that due to privacy questions cannot be mentioned along the work. The provided dataset is composed approximately by 900,000 incident tickets and the dataset contains incidents from November of 2015 to December of 2017. The dataset under study contains incidents correctly categorized that have three levels of categorization. In this work, we will only consider two levels of categories. The incidents are recorded by two different ways, email or phone. Figure 1.3 shows the incidents creation from 2015 to 2017.

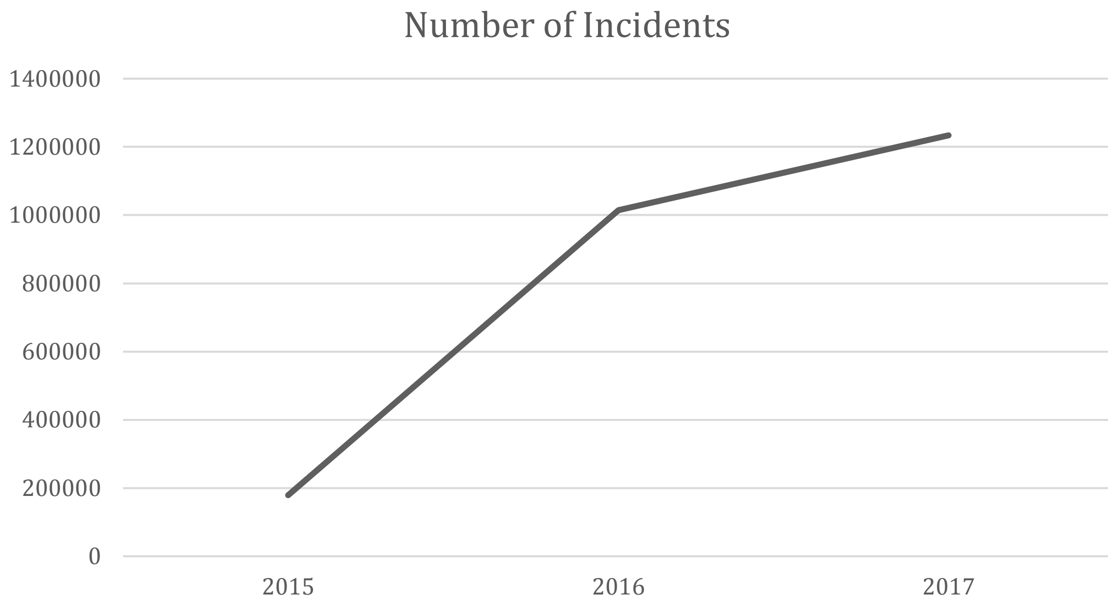


*Figure 1.3: Incidents Distribution from November of 2015 to December of 2017*

The set of first level categories is composed by the ten following categories: application, collaboration, enterprise resource planning (ERP), hosting services, network, security and access, output management, software, workplace, and support. After assigning a first level category to an incident, a second level category is assigned. The set of second level categories is composed by 81 categories: 32 belong to application, 7 to collaboration, 5 to ERP, 8 to hosting services, 5 to network, 3 to security and access, 3 to output management, 2 to software, 9 to workplace, and, finally, 7 to support. Table 1.1 presents the number of incidents per category and subcategory. The subcategories are present by ascending order of incidents.

*Table 1.1: Number of incidents per category and subcategory*

| Category (1st level) | Subcategory (2nd level) | Number of incidents per subcategory | Number of incidents per category | % of incident tickets |
|---|---|---|---|---|
| **Application** | FormCentral | 2,962 | 132,313 | 15% |
| | Healthcare Apps | 3,839 | | |
| | MyIT | 57,573 | | |
| | (…) | 67,939 | | |
| **Collaboration** | MobileIT | 20,229 | 155,612 | 17% |
| | Voice Fixed | 26,942 | | |
| | Messaging | 58,721 | | |
| | (…) | 49,720 | | |
| **ERP** | SAP BW | 1,113 | 19,924 | 2% |
| | SAP | 1,788 | | |
| | SAP KFP | 6,375 | | |
| | (…) | 10,648 | | |
| **Hosting Services** | Managed Application | 12,264 | 89,362 | 10% |
| | EAGLE_DC | 16,085 | | |
| | Managed Filespace | 23,412 | | |
| | (…) | 37,601 | | |
| **Support** | Service Desk | 4,444 | 47,594 | 5% |
| | Project Support | 7,919 | | |
| | Service Request | 24,214 | | |
| | (…) | 11,017 | | |
| **Software** | Other | 1,574 | 81,741 | 9% |
| | Software Asset | 80,167 | | |
| **Workplace** | Client Device | 5,851 | 151,418 | 17% |
| | Virtual Client | 9,246 | | |
| | Client Device | 48,026 | | |
| | (…) | 88,295 | | |
| **Network** | Wide Area Network WAN | 5,909 | 53,617 | 6% |
| | Local Area Network | 13,478 | | |
| | Remote Access Services | 25,881 | | |
| | (…) | 8,349 | | |
| **Security and Access** | Security | 15,160 | 121,196 | 14% |
| | AD Domain Services | 50,964 | | |
| | PKI Services | 55,072 | | |
| **Output Management** | OMS Print Canon | 2,615 | 44,508 | 5% |
| | OMS Print Ricoh | 6,938 | | |
| | Office Printing | 34,955 | | |
| **TOTAL** | | | **897,285** | **100 %** |

### *1.2.4 Development and Demonstration*

To achieve the goal of this research and to answer the proposed research questions, we propose the development of a method capable of automatically categorize incidents without human interaction. To design our artifact, we will follow the steps listed below:

- First, we need to determine which are the attributes that have a better impact on the classifier, which means the attributes that outputs the greater results, categorizing the incidents as correctly as possible. These attributes will serve as input to the next step.
- Using the dataset with the best attributes defined previously, we will apply text pre-process of data with different algorithms to obtain the best classifier. In this step we will present the results obtained with the application of the different algorithms and the NLP and TM techniques used to produce the results and to conclude which is the best method for the categorization.
- Having completed the previous step, it is present the categorization for the first and second levels of categories, using the proposed method. For the second level, we use three different approaches to conclude which better results on our data.

To test our artefact and discover if it fits the purpose, we will use the dataset referred previously, that was provided by a company that due to privacy questions, cannot be mentioned along this dissertation.

The datasets are composed by incidents correctly classified, with an appropriate category and subcategory. Using these datasets, we will experiment several TM and NLP techniques and ML algorithms described previously in the related work section, to be able to assess the performance of them on the respective data.

### 1.2.5 Evaluation and Communication

To evaluate our implementation, we will use the obtained results with the demonstration process to compare with the performance of manual categorization. After selecting which method has better performance, we will answer how this method improves the IM process in comparison to the manual categorization. Finally, we will verify if our method answers the proposed research questions and achieves the defined goal of the research.

Concerning research communication, a part of this research is presented by two papers and the whole research is represented by this document.

## 1.3 Document Structure

The remainder of this dissertation consists of six 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.

The steps took, and the approach adopted to achieve the research proposal are present in the third chapter.

In the fourth chapter, it is present the development and demonstration of our implementation and the respective results.

In the fifth chapter, the solution is assessed to understand how impacts the problem.

In the sixth chapter, our conclusions, possible future work and the felt limitations along dissertation are presented.

# 2 Theoretical Background and Related Work

This chapter presents relevant literature related to the proposed research. The literature discussed here represents and serves as a guideline to the research development and in order to achieve the defined goals. In section 2.1, we describe in more detail the IM Process and the IC. Section 2.2 section presents the Text Categorization (TC) process and their inherent steps, as well as the techniques and algorithms included in the whole process. In section 2.3, we describe how to assess the classifiers' performance. In section 2.4, to conclude the chapter, we introduce the several applications of similar studies in IC.

## 2.1 Incident Management and Incident Categorization

An incident is defined by ITIL as "An unplanned interruption to an IT service or reduction in the quality of an IT service". These incidents can be related with failures, questions or queries and should be detected as early as possible (Cannon & Wheeldon, 2007).

IM is the process responsible for managing disruptions, thus being a crucial factor in creating a high scalable system (Abbott & Fisher, 2009) as well as being responsible for restoring the normal operation, finding as quickly as possible a resolution for the incident, and minimizing business impact (Cannon & Wheeldon, 2007). To attain the success and efficiency of the process, there are four critical success factors that must be achieved such as quickly resolving incidents, maintaining IT service quality, improving IT and business productivity and maintaining user satisfaction (Steinberg, 2013). So, when a disruption on the system is detected, by the system or by users, several activities follow (Cannon & Wheeldon, 2007). Table 2.1 describes the IM activities.

*Table 2.1: Incident Management Process Activities*

| Activity | Description |
|---|---|
| **Incident Detection and Recording** | An incident must be recorded as soon as possible after being detected, if possible before user's damage. |
| **Classification and Initial Support** | **Incident Categorization:** The incident type should be correctly assigned to the incident. |
| | **Incident Prioritization:** This process deals with attributing urgency and impact to an incident. |
| **Investigation and Diagnosis** | In this step, incident escalation is performed, which includes an initial diagnosis to find a resolution. If the resolution is identified, the incident is solved, otherwise the incident is escalated for another support resolution group. |
| **Resolution and Recovery** | In this step, the previously identified resolution must be tested in order to ensure the system is operational. |
| **Incident Closure** | In the closing of an incident it is necessary to check if the categorization done in the second step is correct, if users are satisfied with the respective resolution, and if the documentation related to the incident is correct. |

Our research focuses on automating the IC process included in the second activity.

IC is one of the sub-activities of Incident Classification that is one of the first steps in IM process (Gupta et al., 2009). IC has the purpose of assigning incoming incidents to the most suitable category, which in turn allows to automatically assign the incident to a specific resolution group (Cannon & Wheeldon, 2007).

IC is also useful for reporting, improving the clarity and granularity of data. Ordinarily, ITS has its predefined categories, but due to companies having their own business model in most cases the set of categories is customized to each company, making them unique.

In a common ITS, the categorization is done manually which implies an increased waste of time and it is error prone. Automating IC not only reduces the time spent in the IM process by IT teams, but is also leads to a more accurate classification (Gupta et al., 2009).

## 2.2 Text Mining and Text Categorization

TM is the process responsible for identifying and extracting useful information from unstructured text (Vijayarani, Ilamathi, & Nithya, 2015). This involves text analysis, categorization, clustering, and visualization. Through several TM techniques it is possible to deduce patterns and knowledge from text (Rokach & Maimon, 2008).

TC, also known as, text classification is one of the applications of TM (Vijayarani et al., 2015) and is the process that deals with the assignment of pre-defined categories, topics, or labels to NL texts or documents (Sebastiani, 2002). Automated TC is a supervised learning task (Yang & Liu, 1999) that uses ML in order to learn how to classify from examples that perform the categorization automatically (Joachims, 1998b). Given a set of documents $D = \{d_1, \ldots, d_n\}$ with assigned categories $C = \{c_1, \ldots, c_n\}$ and a new document $d$, the main goal is to predict which category should be assigned to document $d$.

