Tampere University

Vishal Gaur

# *INFORMATION EXTRACTION FROM PATENT OFFICE ACTIONS USING NLP TECHNIQUES*

Faculty of Information Technology
and Communication Sciences
Jyrki Nummenmma
Elizaveta Zimina
October 2020

# ABSTRACT

Vishal Gaur: Information Extraction from Patent Office Actions using NLP Techniques
Master of Science Thesis
Tampere University
Master's Degree Programme in Data Engineering and Machine Learning
October 2020

---

Patent Processing consists of various important tasks such as filling application, searching for prior art, responding to office actions and many more. Researchers mostly focused on working with prior arts and patent application but neglect analysing office actions. This thesis is devoted to classifying and extracting relevant information from office actions. Since office actions have not been in mainstream research, a new dataset of 2300 office actions was created for this research. Office actions were processed using the Tesseract OCR engine to create a raw dataset. The raw dataset was cleaned, pre-processed, and manually labelled into two classes, relevant (1) and irrelevant (0). The cleaned and processed dataset had almost 30000 irrelevant paragraphs and around 6000 relevant paragraphs. The text classification approach was experimented on the processed dataset to find the relevant and irrelevant information.

Since the dataset was created for this research, there were no baseline results. To establish a baseline, the Gaussian Naïve Bayes classifier was used. Random Forest, AdaBoost and XGBoost were chosen as models for evaluation during the experiments. In text classification, researchers apply various traditional algorithms such as SVM, Decision trees along with state-of-the-art methods such as CNN, DNN etc. Random Forest algorithm used in this research is based on decision trees. AdaBoost and XGBoost utilize a gradient boosting technique, that is the dataset is trained using multiple iterations of classifiers. Each iteration of a classifier adjusts the feature weights to improve the accuracy score. The classifier with the best accuracy is opted by the algorithm as it has the most optimized feature weights. The word embedding technique also played an important role in text classification. Therefore, in this thesis, Count Vectorizer and TF-IDF Vectorizer were used to investigate the results on the dataset. Count Vectorizer focuses only on the total frequency of the vocabulary of the dataset. The TF-IDF Vectorizer utilizes frequency of the vocabulary and combines it with the inverse frequency of the words in the document.

The experiments were carried out using the Spyder framework. The baseline results for the research were 93.17% and 87.78% for Count Vectorizer and TF-IDF respectively. The experiments showed XGBoost as the best classifier with the highest accuracy of 96.50% using the TF-IDF Vectorizer. The lowest accuracy achieved was 95.69% for AdaBoost using the TF-IDF Vectorizer. The F1-score for XGBoost was the highest with a value of 0.866 followed by Random Forest with 0.850 and AdaBoost with 0.842. Although the accuracy score for the AdaBoost model was higher using Count Vectorizer, the F1-scores for all the models were better for the TF-IDF Vectorizer.

By analysing the results from the classifiers, it was clear that the paragraphs with contradictory keywords were more prone to misclassification. However, the unevenness of the dataset could also be responsible as the number of irrelevant paragraphs greatly overshadows the relevant ones.

Keywords: Patents, Office Actions, Text classification, Information Extraction

# PREFACE

I want to express my deepest gratitude to my supervisors, Prof. Jyrki Nummenmaa, Researcher Elizaveta Zimina from Tampere University and Mr. Ari Aarnio from Nokia. They guided and supported me throughout the thesis process. I am grateful for their valued insights on the topic without which this master's thesis would not be possible to complete. I would like to appreciate their kindness and patience since the thesis took a lot longer than expected.

I would like to thank my fellow teammates who helped me during difficulties. They helped me to overcome various problems. I would also like to thank my university for providing a lot of resources which proved vital during the process of writing of thesis.

Tampere, Finland, 21st October 2020

Vishal Gaur

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| APAC | Asia Pacific |
| CNN | Convolutional Neural Network |
| DL | Deep Learning |
| EPO | European Patent Office |
| FN | False Negatives |
| FP | False Positives |
| IE | Information Extraction |
| IP | Intellectual Property |
| MLP | Multi-Layer Perceptron |
| NB | Naïve Bayes |
| NLP | Natural Language Processing |
| OA | Office Action |
| OAPI | African Intellectual Property Organization |
| PE | Patent Engineers |
| PN | Patent Numbers |
| R&D | Research and Development |
| SOTA | State-of-the-art |
| SVM | Support Vector Machine |
| TN | True Negatives |
| TP | True Positives |
| USPTO | United States Patent and Trademark Office |
| WIPO | World Intellectual Property Organization |
| XML | Extensible Markup Language |

# 1. INTRODUCTION

A patent is a document that provides innovators with exclusive rights to their invention which can be a process or a product that describes a novel way of accomplishing something or offers a modern technical solution [1]. To obtain a granted patent, an innovator must file an application containing the title of the invention, its description and how it is different from the products/processes that are already patented. Along with this, there is a specific set of claims that the innovator believes are novel and should help his invention to be granted a patent. This application is then examined by various patent offices such as EPO, USPTO, OAPI, etc. after which the examiner responds to the applicant with an office action (OAs).