There are several approaches used on TC, which differ on how they represent documents and decide to assign a category to a document (Cardoso-Cachopo & Oliveira, 2003). TC is divided in two types of classification: binary and multi-class. A binary problem is when a document is assigned to one of two categories. A Multi-class problem is composed by two problem types, single-label and multi-label. The first consists of assigning to the document to exactly one of the pre-defined categories. In multi-label classification, the documents are assigned more than one label at the same time (Wang, 2008).

The generic main steps of the automatic TC are the document pre-processing, the feature selection and extraction, model selection, and finally training and testing the classifier (Dalal & Zaveri, 2011).

### *2.2.1 Text Pre-processing and Feature Selection*

Text Pre-processing is the first step in the TC process (Vijayarani et al., 2015), which starts with the tokenization technique. In this technique the text is split in conformity with pre-defined delimiters. Then, techniques such as stop-word elimination are used, which is the process that eliminates the words that are not meaningful for classification, and stemming, which then reduces words to their base form (Son, Hazlewood, & Peterson, 2014; Srividhya & Anitha, 2010; Dalal & Zaveri, 2011).

After the data preparation process is finished, the main goal is representing each document as a feature vector, selecting the terms that are relevant to identify a document and removing features that are irrelevant to the classification, causing dimensionality reduction of the dataset. One of the used methods in this step is the term frequency-inverse document frequency (TFxIDF) (Son et al., 2014) and consists on assigning to each term a weight based on the frequency of the term in the document. This weight increases with the number of times the term occurs, but is offset by the frequency of the term in the corpus (Altintas & Tantug, 2014). This algorithm is the most used in literature due to the performance achieved in different TC tasks. There is also a technique called Named Entity Recognition (NER), which consists on finding expressions like people’s names, organizations, or entities, and add value to the text analysis. Instead of treating words with no connection to the rest of the text, this technique allows an understanding of the context and an improvement of the analysis performance (Mohit, 2014).

With this process, it is obtained a smaller dataset and consequently lower computational requirements are needed for the TC algorithms, which is crucial to achieve success in this stage (Ikonomakis, Kotsiantis, & Tampakas, 2005).

### *2.2.2 Cross-Validation Model*

The Cross-validation process is a popular strategy ordinarily used in predictions. This technique is used for model selection, allowing to assess the performance of the resulted model. There are different options related to Cross-Validation and one of them, K-fold, consists in dividing randomly the whole training set into *n* subsets of equal size. One subset is used to test the classifier, which means to obtain the predictions, while the *n – 1* subsets of the training dataset is used to obtain the classifier. (Arlot & Celisse, 2009; Chih-Wei Hsu, Chih-Chung Chang, 2008).

### *2.2.3 Classifiers*

Most of the methods used for classification are also used for TC: for example decision trees (DT), support vector machines (SVM), Naïve Bayes (NB) and K-Nearest Neighbours (KNN) classifiers (Aggarwal & Zhai, 2013). These algorithms are used in Supervised Learning, when instances (inputs) in the dataset are assigned to known and correct labels (outputs). Using Supervised Learning classifiers, the dataset instances are learned, and the process is repeated various times, resulting several classifiers. Then, it is taking the vote of the different classifiers and is predicted the correct output of a new instance (Kotsiantis, 2007).

#### 2.2.3.1 *Support Vector Machines*

SVM (Joachims, 1998a) is a binary method of supervised learning introduced in TC, between 1998 and 1999 by Joachims. SVM consist on mapping input vectors into a high dimensional space and outputting the creation of a hyperplane (Vapnik, 2000). With the training data, this algorithm returns the optimal hyperplane, which separates data. As the problem of this work be multi-class, there are several approaches to solve these problem types: One-against-all build the same number of binary classifiers as the number of classes. Each classifier separates a certain class from the rest of the others. The predicted class is obtained according to the highest classifier output; other approach is One-against-one, that consists with $n$ classes, create $n(n-1)/2$ classifiers, which means the creation of a classifier for a pair of classes. The predicted output is obtained by the votes from classifiers (Braun et al., 2010). “SVM are particularly promising because they are very accurate, quick to train and quick to evaluate” (Dumais, Platt, Heckerman, & Sahami, 1998).

#### 2.2.3.2 *K-Nearest Neighbors*

KNN algorithm uses most of times the Euclidean distance to identify which are the *K* nearest neighbors of the instances, however it can also use other similarity measures (Duneja & Puyalnithi, 2017). The class of each instance is determined using a majority vote. ML algorithms and is a popular one in TC (Song, Huang, Zhou, Zha, & Giles, 2007).

#### 2.2.3.3 *Decision Trees*

DT can be used to different ends, being one of them classification, so they can be used to classify an instance to a predefined set of classes.

A rooted tree is a directed tree with a root node. The tree is composed by several nodes that represent features. Each one has an incoming edge, which represents a decision or rule. All the nodes with outgoing edges are nominated internal or test nodes. The final ones are the decision nodes or leaves that represent the outcome. The outcome is calculated according to a certain discrete function through all the input attributes (Rokach & Maimon, 2008).

#### *2.2.3.4 Naïve Bayes*

The NB algorithm uses the Bayes' rule to predict instance's most likely class.

$$P(c|x) = P(c)\frac{P(x|c)}{P(x)} \tag{2.1}$$

In the equation, *x* represents an instance and *c*, a category. With this measure, it is calculated the estimate from training documents. To calculate the probability of each class and to select which is the class with higher probability, the attributes that compose the instances (features), are all considered independents of each other and it is calculated for each feature the probability of belong to a certain category (Mccallum & Nigam, 1997). After calculating the probability for each feature of *x* is possible obtain the probability of the instance *x* to belong to the category c. The category with highest probability is the output of the classifier.

## *2.3 Evaluation*

The classifiers performance must be measured, and in the classification area the performance is usually measure with resort to an error rate. The classifier predicts the category to assign to the instance. If it predicts the correct category, it counts as a success; otherwise it counts as an error. "The error rate is the proportion of errors made over a whole set of instances, and reflects the overall performance of the classifier" (Altintas & Tantug, 2014).

Other metrics that measure the performance are the precision and recall. Dependently of the cases, it is more important to have a higher precision or a higher recall rate.

The precision measure defines how accurate the model is, i.e. quantifies how many instances classified to the target class do in fact belong to the target class. The recall quantifies how many instances are categorized to the target class of those who actually belong in it.

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \tag{2.2}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP + False\ Negative\ (FN)}} \tag{2.3}$$

The F1-Measure considers both precision and recall. The best value for this metric is 1, which means a perfect precision and recall.

$$\mathrm{F1 - Measure} = 2 \times \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision + Recall}} \tag{2.4}$$

## *2.4 Text Categorization Applications in Incident Categorization*

Over the years, approaches that automate the IM process have been studied and developed. One of these approaches is automating incident classification which is the purpose of our research. In this section we describe some work developed in this area and which results were obtained with the respective implementations.

Gupta et al. (2009) focused on automate IM process, specifically in automate the incident ticket classification. The classification of incidents embraces assigning a category, as well as a priority and an impact to an incident. In this research, they present two approaches for automating classification, consisting on the analysing of the incident descriptions written in NL. The analysis is made of two approaches. One of them is knowledge engineering, which is based on a set of rules created by experts. These rules use AND / OR relations and consist on mapping attributes of incident descriptions to corresponding incident categories. When an incoming incident arrives, the respective category is assigned through the defined rules.

The other is based on ML, in which a classifier is automatically created using pre-classified incidents. They denote a sequence of words as incident features and then use Naïve Bayesian classification. The probability of assigning a category to an incident is calculated for all categories. Finally, the incoming incident is assigned to the category that has the maximum value. To ensure that the system classification is done as correctly as possible, a category is only assigned if the probability overcomes a defined threshold.

The authors achieve 70% accuracy with 1,000 features. In agreement with an analysis of IBM internal tools based and built on similar conditions, it is indicated a reduction of tickets' resolution times by over 25%.

Son et al. (2014) to automate XSEDE ticket system focused on automating incident classification. For that, the authors resort to the NB algorithm. Related to NB, one of the variants is Multinomial NB, which was used in this research. Multinomial NB considers the word frequency in the documents, taking into account the words sequence, which sometimes improves the classification. The authors use for training a dataset composed by 7042 tickets and 717 for the test set. The goal is to assign a ticket to a tag. The tag indicates which category the ticket should be assigned to. The algorithm uses as input to the classifier, a word list composed by email subjects. The reason to the authors did that was because the text contained in the ticket subject is much more condensed than in the ticket body. With this text they also used TFxIDF and stop-words removal. Another algorithm used was Softmax Neural Network (SNN). This algorithm also calculates the probability of a feature belongs to a category. One more time, the category with higher probability is the chosen category. The results using Multinomial NB was ~70% of accuracy and using SNN was ~68%;

Agarwal et al. (2017) proposes the development of a tool called SmartDispatch. Their work is based on ticket descriptions. They used weighted vectors of terms to build a classifier and on SVM. They achieved performance between 69% and 81% of accuracy in the three datasets. The three datasets had different domains. Based on these results, they decided to assign a category only if the probability was greater or equal to 90%. and consequently, obtained better results. However, they noted that for two of the three datasets was possible classify 55% and 62% of the incident tickets with a probability higher than 90%, whereas for the third dataset was only possible classify 25% of the incident tickets. So, they decided to develop a new classification approach called discriminative term approach (DTA) which is based on the IDF technique but use SVM. In this approach, they defined the classifier to assign the resolution group with higher probability than 90%. This approach leads to results from 59% to 72% accuracy and 100% precision for the three datasets. Then, they decided to use the best of DTA and SVM, which meant that when an incident ticket arrived the DTA was used, but if there was no group resolution with a score higher or equal to 90%, SVM is used to know which groups have the higher score. And here SVM outperforms DTA.

Altintas & Tantug (2014) propose an extension to integrate into an ITS, which consists on assigning tickets to the suitable person of the support team. If the prediction confidence is greater than a defined threshold, the ticket is assigned to the predicted category, otherwise it is manually categorized. The tickets are composed by date, user, category, subcategory, and subject. The last attribute consists in a NL text and this is the critical attribute to the categorization process. The authors apply feature extraction, using TFxIDF and stop-words removal. To classify the tickets, they experimented with 4 algorithms: SVM, NB, KNN and DT. SVM achieves the higher performance, with 86% of accuracy.

A summary of the approaches described previously are present in the following Table 2.2.

*Table 2.2: Comparison of similar studies*

| No./Authors | | 1/(Gupta et al., 2009) | 2/(Son et al., 2014) | 3/(Agarwal et al., 2017) | 4/(Altintas & Tantug, 2014) |
|---|---|---|---|---|---|
| **Algorithms** | SVM | | | X | X |
| | KNN | | | | X |
| | DT | | | | X |
| | NB | X | X | | X |
| | SNN | | X | | |
| **NLP Techniques** | Tokenization | X | X | X | X |
| | Stemming | X | X | X | X |
| | Stop-words | | X | X | |
| | TFxIDF | X | X | X | X |
| | Lower-case tokens | | | X | |
| | NER | | | X | |

These papers serve as support to our research, and as to be able to propose the best method, we will use all algorithms and techniques described in Table 2.2 to conclude which are the ones that have the greatest impact on our data and propose the method that best fit in the specific data.