Office actions are written documents from examiners, to which an applicant is obliged to respond in order to proceed with the patent granting process [2]. An OA contains a brief about the flaws in the application in accordance with the patent laws, the rejected and allowed claims from the application and other patent regulations. The responses of OAs must clearly acknowledge all rejections and objections mentioned by the examiner. There can be various types of OAs, such as restriction requirements, first OA, non-final OA and final OA [2].

Information extraction (IE) is a sub-discipline of artificial intelligence which oversees the problem of allowing machines to read unstructured data. For example, one company wants to track the sentiments about its various products. Another one wants to find out various technological advancements from the news feed it received. A government agency wants to gather information on a natural calamity and provide a summary of that information to the emergency services [3] and help create a response team. All these examples may have different type, size, and sources of information, but one thing that brings all these problems together is that they all want to read and process a lot of information, but it is humanly impossible to process all that data in a given time period. And as for machines, they cannot process these queries as it is in an unstructured format. This is where IE plays a vital role. Patents are among the documents which contain a wide range of information. Information extraction from patents has been in limelight for a long time. IE applied to patents can be categorized mainly into two categories:

1. Analysis of bibliographic information

2. Processing of patent content, such as abstract, description, title etc.

Various methods and processes have been developed and applied to patent documents focusing on different aspects of patent processes [4][5]. Most of the processes and methodology focus on prior art and patent application, but there is a hidden aspect of the process that has been overlooked. In order to receive a granted patent, one needs not only to find appropriate prior art and claims but should also be able to properly analyse OAs from examiners. If an application is top notch, but the applicant cannot respond properly to the OAs, it will be rejected automatically.

The aim of this thesis is to fill in that gap of analysing OAs and classify the information that is relevant for patent application and analysis. The project is aimed to develop a system that can be used to classify the relevant and non-relevant information from the received OAs. The system was developed on Nokia premises envisioned to extract relevant information from the OAs and easing the patent processes.

The first research question is how to gather data to train the system. Even though patent analysis has been in limelight for quite a while, the OAs section has not received much attention, so there are not a lot of datasets that can be used to design the system. Thus, for this research, a dataset was created manually by gathering 2300 OAs and annotating them into required classes.

The second research question is which model and approach will give the most effective results of classifying the information retrieved from the OAs. The text from the OAs were classified into relevant and irrelevant classes using text classification algorithms. Different traditional models were researched during the research to obtain the best results.

The thesis has the following structure. The introduction is followed by the previous works done in patent analysis, IE and text classification. Section 3 describes the research methodology including the data gathering phase, research of documents, search for various modules in Python, problems faced and their solutions. Along with this there will be information about the NLP techniques and text classification models used in the system. Section 4 comprises the classification results, evaluation, discussion about the results and the future work. Finally, the 5$^{th}$ section draws the conclusion of the thesis with a brief explanation about the discussion.

# 2. PREVIOUS WORK

This chapter provides detailed background information about text classification algorithms. The first section provides the fundamental information about patents, their related documents and patent processes. In section two, concepts related to NLP are described including word embedding models and classification algorithms. The detailed information about the information extraction in patent documents is explained in the third section. In section four, common evaluation metrics for classification models are introduced.

## 2.1 Patent Processes and Office Actions

“The computer is incredibly fast, accurate, and stupid. Man is unbelievably slow, inaccurate, and brilliant. The marriage of the two is a challenge and opportunity beyond imagination.” – Stuart G. Walesh, 1989

The quote aligns perfectly with the current era as the alliance of both computer and man is being prioritized in different areas. Starting from historical times, industries have been found to evolve and expand meeting the forthcoming challenges and imminent technological changes [6]. The chase of being the best and gain slight advantage over the competitors has led industries to substantially increase their Research and Development (R&D) spending [7]. The firms continuously need to innovate and evolve to preserve their competitive edge in the market. In the ever so large growing world, the need to innovate has never been a greater necessity. The innovations are created to enhance the quality of human life, but the industries should also be encouraged to continue their path of innovation. The best way to encourage the industrialists to maintain their innovative paths is to monetize their inventions.