# 3 *Design*

This chapter details the design of the proposal. Section 3.1 describes the process to determine the attributes for the categorization. Section 3.2 describes the text pre-processing which involves all TM techniques applied. Section 3.3 presents the application of the classifiers for the first level of categories. Section 3.4 Presents the approaches used for the second level categories, applying the best output method obtained in the previous section.

All steps to obtain and propose the best method to categorize the incidents automatically, that involve discovering the most suitable attributes, the application of the TM techniques and the respective algorithms, are described above in detail. Figure 3.1 presents the workflow that serves as support to the development and demonstration step.

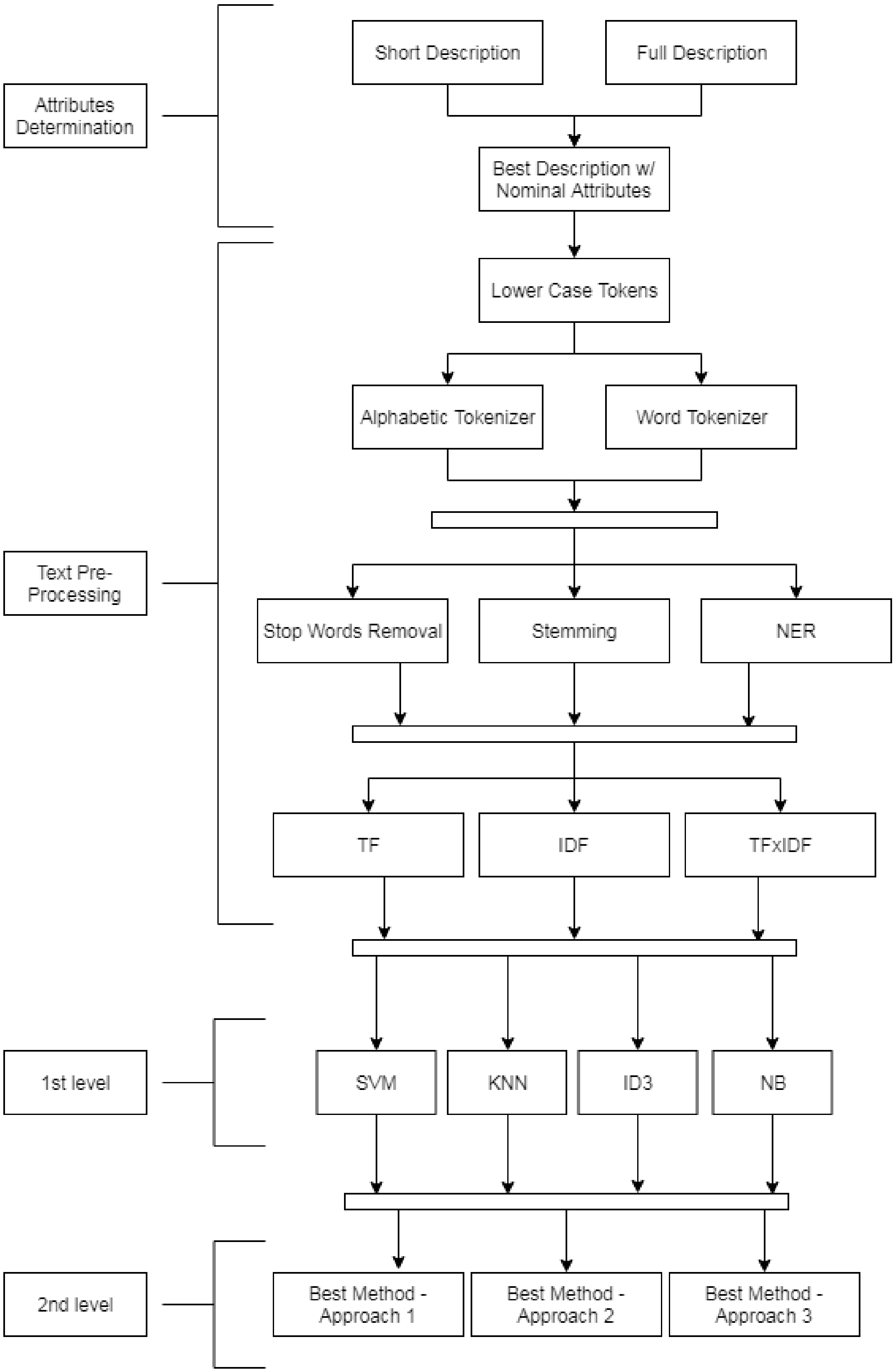


*Figure 3.1: Design Workflow*

## 3.1 Attributes Determination

Even though the dataset has incidents written in several languages: English, German, Spanish, French, and Portuguese, we have decided to choose English as the one to be studied and exclude all the incidents in other languages from our analysis, being that English is the most common one.

We start by analyzing which attributes are the most relevant for the IC process. Each incident ticket has, in addition to the descriptions, the following information: its severity, which is divided in three levels: 1 refers to a critical state, 2 to medium and 3 to low. The other attribute is its location, which is the geographical location of the user that opens the incident. The attribute is represented by different abbreviations. Each abbreviation represents the country and the respective city. To choose the most suitable attributes for the classifier performance, we need to classify the data taking into account the different attributes, to be able to compare the different results and conclude which are the attributes that has higher impact on data. In this step, we will focus on the difference between considering only the nominal attributes and considering both descriptions separately. The full is a detailed description of the incident with 35 words approximately, while the short one is a focused description with 6 words approximately. This is one crucial step in order to get a good performance of the classifier, since we are defining the attributes that better characterize the categories. Table 3.1 shows an example of one incident ticket that composes the used dataset.

*Table 3.1: Incident ticket example*

| | |
|---|---|
| **Full Description** | “Please could you reset the Windows password and unlock the account?” |
| **Short Description** | “Windows password reset” |
| **Severity** | 3 – “Low” |
| **Localization** | MEX EC |
| **Category** | Security and Access |
| **Subcategory** | Identity and Account Management |

In order to verify which are the attributes with the biggest positive impact on categorization, we will use only the first level of categories.

In all approaches we will consider 5,000 incident tickets per category, in order to have the same amount of data to each category and to the results not be affected by the frequency of each category in the dataset. For this train we will use SVM and KNN algorithms and cross-validation technique with 10 folds. With the application of these two algorithms is possible compare the results and be sure which are the attributes with better impact for the categorization. In this step, we use SVM and KNN, since the algorithms show good performances with textual data. In the first approach is only used as attributes, the severity and the location. In the second is only the incident short description and finally, in third is only the incident full description.

To train the classifiers with SVM, we need the kernel type and the *C* parameter. The *C* represents the hyperplane margin, high values of *C* represent a small margin and consequently a misclassification rate lower on the training set, while low values represent a high margin and a misclassification rate higher. The only parameter of KNN is the number of neighbors. In this step we focus on test different parameters of the algorithms in order to achieve the best performance of them, instead of exploiting different parameters. The goal is test which attributes we should consider to the categorization. In all approaches we only apply one TM technique to the both descriptions, the Alphabetic tokenization. As further study other possibilities are present in the next section.

## 3.2 Text Pre-Processing

Concluding the relevance of textual data for the IC, we move to the first step of TC, the text pre-processing. We start by analysing two different tokenization strategies: alphabetic and word tokenization. The alphabetic tokenizer is a simple tokenizer that only considers tokens composed by alphabetic sequences. The word tokenizer is a standard word tokenizer that splits words according to predefined tokens, such as space, punctuation, etc. Then we analyse the application of transforming all tokens contained on descriptions into lower case. Another aspect that we have explored is stop-words removal, that consists on removing words which ordinarily do not improve the classification performance. Ordinarily, stop-words are words commonly used to connect speech like propositions. Another explored technique is stemming, which consists on reducing the words to their base form, thus lowering the number of entries of the dictionary. We also explored named-entity recognition, focusing on the identification of organizations and used them as features to improve the categorization. Finally, the last aspect that we have explored was the descriptions representation. In that sense, we represent descriptions as feature vectors of term frequencies (TF), $\log(1 + f_{ij})$, $f_{ij}$ is the frequency of word *i* in document *j* (other dampening strategies could be used); inverse document frequencies (IDF), log(num of Docs/num of Docs with word i); and, TFxIDF. TFxIDF increases with the number of times a term occurs in a document but is offset by the document frequency of the term in the corpus. This technique turns possible give weight to the terms that better categorize the incidents.

## 3.3 First Level Categorization

In order to train the classifiers, we use the algorithms described in the related work, which are the most suitable for our problem. As we have a labelled dataset, the approach that we use is based on supervised methods. Therefore, we use SVM, KNN, DT and NB, which were the algorithms used in the related work and which produced good results. One more time, we used cross-validation with 10 folds. The purpose is to compare the different algorithms using the several TM techniques described in sub-section 3.2 and verifying which algorithm and which techniques present the best classifier performance.

Related to this section, four sub-sections are presented, each one related with an algorithm. Along each sub-section the application of several TM techniques is present. At the end are present for each algorithm the techniques that present the better results.

At the end of this step it is expected to obtain the method that best fits on assigning a category to an incident.

## 3.4 Second Level Categorization

Concerning the second level of categories, we considered 617 or 618 incident tickets by subcategory, which composes a dataset with 50,000 incidents. One more time we defined the same number of incidents, but now by subcategory. With this dataset it was explored three different approaches. The first one is performing the categorization assuming that the first level category is correctly assigned to the incident. Basically, we use the first level category as an attribute to build the classifier that assigns the second level category to a given incident. In this approach we try understanding if there are assigned subcategories that do not belong to the respective category.

The second approach does not take into consideration the first level categorization. Therefore, the incident is categorized with the same data that we use in the first categorization, but instead of assigning a category, it is a subcategory assigned, which means the category is not seen as it is in the first approach. Related to this approach, it is important to note that when the classifier automatically assigns a second level category, we are also automatically assigning the respective first category.

In the third approach, we divide the data by category, obtaining ten datasets, where each one is related to a category. So, in this last approach the incidents can only be assigned to a subcategory that has a category that includes the assigned subcategory. This approach helps to understand the type of errors found in the first approach.

In the three approaches, we build classifiers using the same method that achieved the best performance in the first categorization. Nevertheless, the difference between the data used in the first level and in the second level can influence the classifier performance. This reason leverages us to present three approaches mentioned previously to categorize the incidents with a subcategory. Those three approaches are based in the way that the dataset is used.

# 4 Development and Demonstration

In this chapter, we present all the results after applying the steps described in the Design chapter. For that, we will use the dataset provided, which is a real environment, to obtain the most suitable method. In section 4.1, we present the results to conclude which are the relevant attributes for the categorization process. In section 4.2 are present the results related to the first level categorization, using the described four algorithms and the several TM techniques are detailed in this section. Section 4.3 presents the results related to the second level categorization.