In the legal terms, monetizing the innovations implies creating them as Intellectual Properties (IPs). “Intellectual Property refers to the creations of mind, such as inventions; literary and artistic works; designs; and symbols, names and images used in commerce” [1]. IPs are categorized under three major categories:

1. *Copyright*s – Copyrights are mostly related to entertainment industry, that is all the artwork composed by musicians, artists, producers etc. is copyrighted under their name, thus providing them with the rights to protect their creation/products.

2. *Trademarks* – A trademark is a symbol that distinguishes a certain type of products or amenities provided by a specific individual or company.
3. *Patents* – A patent is an exclusive document that provides the inventor with the ownership rights of his/her invention for a certain period.

According to WIPO, “A patent is an exclusive right granted for an invention – a product or process that provides a new way of doing something, or that offers a new technical solution to a problem” [8]. A patent allows the patent owners to protect their invention from duplicity, theft and other illegal activities. This protection is provided for a certain duration of time, usually 20 years, before the invention is made publicly available. A patent not only provides protection to the inventors for their creation but also incentives by recognition of their work, possibility to market the product and so on. These incentives then encourage them to innovate and enhance the quality of human life.

Although a patent provides various forms of protection for a long time, receiving a granted patent takes years of struggle along with numerous documentation steps. The first step is to file a patent application which describes the invention and how is it different from the products that are already patented. An application should contain a specific set of claims that the innovator believes are novel and should help his invention to be granted a patent. Then this application is sent to patent offices such as EPO, OAPI, USPTO etc. where the offices assign an Examiner to look over the filed Patent application. The examiner compares the current application with the previous granted patents, filed patent applications or prior art to find the similarities between them. Once he is done with the application, he will note down his findings in an official document. The document created in response to the filed patent application is known as an Office Action (OA).

In the process of patent application, the applicant will eventually receive an OA [9]. The OAs can be classified into one of the three categories. In one response, the examiner searches for the prior art, finds nothing similar to the claims filed and accepts the application. In a second possible response the examiner finds various similarities between filed application and the prior art which makes the current claims of the application unpatentable. In this case the OA contains the information about the patent numbers which have similar claims and comments about the similarities in both applications. In a third response, the examiner finds the filed application in discord with various patent application requirements which must be corrected before conducting any searches [9]. Mostly, the second type of responses are the OAs which are needed to be examined properly by Patent Engineers (PE). A PE is a person who has in-depth knowledge of a company’s technology and also substantial information about the patent laws. They help in crafting

a proper patent strategy which includes drafting application, performing prior art searches, responding to OAs etc [9].

## 2.2 Natural Language Processing

### 2.2.1 Introduction to Text Classification

Researchers have fixated their attention on text classification for a long time. Either new systems are being developed or the existing algorithms are improvised to create models yielding higher computational results [10][11][12]. There are numerous pre-trained models like BERT [53], XLNet [52], RoBERTa [54] etc. which are available for language processing tasks and they are still being worked upon to get better results. Most recent update came out about Facebook's (FB) KILT [13] and Google's PRADO. KILT by FB provides answers to the multi-domain language processing problems [13] whereas PRADO which was launched almost a year ago was improved to be able to achieve BERT level accuracy and making the model 4 times smaller [14]. The purpose of text classification is identifying and categorizing the portion of text to different labels or classes precisely. It can be further categorized into single-label and multiple-label classification. Table 2.1, 2.2 and 2.3 show a few examples of text classification. Table 2.1 represents single-label classification. Table 2.2 shows an example of single-label multi-class. Table 2.3 demonstrates the multi-label multi-class classification.

| Examples | Classification |
|---|---|
| It was the best of times. | 1 |
| It was the worst of times | 0 |
| It was amazing age of industries | 1 |

***Table 2.1*** Single-label classification

| Examples | Classification |
|---|---|
| It was the best of times. | 1 |
| It was the worst of times | 2 |
| It was amazing age of industries | 1 |
| It was foolish act of bravery. | 0 |

***Table 2.2*** Single-label multi-class classification

| Examples | Classification |
|---|---|
| It was the best of times. | 0, 1, 2 |
| It was the worst of times | 1 |
| It was amazing age of industries | 0, 2 |
| It was foolish act of bravery. | 0 |

***Table 2.3*** Multi-label multi-class classification

Single-label classification with just two target class is also referred to as Binary classification. Since there are only two target classes, the classification result is either of them.