## 4.1 Attributes Determination

Figure 4.1 and 4.2 present the achieved results with the application of SVM and KNN algorithms, respectively. In these results, it is possible to compare the use of only nominal attributes, such as the severity and location, only the full description, and only the short description. The TM technique used here was only the Alphabetic tokenizer with the full and short descriptions. After verifying which is the best attribute, it is possible combine it further with the other attributes.

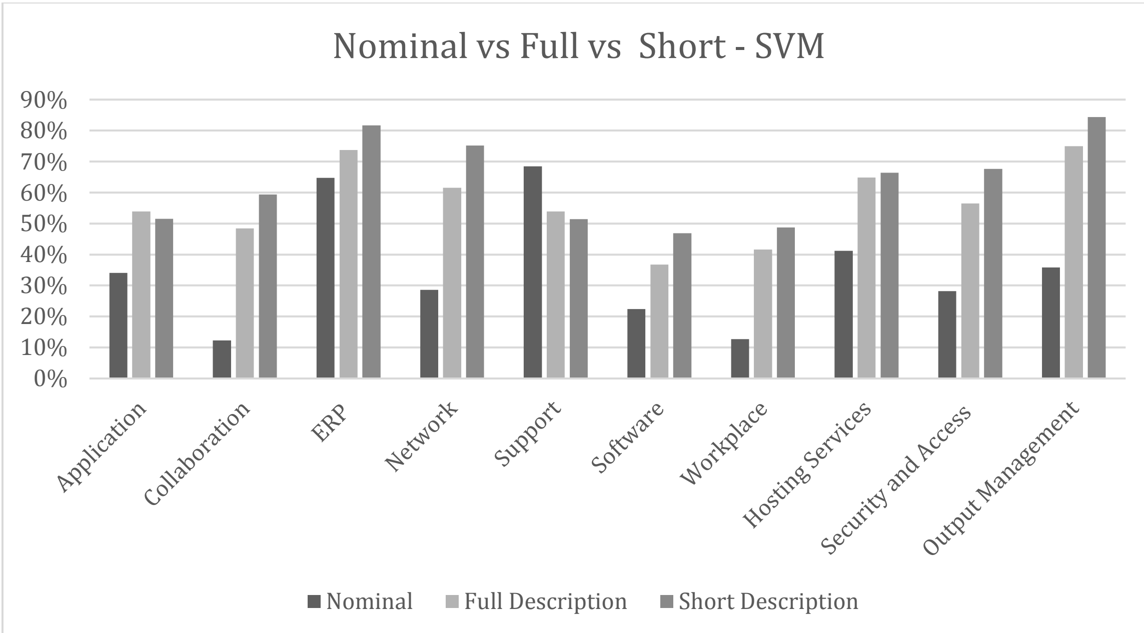


*Figure 4.1: Accuracy of nominal attributes vs full description vs short description using SVM*

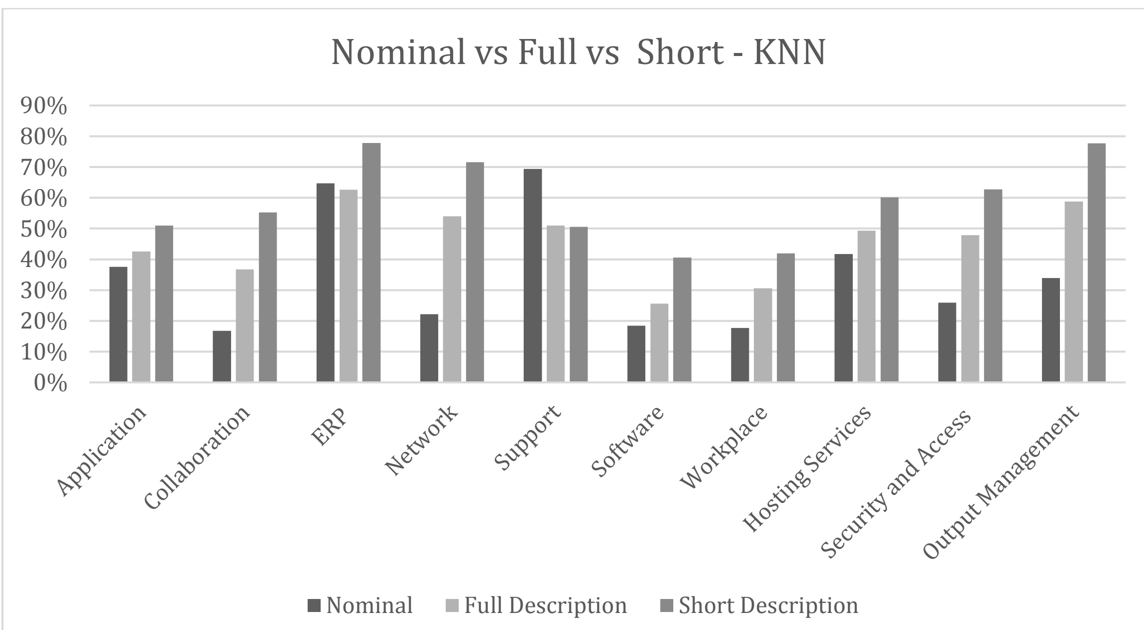


*Figure 4.2: Accuracy of nominal attributes vs full description vs short description using KNN*

In both figures, is possible to observe the relevance of textual data for a correct categorization. In most categories, using the full and short description leads to a greater accuracy, especially when using only the short description. It is peculiar that the support category with both algorithms presents a better accuracy when using only nominal attributes, alternatively of what happen with the other categories. ERP category is also a peculiar case, obtaining a higher accuracy using only nominal attributes than with the full description. So, we analyzed the number of locations for each category, which is presented in Figure 4.3.

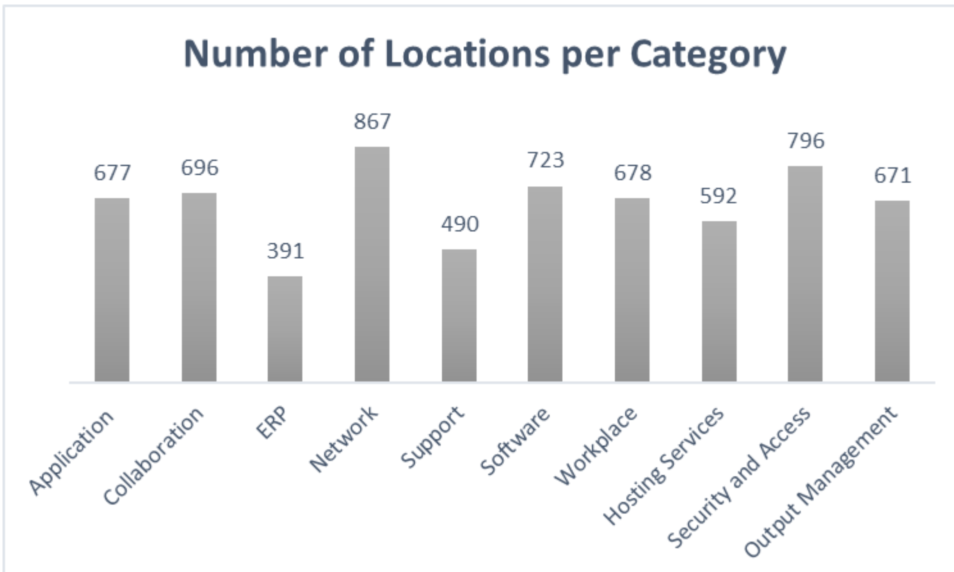


*Figure 4.3: Number of locations assigned to each category*

Analyzing the number of locations assigned to each category it is interesting observe that the categories of support and ERP are also the categories with the lower number of locations. The reason to both categories has this higher accuracy with the nominal attributes can be due the fact that are incidents very characterized of specific locations which turns possible a well categorization with the nominal attributes, against of what happen with the others.

Since the accuracy metric, is not enough to evaluate the classifier performance, given that the metric only represents the rate of the TP and the FN, we also present precision and recall. For IC it is important to have both a high precision and a high recall. If precision and recall are low means that there are lower number of TP, which implies a lower classifier performance. So, Table 4.1 and 4.2 present the precision and recall related to SVM and KNN algorithms.

*Table 4.1: Precision and Recall of nominal attributes vs full description vs short description using SVM*

| | Precision | | | Recall | | |
|---|---|---|---|---|---|---|
| **Category** | **Nominal** | **Full** | **Short** | **Nominal** | **Full** | **Short** |
| **Application** | 25% | 38% | 53% | 34% | 54% | 52% |
| **Collaboration** | 27% | 54% | 61% | 12% | 48% | 59% |
| **ERP** | 42% | 66% | 51% | 65% | 74% | 82% |
| **Network** | 32% | 66% | 78% | 29% | 62% | 75% |
| **Support** | 50% | 47% | 60% | 69% | 54% | 51% |
| **Software** | 22% | 43% | 51% | 22% | 37% | 47% |
| **Workplace** | 27% | 48% | 56% | 13% | 42% | 49% |
| **Hosting Services** | 42% | 65% | 73% | 41% | 65% | 66% |
| **Security and Access** | 31% | 62% | 68% | 28% | 57% | 68% |
| **Output Management** | 35% | 89% | 92% | 36% | 75% | 84% |

One more time is curious look to the results of ERP category and despite the higher accuracy was obtained with the short description, the higher precision was with the full description. Looking to the support is obtained, the best accuracy with the nominal attributes, however the higher precision was with the short description.

*Table 4.2: Precision and Recall of nominal attributes vs full description vs short description using KNN*

| | Precision | | | Recall | | |
|---|---|---|---|---|---|---|
| **Category** | **Nominal** | **Full** | **Short** | **Nominal** | **Nominal** | **Short** |
| **Application** | 26% | 38% | 44% | 42% | 43% | 51% |
| **Collaboration** | 27% | 33% | 51% | 14% | 37% | 55% |
| **ERP** | 45% | 59% | 51% | 66% | 63% | 78% |
| **Network** | 36% | 44% | 73% | 25% | 54% | 72% |
| **Support** | 52% | 44% | 55% | 75% | 51% | 51% |
| **Software** | 22% | 30% | 47% | 19% | 26% | 41% |
| **Workplace** | 29% | 38% | 52% | 17% | 31% | 42% |
| **Hosting Services** | 48% | 65% | 72% | 43% | 49% | 60% |
| **Security and Access** | 33% | 47% | 71% | 31% | 48% | 63% |
| **Output Management** | 36% | 67% | 84% | 37% | 59% | 78% |

One more time, looking for the ERP category is obtained the best accuracy with the short description although is obtained the higher precision with the full description. Related to the support is obtained the higher accuracy with the nominal attributes, however the higher precision was with the short description.

Concerning these results, we can observe that for both algorithms, the textual data represents an essential role for a correct categorization. For both algorithms, Support category has better accuracy when using only nominal attributes. Again, it is interesting to observe that only this category has higher accuracy with only nominal attributes than with the text attributes. However, when we analyze the precision for the Support category with the different attributes, is possible observe a higher precision when is used only the short description, instead of what happens with the accuracy.

However, is very interesting to verify that we achieve better accuracy using the short description when compared with the full description. A possible reason that we found to justify such finding might be the fact that when the user describes an incident with limited text that results in a greater focus on explaining the incident. On the other hand, in the full description the user has tendency to disperse. Table 4.3 presents all results with the nominal and text attributes.