Single-label multi-class classification is slightly different in a way that there are more than two target classes, and the classification result can be either of those classes. Multi-label multi-class classification can be explained as a combination of the previous two, that is there are multiple target classes, and the classification result can be either or all of those classes. Text classification is being utilized in various domains such as industries, finance, medicines [15], Patents [16][17][18] etc. Different domains require researchers to carve text classification techniques as per their requirement such as news classification, categorization of document parts, detection of prior art from patent applications [19], extracting user information from patents [20] and so on. News classification can be taken as one of the most common classification tasks. The news text is related to different topics that are needed to be classified accordingly. There are several topics, for instance politics, sports, economy, business, technology etc. These topics can be thought of as target classes. As a result, we have multiple classes, and the news text can be classified into any one of these categories which makes this problem as a single-label multi-class classification. Another similar example in relation to patents can be classifying patents into their technological classes. Patent applications are classified into different technologies such as radio, telecommunication, software, machine learning etc. Unlike the news classification example, patents can be classified as multiple classes which makes it a multi-label multi-class classification problem. As text classification has been researched upon for a long time, there are numerous examples for binary, single-label, and multi-label classification. It depends on how researchers look at the problem.

### 2.2.2 Embedding Models

Machines require numbers to perform calculations. In order to process textual information, these texts are to be converted into some form of numerical structure. Word embedding is a type of algorithm that maps textual information to vectors in numeric form. Every word is represented by a vector using the word embedding techniques. Figure 2.1 represents various word embedding techniques used in state-of-the-art methods.

Statistical word embedding models like Bag of Words (BoW), TF-IDF etc. are traditional word embedding techniques that have been handy for language processing tasks where the vocabulary is limited to a certain extent. These models put an emphasis on a vector of feature weights for solving the problem. The whole document is embodied as a vector of features where features are words in the document. On the other hand, the language model-based embedding techniques provide different advantages over statistical methods and focus on more diverse textual information, that is they also consider semantic information which is useful in various language processing tasks. Language model-

based embedding models usually work with a large text corpus to reduce the dimensionality of feature vectors. If the dataset is not large enough, the statistical methods provide better results. The feature types also matter while selecting the embedding algorithms. There are different feature types such as n-grams, taxonomies of features, domain specific features [21] etc. for different tasks. Since the dataset in my thesis does not contain a huge amount of information, BoW and TF-IDF as statistical embedding techniques in combination with N-grams were used.

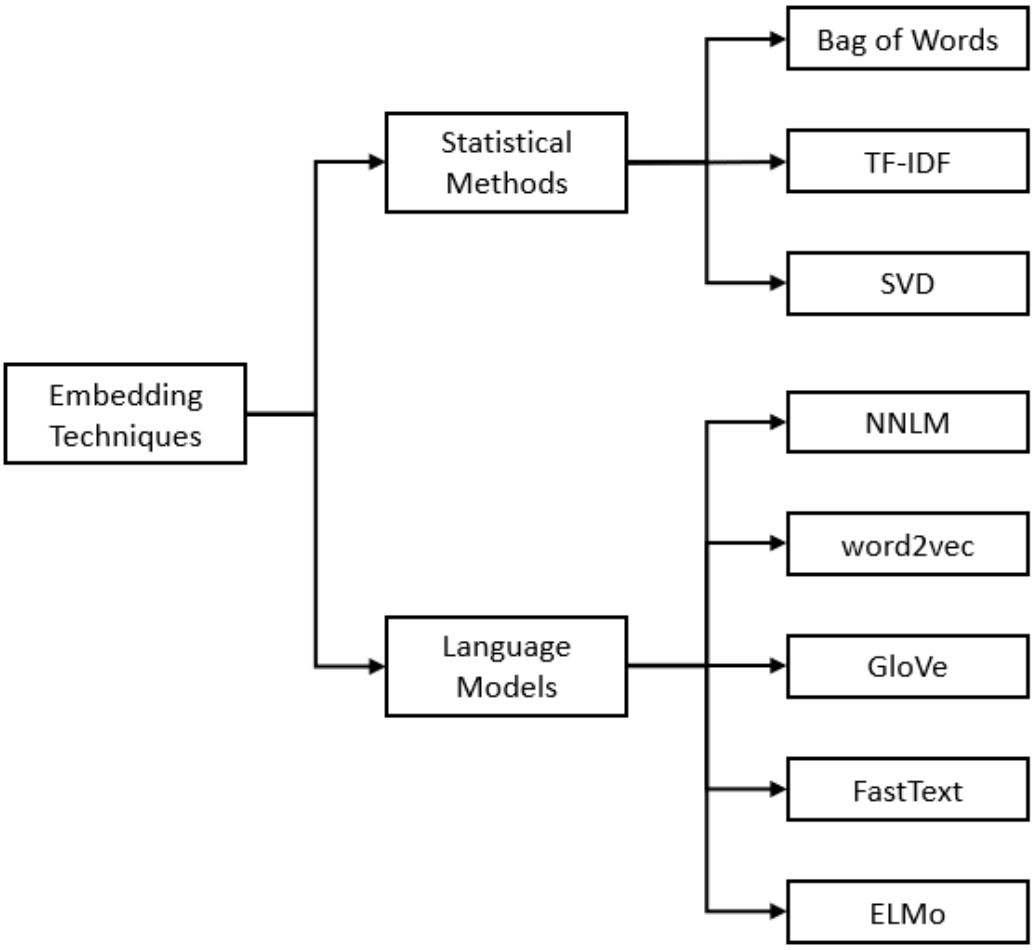


***Figure 2.1*** Word Embedding Techniques

**Bag of Words**

BoW is one of the most used and commonly known feature extraction procedures while working with sentences and documents [22][23][24]. This approach depends on the vocabulary size of the input documents. BoW is a statistical method which depends on the term frequency (TF) of all the words as feature to accomplish the task. Table 2.4 demonstrates an example of how BoW created the features from the input provided.

| Statement | Features |
|---|---|
| It was the best of times | It, was, the, best, times, of |
| It was the worst of times | It, was, the, worst, of, times |
| It was amazing age of industries | It, was, amazing, age, of, Industries |
| It was foolish act of bravery. | It, was, foolish, act, of, bravery |

***Table 2.4*** BoW example

BoW is a simple model that considers words as features and its frequency as weight. For example, vocabulary created from Table 2.4 will be, "*It, was, the, best, of, times, worst, age, industries, foolish, act, amazing, bravery*". The feature matrix that will be created from the BoW model for the statement given in Table 2.4 can be seen in Table 2.5.

| Features | Statement 1 | Statement 2 | Statement 3 | Statement 4 |
|---|---|---|---|---|
| It | 1 | 1 | 1 | 1 |
| Was | 1 | 1 | 1 | 1 |
| The | 1 | 1 | 0 | 0 |
| Best | 1 | 0 | 0 | 0 |
| Of | 1 | 1 | 1 | 1 |
| Times | 1 | 1 | 0 | 1 |
| Worst | 0 | 1 | 0 | 0 |
| Age | 0 | 0 | 0 | 0 |
| Industries | 0 | 0 | 1 | 0 |
| Foolish | 0 | 0 | 0 | 1 |
| Act | 0 | 0 | 0 | 1 |
| Amazing | 0 | 0 | 1 | 0 |
| Bravery | 0 | 0 | 0 | 1 |

***Table 2.5*** Feature weights using BoW

The feature weights in Table 2.5 are numbers of occurrences of that feature in the input statements. BoW uses the whole vocabulary from the input data to create feature vectors and assigns weights. Since the number of features is directly proportional to the number of words in the document, this approach increases the dimensionality of the feature vector and is considered a liability when working with larger amounts of data. BoW can be as simple or complex as we like which is why this approach becomes very handy while working with smaller datasets.

**TF-IDF**

TF-IDF is another technique that is used in text classification tasks [25][26]. It also works by weighing the words in the document but is considered better than BoW. The problem with BoW is that the most frequent words will be assigned highest weights. Not always the most frequent words are the best features. Some important features can be overshadowed by insignificant features because of more frequent occurrences. To overcome this setback, the Inverse Document Frequency (IDF) method was proposed by K. Sparck Jones [29] in 1972. IDF in conjunction with TF lessens the effect of implicitly common

words in the feature list. The TF-IDF approach cross references the frequencies of the words in statements with the whole document. The most common words across the documents are also penalized. The TF for the documents is calculated as in BoW but the IDF improves the weighting of features. Thus, the IDF of a rare feature is a higher value and vice versa for a frequent term [27]. Table 2.6 shows the feature weights calculated using TF-IDF approach.

| Features | Weights |
|---|---|
| It | 0 |
| Was | 0 |
| The | 0 |
| Best | 0.4 |
| Of | 0 |
| Times | 0.21 |
| Worst | 0.45 |
| Age | 0.3 |
| Industries | 0.5 |
| Foolish | 0.6 |
| Act | 0.5 |
| Amazing | 0.7 |
| Bravery | 0.7 |

***Table 2.6*** Feature Weights using TF-IDF

In Table 2.5, the feature weights depend on the term frequencies which made keywords such as "*It*", "*Was*", "*The*" good features. TF-IDF changed the feature weights after comparing those frequencies to the whole text corpus and showed a different set of features which can be used to get the tasks completed.