*Table 4.3: Results with SVM and KNN*

| **Attributes** | **Metric** | **SVM** | **KNN** |
|---|---|---|---|
| **Nominal** | Accuracy | 35 % | 35 % |
| | Precision | 33 % | 35 % |
| | Recall | 36 % | 37 % |
| **Full description** | Accuracy | 57 % | 46 % |
| | Precision | 58 % | 47 % |
| | Recall | 57 % | 46 % |
| **Short description** | Accuracy | 63 % | 59 % |
| | Precision | 64 % | 60 % |
| | Recall | 63 % | 59 % |

After concluding that short description presents the best results, we decided to apply the same previous approaches but with the nominal attributes and short description together, with the nominal attributes and the full description, and finally with the all attributes. Table 4.4 presents the results of these approaches.

*Table 4.4: SVM vs KNN using nominal attributes and the short description*

| Attributes | Metric | SVM | KNN |
|---|---|---|---|
| **Nominal & Short Description** | Accuracy | 72 % | 60 % |
| | Precision | 72 % | 61 % |
| | Recall | 72 % | 60 % |
| **Nominal & Full Description** | Accuracy | 68 % | 47 % |
| | Precision | 68 % | 48 % |
| | Recall | 68 % | 47 % |
| **Nominal & Short & Full** | Accuracy | 71 % | 51 % |
| | Precision | 72 % | 52 % |
| | Recall | 71 % | 51 % |

As it is possible observe in both algorithms using the short description with the nominal attributes the metrics of performance are greater, increasing 8p.p. for SVM precision and 9p.p. for accuracy and recall. Related to KNN, the increased values are not so good as with SVM, increasing only 1p.p. for all metrics. Related to the nominal and full description and using SVM, the results increase in comparison with the short description, however are lower than when is used the nominal and short description. Using KNN, the results decrease compared with the short description results. Concerning the use of all attributes and using SVM, the results are greater than with only the short description, however when it is compared with the results of the nominal and short description is possible to note similar results, but still lower. The same does not happen with KNN, and lower results are obtained in contrast to the results with only the short description.

With these results, we conclude that is better use the nominal attributes and the short description, against of use only the short description.

## 4.2 First Level – Categories

This section is divided in four sub-sections, where each one is related with a supervised learning algorithm. Each sub-section presents the results of the applications of the different techniques used. As described previously, this process includes tokenization, stop-words removal, stemming, NER, and TFxIDF. These techniques will be applied considering the nominal attributes and the short incident description, which are the best attributes for IC as we had concluded before.

### 4.2.1 Support Vector Machine

The first algorithm that we use is SVM. For that, we need the following arguments: a matrix with values, where each row represents an incident ticket, the type of the kernel, and the C parameter. There are no rules to define the values of the last two parameters. The only way to choose the best parameters value is try out possible ways and conclude which are more appropriate for this task. However, in this step the goal is to verify which techniques show better results in the feature selection process for this algorithm, therefore we use for the SVM parameters always the same values. We apply as kernel type, the poly kernel and as *C* value, 0.5.

Figure 4.4 presents the results of the application of the two tokenizers: there was not a significative difference between both approaches, with the alphabetic tokenizer achieving an accuracy of 72% and the word tokenizer achieving an accuracy of 70%.

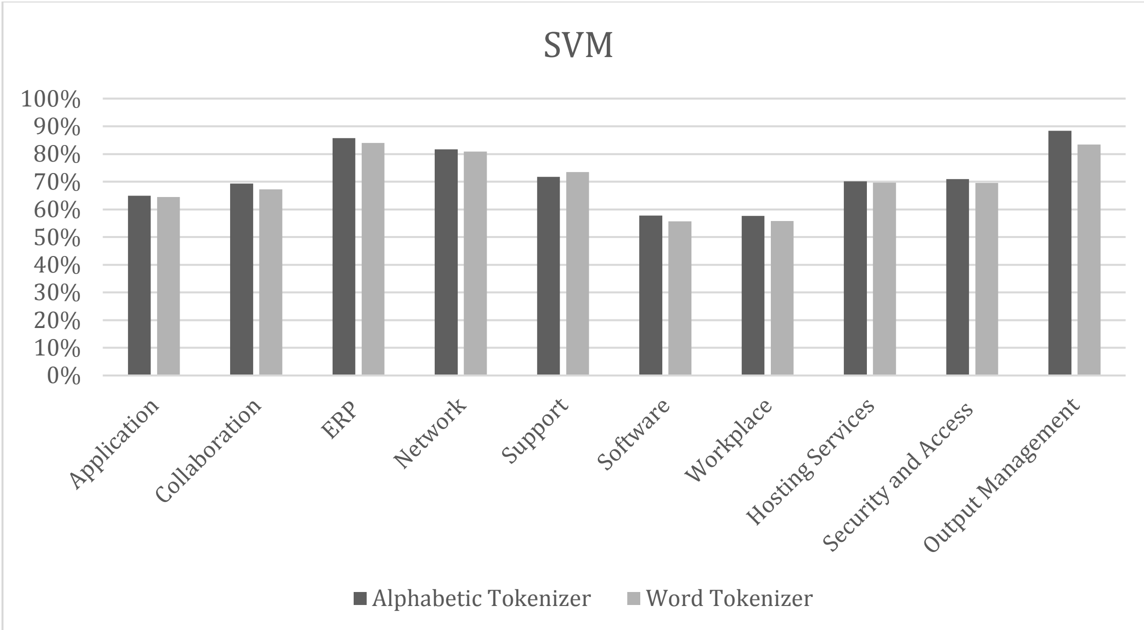


*Figure 4.4: Accuracy of Word Tokenizer vs Alphabetic Tokenizer with SVM*

To be able to really conclude which are the best tokenizer for our data, Table 4.5 presents the results, for all metrics for both tokenizers.

*Table 4.5: Results of Alphabetic tokenizer vs Word tokenizer with SVM*

| Tokenizer | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Alphabetic** | 72% | 72 % | 72 % | 72 % |
| **Word** | 70 % | 71 % | 70 % | 71 % |

With the information present in Table 4.5, we conclude that Alphabetic tokenizer has a better impact on our data.

As next step, we transform all the term tokens that resulted of the tokenization into lower case tokens, so same words written with upper or lower cases are considered as the same word. In this step, we want to understand if this technique presents better results. The average difference of the two approaches is 6% for accuracy, precision, and recall, being 78% for all values.

Using stop-words removal and stemming techniques does not lead to improvements. The reasons for this to happen can be the small quantity of words and the domain-specific language that composes incident descriptions. Therefore, in the short description there are very few words that connect the text, also that can be reduced to a common form, having no impact in the results.

We also apply Named Entity Recognition (NER) (Manning et al., 2014) on the short description. Our use of NER focus on the identification of entities and organizations. In general, a single entity is composed by more than one word and with tokenization the words are all split independently of the link between them. If there are entities composed by two or more words, the split is not done between these words, and consequently the IC process can be improved. However, this has no impact on classification, mainly due to the fact that the NER tool was not able to identify entities. Again the domain-specific language may be the major difficulty in this step. For this reason, this step will not be included in the application of the other algorithms.

As final step, we explore the application of three approaches in the previous resulted data: TF, IDF and TFxIDF. Using TF, the short descriptions are represented by feature vectors that contains the number of times a term occurred in the document. Table 4.6 presents the results obtained with the three approaches.

*Table 4.6: Results of IDF, TF, TFxIDF with SVM*

| Techniques | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Lower-Case & Alphabetic & IDF** | 80 % | 80 % | 80 % | 80 % |
| **Lower-Case & Alphabetic & TF** | 79 % | 79 % | 79 % | 79 % |
| **Lower-Case & Alphabetic & TFxIDF** | 79 % | 79 % | 79 % | 79 % |

As it is possible observe the 3 approaches produce very similar values, however the best accuracy is obtained using IDF: 80%, versus 79% for the other approaches.

The SVM classifier achieves the best results with the following TM techniques: Alphabetic tokenizer, lower-case tokens, and IDF.

### *4.2.2 K-Nearest Neighbors*

To run KNN, we need to define the number of *k*, which is the number of neighbors that the algorithm considers to attribute a category. The adopted value for k is 1, being the default value for this parameter. Figure 4.4 presents the accuracy results related with the word and Alphabetic tokenizer.

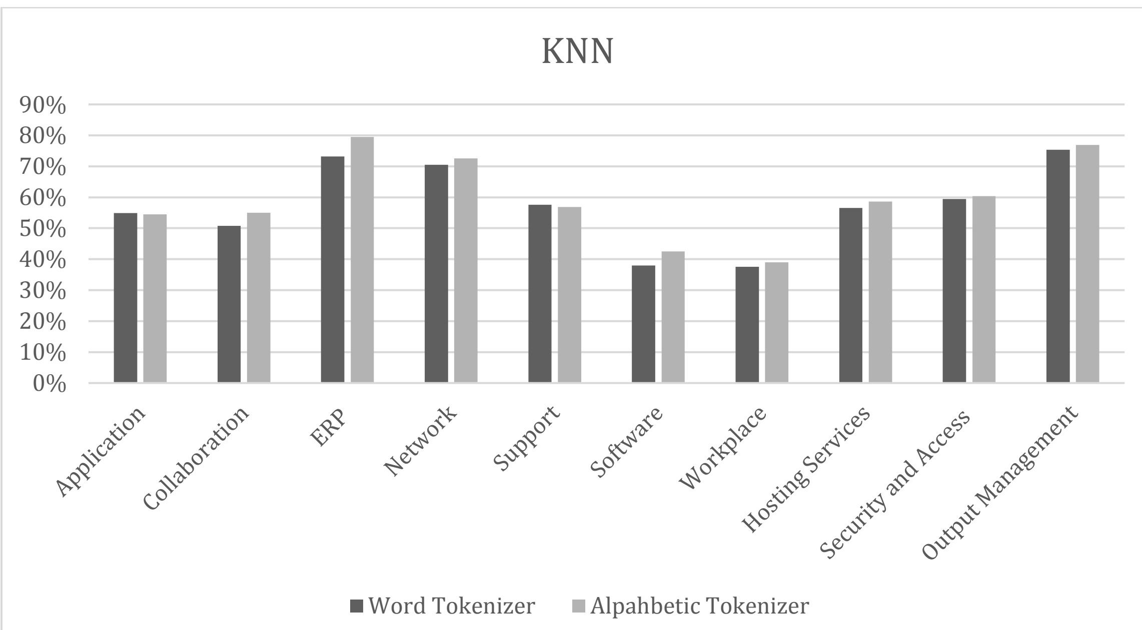


*Figure 4.5: Accuracy of Word Tokenizer vs Alphabetic Tokenizer with KNN*

One more time the results with Alphabetic tokenizer are higher. The use of the Word tokenizer leads to an accuracy of 53%, while with the Alphabetic tokenizer the accuracy is 57%. Table 4.7 shows the results of the other metrics.