**N-grams**

The embedding models require a list of features to assign different weights. In BoW and TF-IDF models, the features are taken as words or tokens from the input documents. As seen in Tables 2.5 and 2.6, we have multiple zeros. These tables were created just from a few lines of text. When provided with a large enough input, these matrices will have numerous zeros, that is the feature vectors created will contain a lot of zeros. These vectors are called sparse vectors. These sparse vectors require more computational resources and memory making the statistical approaches very difficult to use. To overcome such challenges various pre-processing such as removing punctuation, stop words, stemming etc. is done. To further improve and get more sophisticated results, the tokens are grouped together to provide more meaningful features. While grouping the features, each token is termed as "gram". A sequence of *n* number of tokens is referred to as an N-gram. If a group contains only one token it is known as Unigram, with two tokens it is

known as Bigram and so on [28]. Figure 2.2 demonstrates the n-grams created from an example text.

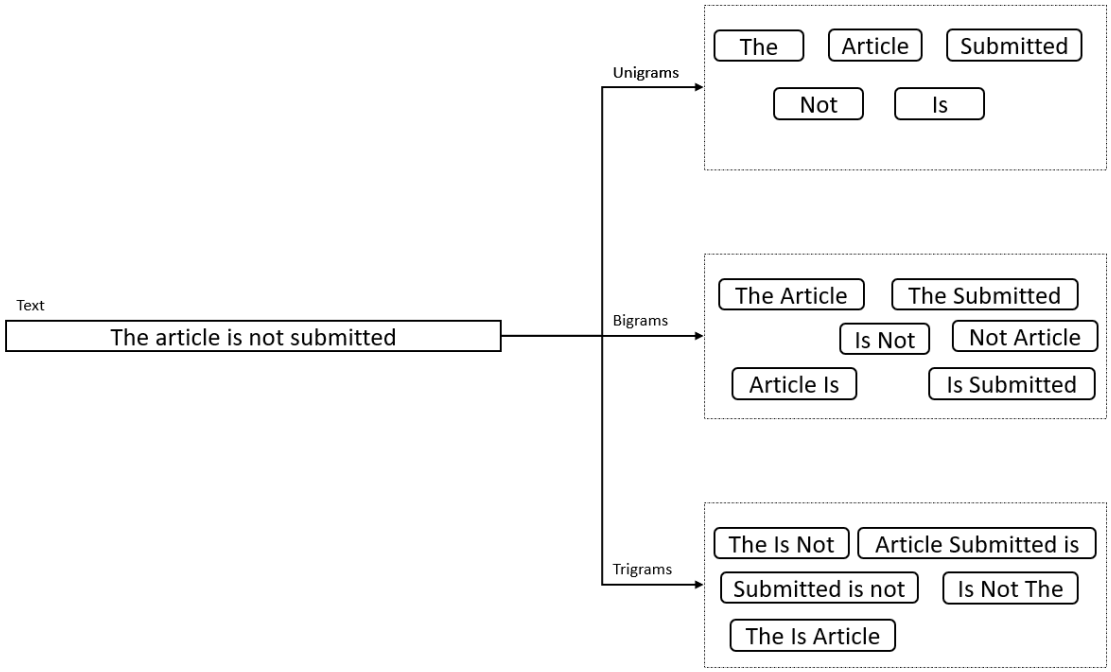


***Figure 2.2*** N-gram example

There are three different n-grams created from a single statement. Bigrams and trigrams consist of a better feature vector than the unigrams. Usually the bigram approach may provide us with better results than the unigrams and trigrams can further enhance the results. The author [24] states that “a bag of bigrams representation is much more powerful than bag-of words, and in many cases proves hard to beat”, that is a bag of n-grams combined with statistical embedding models can yield compelling results. N-grams are also a very common way to work with many text classification problems [25][26].

### 2.2.3 Overview of Algorithms for Text Classification

Various text classification studies have been conducted over the years [21][30]. Kowsari et al. [30] recently conducted a survey for the text classification algorithms along with feature extraction, dimensionality reduction and other relevant stages while performing text classification. The authors analysed various text classification algorithms along with their limitations to the real-world applications.

Generally, text classification approaches can be applied to 4 different scope levels. First is the document level scope which means that the algorithm obtains relevant categories from the whole document. The classification algorithm takes the whole document as an input and provides a prediction for the same. Second is the paragraph level approach

which collects the information from the paragraphs of the text document. The paragraphs are taken in as input by the algorithm, feature engineering is performed on all the paragraphs of the document and the target class is predicted. Thirdly, the sentence level scope focuses on sentences to extract information and lastly, the subsentence level scope draws the information and categories from sentences within the sentences. Different scopes take in input as per their scope levels and perform feature engineering on the same. For example, in the paragraph scope, the bigrams created can consist of multiple sentences which can be used as features by the classification algorithm.