*Table 4.7: Results of Word Tokenizer vs Alphabetic Tokenizer with KNN*

| Tokenizer | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Word** | 57 % | 59 % | 57 % | 57 % |
| **Alphabetic** | 60 % | 61 % | 60 % | 60 % |

As with SVM, with KNN the best results are obtained with the Alphabetic tokenizer.

Transforming all tokens into lower case increase each metric 3p.p.: accuracy, recall, and F-measure are 63%, and precision is 64%. These results are achieved using the Alphabetic tokenizer.

In addition to this processing sequence, we experimented stop-words removal and stemming, separately. With stop-words removal all the metrics improved 2p.p., while with stemmer, there were no changes.

As final step, we explore the use of TF, IDF, and TFxIDF. The results are present in Table 4.8.

*Table 4.8: Results of IDF, TF, TFxIDF with KNN*

| Technqiues | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Lower Case Tokens & Alphabetic Tokenizer & Stop-Words & IDF** | 52 % | 54 % | 52 % | 52 % |
| **Lower Case Tokens & Alphabetic Tokenizer & Stop-Words & TF** | 56 % | 58 % | 56 % | 56 % |
| **Lower Case Tokens & Alphabetic Tokenizer & Stop-Words & TFxIDF** | 56 % | 58 % | 56 % | 56 % |

None of the approaches leads to better results. The best result 65% of accuracy was achieved using the Alphabetic tokenizer, lower case tokens and stop-words removal.

### *4.2.3 Iterative Dichotomiser 3*

Iterative Dichotomiser 3 (ID3) is one of the most usual DT algorithms (Singh & Gupta, 2014). To evaluate this algorithm, we need to specify the number of folds, the confidence factor, and the minimum number of instances per leaf. For the parameter values, we define the default values of the algorithm, this means, 3 to the number of folds, 0.25 to the confidence factor and 2 to minimum number of instances per leaf.

In Figure 4.5, it is possible observe the accuracy after applying both tokenizers.

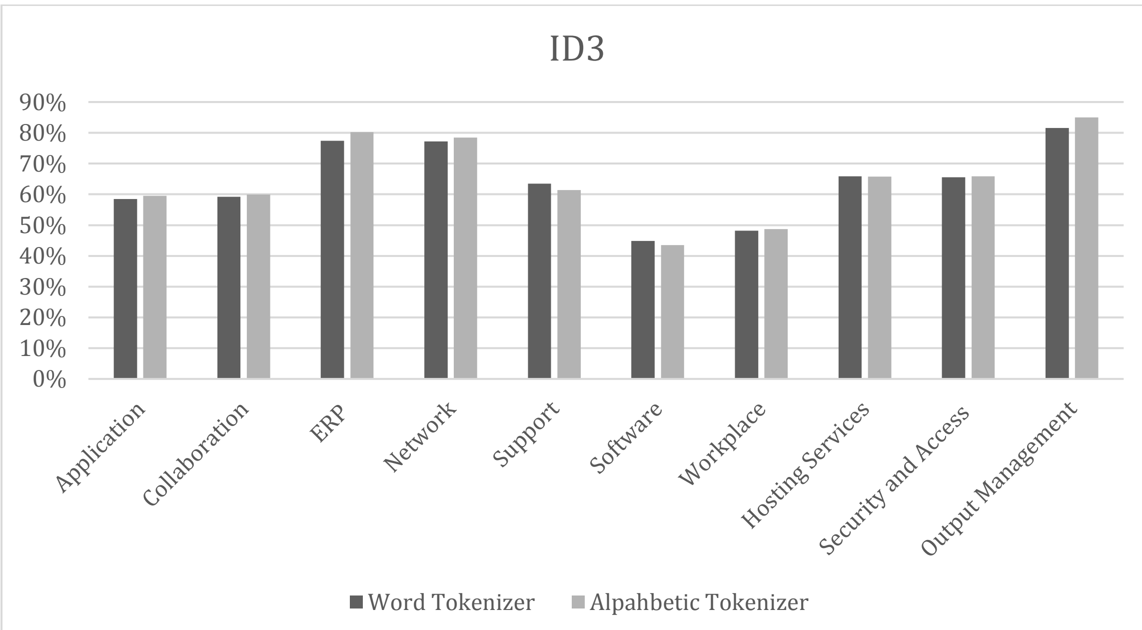


*Figure 4.6: Accuracy of Word Tokenizer vs Alphabetic Tokenizer with ID3*

One more time, the Alphabetic tokenizer presents the best accuracy. Table 4.9 presents the results of the other metrics.

*Table 4.9: Results of Word Tokenizer vs Alphabetic Tokenizer*

| Tokenizer | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Word** | 64 % | 65 % | 64 % | 64 % |
| **Alphabetic** | 65 % | 65 % | 65 % | 65 % |

The difference is minimum, but the best results were achieved using the Alphabetic Tokenizer. The use of lower case tokens increases accuracy, recall and F-Measure to 67% and precision to 68%. Stop-words removal and stemming did not lead to improvements. there is no improvement on results. Table 4.10 presents the results concerning the use of IDF, TF, and TFxIDF.

*Table 4.10: Results of TF vs IDF vs TFxIDF*

| Techniques | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Lower-Case & Alphabetic & IDF** | 67 % | 68 % | 67 % | 67 % |
| **Lower-Case & Alphabetic & TF** | 67 % | 68 % | 67 % | 67 % |
| **Lower-Case & Alphabetic & TFxIDF** | 67 % | 68 % | 67 % | 67 % |

None of these techniques lead to better results for categorization: the metrics values are the same with or without their application. The best results were obtained with the Alphabetic tokenizer and lower-case tokens.

### *4.2.4 Naïve Bayes*

Naïve-Bayes does not require parametrization. Figure 4.6 presents the results related to the Alphabetic and Word tokenizer.

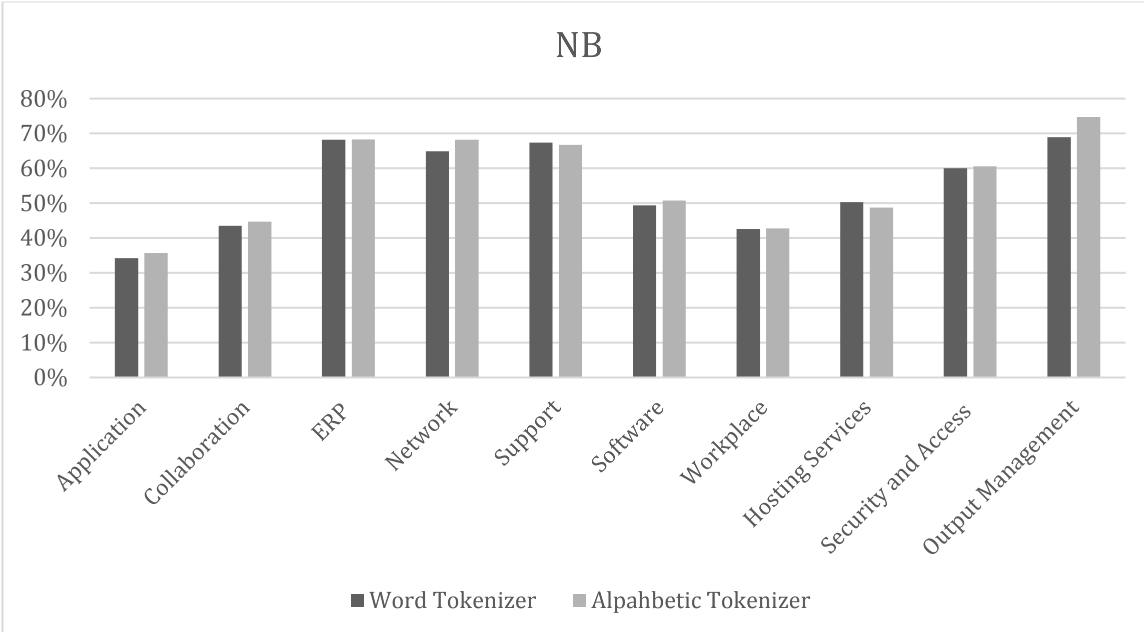


*Figure 4.7: Accuracy of Word Tokenizer vs Alphabetic Tokenizer with NB*

One more time, the accuracy results using the Alphabetic tokenizer are higher than with the Word tokenizer, however the difference is minimum as it is possible observe in Table 4.11.

*Table 4.11: Results of Alphabetic Tokenizer vs Word Tokenizer using NB*

| Tokenizer | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Word** | 55 % | 58 % | 55 % | 56 % |
| **Alphabetic** | 56 % | 58 % | 56 % | 57 % |

Applying lower case tokens, the results improved, the accuracy and recall increase to 59%, while precision and F-Measure increase to 60%.

With stop-words, the results improved, with accuracy and recall increasing to 60%, precision to 62% and F-Measure to 61%. With stemming, there is no improvement. The results related to IDF, TF and TFxIDF are shown in Table 4.12.

*Table 4.12: Results of TF vs IDF vs TFxIDF using NB*

| Techniques | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Lower-Case & Alphabetic & Stop-Words & IDF** | 60 % | 62 % | 60 % | 60 % |
| **Lower-Case & Alphabetic & Stop-Words & TF** | 55 % | 58 % | 55 % | 56 % |
| **Lower-Case & Alphabetic & Stop-Words & TFxIDF** | 55 % | 58 % | 55 % | 56 % |

There is no improvement with the application of these techniques. So, the best results were obtained with the Alphabetic Tokenizer, lower case tokens, and stop-words removal.

### *4.2.5 Overall Results*

The best results obtained with each algorithm are present in Table 4.13.

*Table 4.13: Summary of results per algorithm*

| Algorithm | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **SVM** | 80 % | 80 % | 80 % | 80 % |
| **KNN** | 65 % | 66 % | 65 % | 65 % |
| **ID3** | 67 % | 68 % | 67 % | 67 % |
| **NB** | 60 % | 62 % | 60 % | 61 % |

As it is possible observe, the best results are achieved using SVM algorithm, the techniques used to obtain the best results for each algorithm are present in Table 4.14.

*Table 4.14: Summary of best TM techniques per algorithm*

| Algorithm | Alphabetic Tokenizer | Lower case tokens | Stop-Words Removal | IDF |
|---|---|---|---|---|
| **SVM** | X | X | | X |
| **KNN** | X | X | X | |
| **ID3** | X | X | | |
| **NB** | X | X | X | |

The techniques that achieve better performance are not the same among algorithms, however SVM is the algorithm that achieves better performance.

# *4.3 Second Level – Subcategories*

Since SVM achieved the best results in the first level categorization, we will use it for the second level categorization.

## *4.3.1 First Approach – Data with assigned category*

In this section, we present the results of the second level of categories. This approach consists in considering the category as feature to categorize the incidents with a subcategory, using besides the short description and the nominal attributes, the respective category. Table 4.15 presents the average accuracy of the subcategories categorization for each category.