The Rochio classification is amongst the earliest text classification algorithms and was first proposed by J.J. Rochio in 1971 [31]. This classification algorithm also uses the TF-IDF weights for the words that are considered to be informative and builds a feature vector for each class from the training set. Using these vectors, the classes with most similarity index are assigned to the test set.

The statistical machine learning approach for text classification involves two steps. First is to extract the features from the input data. The second step is to use the machine learning algorithms to predict the target outputs. Kowsari et al. [30] have also discussed various machine learning algorithms with two most popular techniques in ensemble learning: Bagging and Boosting. The traditional statistical classification approaches like Logistic Regression (LR) [32], Naïve Bayes [55], Support Vector Machines (SVMs) [33], K-Nearest Neighbours (KNN) [34] etc. are still used extensively in the scientific community. Decision Trees [35] and Random Forests [56] are tree-based classification algorithms which makes them faster and more accurate than the traditional approaches. Boosting algorithms like AdaBoost [57], XGBoost etc. are ensemble techniques which modify the distribution of the training dataset given the classification reports of the trailing classifiers. Along with the traditional approaches, the authors also compared the state-of-the-art deep learning (DL) approaches like Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) etc.

In this thesis, the traditional approach, Naïve Bayes classification was used to create the baseline for this research. This technique has been widely in use for various language processing tasks since early 1950s. This technique utilizes the generative modelling which is the most traditional approaches in text classification. Since it is one of the oldest approaches, Naïve Bayes classifier was used to set the baseline for the research. The other traditional approaches were not used in the research as the baseline results were better than expected. On analysing the baseline results, it was found that the results were exceptionally good because of the structure of OAs.

Random Forest is an ensemble technique which was introduced by T.Kam Ho [56] in 1995. This technique works on the same principal of decision trees classifier by using n number of parallel trees. The best advantage of this technique over other SOTA approaches like deep learning techniques is that it can be trained faster on compromising the prediction rate.

All the traditional approaches have been overcome by the SOTA DL techniques like CNNs, DNNs, LSTMs, GRUs, DBNs etc. The DL approaches have achieved marvellous results across a variety of domains, including a diverse range of NLP applications. All the DL approaches are created by several layers with multiple neurons in each layer which are interconnected to the neurons on the next layer. This structure helps the DL models to learn better than the traditional approaches. A neuron itself consists of a few layers. A representation of a single neuron can be seen in Figure 2.3 where the first layer is the inputs followed by the weight 'w' of the input 'x'. The input and weights are multiplied and added to the bias b. This result is inputted to the activation function 'f' to get final output from a single neuron.

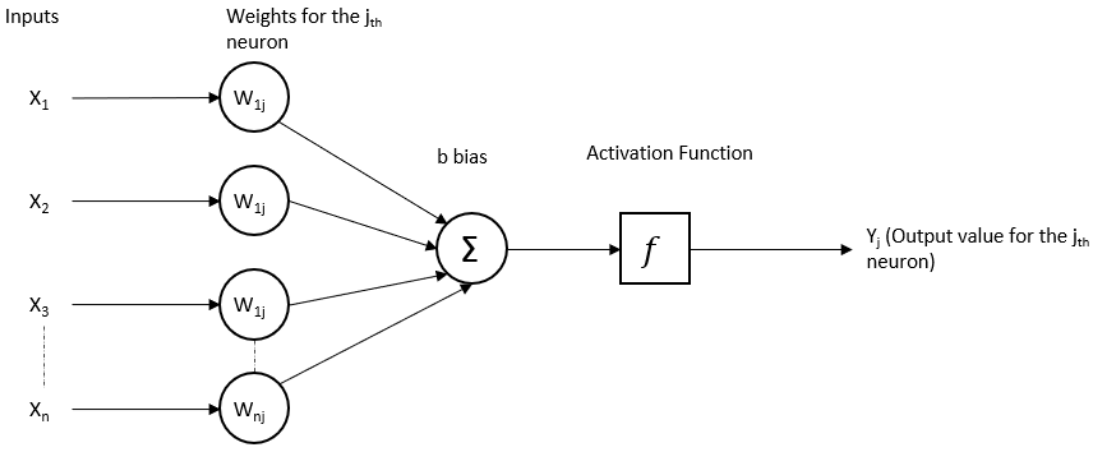


***Figure 2.3*** Structure of a single neuron

Deep neural networks incorporate multiple layers of neurons where every single neuron in each layer is connected to all neurons in the next one. Through this structure the neurons in each layer receive information only from the previous layer. The structure of a DNN can be seen in Figure 2.4 which represents the input layer, output layer and the hidden layers. A CNN is another type of deep learning architecture that is commonly used in various machine learning tasks. The major difference between the DNN and CNN is that the CNN uses different kernels to convolve over the given input and help reduce dimensionality of the data. It was originally developed for image processing but has been useful to provide exceptional results regardless of the domain.