*Table 4.15: Results of subcategories using dataset with category*

| Category | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Application** | 80 % | 81 % | 80 % | 80 % |
| **Collaboration** | 80 % | 81 % | 80 % | 80 % |
| **ERP** | 84 % | 85 % | 84 % | 85 % |
| **Hosting Services** | 74 % | 74 % | 74 % | 74 % |
| **Network** | 82 % | 83 % | 82 % | 82 % |
| **Output Management** | 80 % | 80 % | 80 % | 80 % |
| **Security and Access** | 84 % | 84 % | 84 % | 84 % |
| **Software** | 97 % | 98 % | 97 % | 98 % |
| **Support** | 56 % | 57 % | 56 % | 56 % |
| **Workplace** | 72 % | 73 % | 72 % | 73 % |
| **Overall** | **79 %** | **80 %** | **79 %** | **79 %** |

With this approach the results average of all subcategories were 79% to accuracy, recall and F-Measure and 80% to precision.

### *4.3.2 Second Approach – Data divided in 10 datasets*

The approach consists in the difference of dividing the input data in ten sets, where each one contains data of only one category. With this approach, it is possibly to understand if there are incidents that were being assigned to subcategories which do not correspond to the correct category. The results are present in table 4.16.

*Table 4.16: Results of subcategories using separated datasets for each category*

| Category | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Application** | 79 % | 80 % | 79 % | 80 % |
| **Collaboration** | 80 % | 80 % | 80 % | 80 % |
| **ERP** | 83 % | 84 % | 83 % | 84 % |
| **Hosting Services** | 74 % | 74 % | 74 % | 74 % |
| **Network** | 82 % | 83 % | 82 % | 82 % |
| **Output Management** | 80 % | 80 % | 80 % | 80 % |
| **Security and Access** | 84 % | 84 % | 84 % | 84 % |
| **Software** | 93 % | 93 % | 93 % | 93 % |
| **Support** | 55 % | 56 % | 55 % | 55 % |
| **Workplace** | 71 % | 72 % | 71 % | 71 % |
| **Overall** | **78 %** | **79 %** | **78 %** | **78 %** |

Classifying each dataset related to a specific category lead to 78% for accuracy, recall, and F-Measure, and 79% for precision.

### *4.3.3 Third Approach – Data with no assigned category*

In this is present the results with no category as feature, which means the data used has no category attribute. Here the subcategory is predicted considering only the nominal attributes and the short description. Table 4.17 presents the average results of the respective subcategories of each category.

*Table 4.17: Results of subcategories without category as feature*

| Category | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Application** | 78 % | 74 % | 78 % | 76 % |
| **Collaboration** | 66 % | 72 % | 66 % | 68 % |
| **ERP** | 74 % | 74 % | 74 % | 74 % |
| **Hosting Services** | 59 % | 66 % | 59 % | 62 % |
| **Network** | 66 % | 68 % | 66 % | 67 % |
| **Output Management** | 69 % | 75 % | 69 % | 71 % |
| **Security and Access** | 53 % | 57 % | 53 % | 54 % |
| **Software** | 92 % | 79 % | 92 % | 85 % |
| **Support** | 44 % | 48 % | 44 % | 46 % |
| **Workplace** | 60 % | 64 % | 60 % | 62 % |
| **Overall** | **66 %** | **68 %** | **66 %** | **67 %** |

With the last approach, we achieved 66% of accuracy and recall, 68% of precision and 67% of F-Measure.

### *4.3.4 Overall Results*

The overall results related with the three approaches are present in Table 4.18.

*Table 4.18: Summary of results per approach*

| Approach | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **First** | 79 % | 80 % | 79 % | 79 % |
| **Second** | 78 % | 79 % | 78 % | 78 % |
| **Third** | 66 % | 68 % | 66 % | 67 % |

Analysing the results of the three approaches, it is possible observe that the first two obtained the best results. The results of the first and second approaches are very similar, which means that in the first, the incident tickets are not being assigned to subcategories which do not belong to the respective category. Therefore, related to the second level of categories, we propose to categorize incidents using the first approach.

# 5 *Evaluation*

In this chapter, we show the real impact of our implementation, the impact of the proposed method to categorize the incidents, for the first and second level as demonstrated in the previous chapter. This impact is analysed comparing the manual and the automatic categorization. It is present here the difference between both approaches. This analysis includes quantification of which incidents are correctly and incorrectly categorized, how many incidents are recategorized and how much time is spent with this recategorization. With this analysis, it is possible to conclude the impact of adopting automatic categorization, specifically, the proposed method.

This chapter is structured in the following sections:

- **Section 5.1**: In this section, we describe the quantity of incidents recategorized for both categorization levels.

- **Section 5.2:** This section presents, for both categorization levels, the difference of accuracy between the automatic and manual categorization.

- **Section 5.3**: In this section, we describe the time spent in the recategorization process, and how much time could be saved, by replacing the manual by the automatic categorization process.

To understand the impact of the automatic categorization in the IM process, specifically in the IC sub-process, it is necessary to know the quantity of recategorizations that are done ordinarily per incident in the manual categorization. Moreover, if we know how much time is spent from the first recategorization to the last, which means to know the time that an incident is in the queue wrongly categorized, it is possible to know how much time is wasted and in turn how much it is possible to save. By knowing these metrics, we know how the process is improved and the real impact of our proposal.

## *5.1 Recategorization Analysis*

In order to estimate the recategorization times and to obtain the accuracy of the manual categorization, it is necessary to analyse the number of recategorized incidents. To calculate this metric, we will use the same dataset used in the 1$^{st}$ level of categorization and for each incident we will get the number of updates for category. If an incident ticket has zero as update number category, it was assigned to the correct category. If there are one or more updates, the incident was recategorized one or more times. To compare the manual and automatic categorization we will present the number of incident tickets correctly categorized, with an update assigned number zero, and the number of incorrectly categorized, with one or more update number assigned.

Regarding the first level of categories, there are incidents with 207 recategorizations. We perform the same analysis for the subcategories, and one more time we found an update number of 207. In Table 5.1 is it possible observe the distribution of recategorizations per categories and subcategories.

*Table 5.1: Number of recategorizations per categories vs subcategories in the same dataset*

| Number of recategorizations | Occurrences (category) | Occurrences (subcategory) |
|---|---|---|
| **[0-1[** | 32,007 | 31,483 |
| **[1-36]** | 17,512 | 18,011 |
| **[37-73]** | 343 | 352 |
| **[74-110]** | 93 | 101 |
| **[148-183]** | 20 | 19 |
| **[184-207]** | 10 | 10 |
| **Total** | **17,993** | **18,517** |

Analysing the results in Table 5.1, it is possible to verify that most incidents are recategorized between one and 36 times. However, at the same time there are incidents with a very high number of recategorization, which has a negative impact, due to the spent time from the first to the last recategorization.

The results for both cases are very similar, however the number of recategorizations is higher for the subcategories, which is expected, since when an incident changes its category, automatically changes its subcategory. When the category is finally correct, the incident may still change the assigned subcategory, leading to a higher number of recategorizations. In the next sub-section, we use this analysis to verify the amount of time really spent in the recategorization process and how much time of it could be saved if we replaced the manual categorization.

## 5.2 First and Second Level Analysis

As presented previously, the best method achieves results of 80% accuracy, which means 80% of incident tickets are assigned correctly and 20% incorrectly. With the manual categorization, we observed for the dataset with 50,000 incidents that 17,993 had an update number greater than zero, meaning an incorrect category assignment. These values represent an accuracy of 64% in manual categorization.

The same process was done for the second level categorization. We analysed the same dataset, but this time we focused on the update number of subcategories. We observed that 18,517 incidents were incorrectly categorized, which means an accuracy of 63%, less 1% than for the first level. Table 5.2 presents the accuracy results for both categorization types.

*Table 5.2: Accuracy of manual vs automatic categorization*

| Categorization Type | 1st level | 2nd level |
|---|---|---|
| **Manual** | 64 % | 63 % |
| **Automatic** | 80 % | 79 % |

Related to the first level categorization, the automatic categorization presents more 16p.p. of accuracy than the manual. For the second level, the highest accuracy was 79%, also more 16p.p. than the accuracy obtained with the manual categorization.

Analysing these results is possible verify the increasing of incidents correctly categorized, and consequently confirm the positive impact of the automatic categorization.

However, in addition to the analysis of the accuracy of manual and automatic categorization, we also analysed the time that is possible to reduce with the automatic IC. For that, we analyse the same dataset used before for the categories and subcategories. We analysed the time between the first and last update time of category assignment of each incident, verifying that the time wasted for the 17,993 recategorized incidents between the first and last categorization was approximately 756,308 hours, which gives an average of 42 hours per incident until finally assigning the correct category. Comparing to the automatic categorization, where 9,992 incidents were incorrectly categorized, and assuming 42 hours per incident, the waste is 419,664 hours if we consider that the classifier waste the same time to assigning the correct category. With the automatic categorization it is possible to reduce 336,644 hours.

Related to the subcategories, we analysed the data one more time, but in this case the first and last update time of assigning a subcategory. In this case, the time spent between the two recategorizations is 833,265 for 18,517 incidents, which means an average of 45 hours per incident. In the automatic case, there are 10,492 incidents incorrectly categorized which corresponds to 472,140 hours. By applying the automatic subcategorization, makes it possible to save 361,125 hours. Table 5.3 presents the average time spent per incident, the number of incidents wrongly categorized, the total time spent with these incidents and the time that can be saved.

*Table 5.3: Time used in 1st vs 2nd Level*

| | **1st Level** | | **2nd Level** | |
|---|---|---|---|---|
| | **Manual** | **Automatic** | **Manual** | **Automatic** |
| **AVG hours per incident** | 42 | | 45 | |
| **Number of incidents incorrectly categorized** | 17,993 | 9,992 | 18,517 | 10,492 |
| **Total hours** | 756,308 | 419,664 | 833,265 | 472,140 |
| **Total save time** | 336,644 (44.5%) | | 361,125 (43.3%) | |

It is relevant to clarify that for the automatic categorization, we are assuming that for each incident 42 hours are spent for the first level and 45 hours for the second level, however this is not realistic. Due to not having data of the incidents automatically recategorized, it is not feasible get the average time spent in the recategorization process with the automatic categorization. So, we assume the worst case: it means the 42 hours to the categories and 45 hours to the subcategories in the recategorization process just like in the manual categorization. Even with this scenario, where the classifier assigns a wrong category and then assigns to the incident the next most likely category and so on, the amount of time saved is significant.

Related to the proposed method we should also refer that resources such as the ITIL agents and the resolution groups are always wasting time and productivity when an incident is wrongly assigned to a category, because they must check if the incident ticket is correctly categorized. However, if the incident is wrongly categorized, with the automatic categorization is possible assign the next more suitable category, without resorting to an ITIL agent to do this categorization. As we described above, it is possible to assign correctly more 16p.p. of incidents than in the manual categorization. So, automatic IC always save more resources, and consequently costs.

# 6 *Conclusion*

IC is an essential process in IM Process. With the increasing of data in companies, there is the need to develop methods which automate categorization. With this, the resource reduction is acquired, which brings less costs and the possibility to classify the incidents more accurately.

Merging TM and NLP techniques with ML algorithms turns able to achieve what we had proposed previously. In this work, we used real datasets to understand and apply all techniques and algorithms and to conclude which have the best impact on the data. NL techniques and algorithms of Supervised Learning were used to compare all the results and conclude which presents the higher performance. The obtained metrics for each algorithm made possible the comparation between the several TM techniques and consequently between the algorithms used. The assessment of the metrics allows us to propose the best method for IC.

We introduced a method that excludes the full description as attribute for the categorization, but maintains the short description, the location, and severity. The method uses the SVM algorithm and the alphabetic tokenizer, the transformation of tokens into lower case, and the IDF technique. Then, we tested this method for the first and second level of categorization, in order to demonstrate the applicability of the proposed method and show how our method improves the IC step. Both applications present better results than the manual categorization, in terms of reducing the number of categorization errors and of reducing the time spent from the first categorization to the last, i.e. the recategorization time.

We demonstrate that our method improves the IC, since there are more 16p.p. of incidents correctly categorized. Regarding the times, it is possible to reduce at least 44.5% of the time wasted with the recategorization process.

## 6.1 Research Questions Analysis

In this work, the main objective is to propose the best method to categorize automatically and correctly the incidents, therefore there are three main questions that we had proposed to answer:

RQ1: Can we use supervised learning algorithms and TM techniques to categorize automatically incident tickets?

RQ2: Can we find a method that categorizes correctly the most of incident tickets?

RQ3: The proposed method improves the IC process?

Along the development and demonstration chapter, after defining which are the attributes that represent the best classifier performance was used several TM techniques and algorithms to conclude if it was possible categorize incidents automatically. At the same time, was tested which is the best combo to obtain the best method to categorize the incident tickets. From the four algorithms, the best performance was with the use of SVM and it was obtained a performance much better than with the others, proving be useful and capable to assign the correct category to the incident. The TM techniques were also a critical step used, improving the results in the application of all algorithms, however all algorithms have their specific techniques with which is obtained better results. The techniques used with SVM and with are obtained the best results were Alphabetic Tokenizer, lower-case tokens and IDF. As seen in chapter four, our method presented values of 80% to accuracy, precision, recall and F-Measure. The proposed method achieved the defined goal, categorizing correctly the most of incident tickets.

Regarding with the automatically and correct categorization, there are metrics that allowed us to compare the different categorization types and to understand the impact that the automated categorization can brings, as described in the chapter five.

With this we have the information needed to answer to our research questions:

**RQ1:** As shown in the Development and Demonstration chapter it is established the use of four supervised learning algorithms together with several TM techniques to categorize automatically the incidents, without any human interaction.

**RQ2:** The best combo among the algorithms and the techniques is composed by the SVM algorithm and the alphabetic tokenizer, lower-case tokens and IDF. This combo constitutes the best method to categorize the incident tickets, proved categorize correctly 80% of all incidents.

**RQ3:** Regarding the method had improved the IC process is shown in the Evaluation chapter the difference between the manual and the automatic categorization. The two metrics were the number of categorization errors and the time spent since the first recategorization to the last, when is finally found the correct category. As seen, for the automatic categorization, the errors are reduced in 16p.p. for both categorization levels. Related to the time, with the automatic is reduced 44,5% for the first level and 43% for the second level.

## *6.2 Limitations*

In this research it was inevitable to find some limitations with the proposed method. With the application of the method it was impossible categorize correctly all incident tickets, however we achieved our objective, since the proposed method categorize more incident tickets correctly, than the manual categorization does.

A limitation identified was to obtain a better classifier, using other algorithms or changing the current parameters of the classifier to categorize more incidents correctly. When we concluded that SVM was the algorithm with better performance, we could focus on improving the algorithm results, using other parameter values of SVM, and increasing the general performance.

We also know that we have more data that was available to use, since there are a higher number of incidents in historic than the number that we used, however due to capacity issues, we used less data than the available data. This had produced worst results than if we had used the total data. However, if we consider all available data, the training time will be much higher, which is not good when we consider integrate it in an ITS.

## 6.3 Future Work

Our research proves that we are able to categorize automatically the incidents, and more accurately than with the manual categorization. So, with our results we pretend extend the IC automation to the whole activity of incident classification and initial support included in the IM process. This process includes the assignment of a priority and an urgency to incidents, that currently is also done manually by an ITIL agent. Almost all activities of IM process could be automated, therefore one other activity that we pretend automate is the resolution and recovery of incidents. This process is based on finding and suggesting automatically a positive and suitable resolution to an incoming incident, also considering the incident descriptions and most likely the respective location and severity. The principle of assign a resolution to an incident ticket is the same of assign a category. Both require the text pre-processing and consequently, instead of assigning a category is assigned an incident resolution.

In our approach we only involve a first and a second level of categories, but we intended to add a third level. Currently in the company case is used a third level that helps to specify more deeply the incident type and it is our goal introduce this third level automatically. More deeply we categorize incident tickets, more accurately is assigned the right resolution group.

Even knowing the advantages related with the time and the categorization errors, we want to carry out the integration and assess the impact of IC automation by performing interviews to the all IT teams responsible by IM process. With the automatic IC process, all team members related with IM process are impacted.

# *Bibliography*

Abbott, M. L., & Fisher, M. T. (2009). The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise. In *Director* (p. 592).

Agarwal, S., Sindhgatta, R., & Sengupta, B. (2017). SmartDispatch: enabling efficient ticket dispatch in an IT service environment. In *KDD '17 Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1393–1401). https://doi.org/10.1145/2339530.2339744

Aggarwal, C. C., & Zhai, C. X. (2013). *Mining text data*. *Mining Text Data* (Vol. 9781461432). https://doi.org/10.1007/978-1-4614-3223-4

Altintas, M., & Tantug, A. C. (2014). Machine learning based volume diagnosis. *Proceedings of the International Conference on Artificial Intelligence and Computer Science (AICS 2014)*, (September), 195–207.

Arlot, S., & Celisse, A. (2009). A survey of cross-validation procedures for model selection, *4*, 40–79. https://doi.org/10.1214/09-SS054

Braun, A. C., Weidner, U., & Hinz, S. (2010). Support Vector Machines for Vegetation Classification – A Revision. *Photogrammetrie - Fernerkundung - Geoinformation*, *2010*(4), 273–281. https://doi.org/10.1127/1432-8364/2010/0055

Cannon, D., & Wheeldon, D. (2007). *ITIL Service Operation*. *Itil*. https://doi.org/10.1007/978-0-387-77393-3

Cardoso-Cachopo, A., & Oliveira, A. L. (2003). An Empirical Comparison of Text Categorization Methods. In *Proceedings of SPIRE-03, 10th International Symposium on String Processing and Information Retrieval* (pp. 183–196). https://doi.org/10.1007/b14038

Chih-Wei Hsu, Chih-Chung Chang, and C.-J. L. (2008). A Practical Guide to Support Vector Classification. *BJU International*, *101*(1), 1396–400. https://doi.org/10.1177/02632760022050997

Dalal, M. K., & Zaveri, M. A. (2011). Automatic Text Classification: A Technical Review. *International Journal of Computer Applications*, *28*(2), 37–40. https://doi.org/10.5120/3358-4633

Dias Freire De Mello, T., & Lopes, E. C. (2015). Utilizando Racioc??nio Baseado em Casos em uma Metodologia de Apoio ?? Decis??o para Controle de Resolu????es de Incidentes em TI. *2015 10th Iberian Conference on Information Systems and Technologies, CISTI 2015*. https://doi.org/10.1109/CISTI.2015.7170448

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *7th International Conference on Information and Knowledge Management (CIKM'98)*. https://doi.org/10.1145/288627.288651

Duneja, A., & Puyalnithi, T. (2017). Enhancing Classification Accuracy of K-Nearest Neighbours Algorithm Using Gain Ratio. *International Research Journal of Engineering and Technology*, *4*(9), 1385–1388.

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

Gupta, R., Prasad, K. H., Luan, L., Rosu, D., & Ward, C. (2009). Multi-dimensional knowledge integration for efficient incident management in a services cloud. *SCC 2009 - 2009 IEEE International Conference on Services Computing*, 57–64. https://doi.org/10.1109/SCC.2009.48

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

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS Transactions on Computers*, *4*(8), 966–974.

Joachims, T. (1998a). Text categorization with support vector machines: Learning with many relevant features. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1398*, 137–142. https://doi.org/10.1007/s13928716

Joachims, T. (1998b). Text categorization with Support Vector Machines: Learning with many relevant features BT - Machine Learning: ECML-98. In C. Nédellec & C. Rouveirol (Eds.) (pp. 137–142). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, *31*, 249–268. https://doi.org/10.1115/1.1559160

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. https://doi.org/10.3115/v1/P14-5010

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

Mccallum, A., & Nigam, K. (1997). A Comparison of Event Models for Naive Bayes Text Classification.

Mohit, B. (2014). Named Entity Recognition, (July), 221–245. https://doi.org/10.1007/978-3-642-45358-8_7

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

Rokach, L., & Maimon, O. (2008). *Data mining with decision trees : theroy and applications*. *Series in machine perception and artificial intelligence*. https://doi.org/10.1142/9097

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

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47. https://doi.org/10.1145/505282.505283

Shao, Q., Chen, Y., Tao, S., Yan, X., & Anerousis, N. (2008). Efficient Ticket Routing by Resolution Sequence Mining, 605–613.

Singh, S., & Gupta, P. (2014). Comparative study ID3, cart and C4 . 5 Decision tree algorithm: a survey. *International Journal of Advanced Information Science and*

*Technology (IJAIST)*. https://doi.org/10.15693/ijaist/2014.v3i7.47-52

Son, G., Hazlewood, V., & Peterson, G. D. (2014). On Automating XSEDE User Ticket Classification.

Song, Y., Huang, J., Zhou, D., Zha, H., & Giles, C. L. (2007). IKNN: Informative K-Nearest Neighbor Pattern Classification. *Proc. of the European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 248–264. https://doi.org/http://dx.doi.org/10.1007/978-3-540-74976-9_25

Song, Y., Sailer, A., & Shaikh, H. (2009). Problem classification method to enhance the ITIL incident and problem. *2009 IFIP/IEEE International Symposium on Integrated Network Management, IM 2009*, 295–298. https://doi.org/10.1109/INM.2009.5188825

Srividhya, V., & Anitha, R. (2010). Evaluating preprocessing techniques in text categorization. *International Journal of Computer Science and Application*, (2010), 49–51. Retrieved from http://www.sinhgad.edu/IJCSA-2012/pdfpapers/1_11.pdf

Steinberg, R. A. (2013). *Measuring Itsm: Measuring, Reporting, and Modeling the It Service Management Metrics That Matter Most to It Senior Executives*. Trafford Publishing. Retrieved from https://books.google.co.uk/books?id=MkCJAgAAQBAJ

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. *Springer* (Vol. 8). https://doi.org/10.1109/TNN.1997.641482

Vijayarani, S., Ilamathi, J., & Nithya, M. (2015). Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99* (pp. 42–49). https://doi.org/10.1145/312624.312647

Zhou, W., Xue, W., Wang, Q., & Shwartz, L. (2017). STAR : A System for Ticket Analysis and Resolution. In *KDD 2017 Applied Data Science Paper* (pp. 2181–2190). https://doi.org/10.1145/3097983.3098190