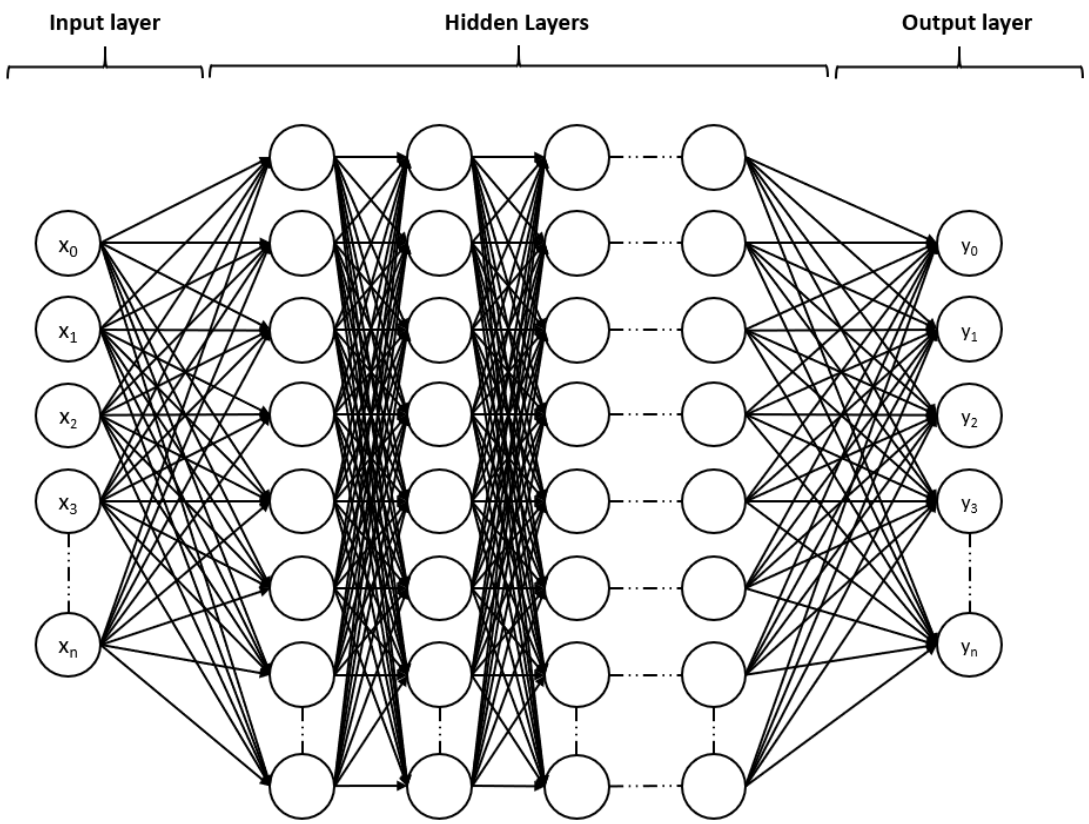


***Figure 2.4*** Structure of a deep neural network

For text classification, the common point between traditional and SOTA DL approaches is the way to provide information, that is, conversion of textual information using embedding models so that it can be encoded for the machine to understand. It is true that DL approaches provide very good results for various tasks across domains, but the biggest drawback is the amount of data. Because the DL approaches have several layers, they need an appropriate amount of data to be trained. Without such a dataset, the usage of DL models leads to misuse of computational resources as they will provide very similar results to the traditional approaches. This is also one of the reasons why the DL approaches were not experimented with in this research.

## 2.3 Information Extraction in Patent Documents

The patent systems consist of various segments such as patent applications, claims, prior arts etc. Different language processing techniques are being applied on all the segments using different approaches, for instance language processing, property-function based approaches, rule-based approaches, neural networks etc. Different state of the art patent analysis methods is being researched upon to improve information extraction form patent documents. Patents contain different types of information. Clams in a patent application need to be novel to grant a patent. If a patent is about a new technology, it

will contain information about the previous work done on that technology, if it is about an astonishing new device, the application will be created after collecting all the information about what were the previous versions of the device etc. The patent team must go through all the literature in order to find all this previous information. Various information extraction techniques [36][37][38] have been developed and researched. Different researchers tried to extract various types of information for patents, for example Helmers et al. [39] focused their attention on searching for the prior art for patent applications by utilizing the NLP and deep learning approaches. Luo et al. [18] studied Named Entity Recognition (NER) in chemical patents using neural networks. Similar works such as user information extraction [40], extracting technical causality [41] are results of utilizing the language processing and machine learning techniques in information extraction from patent documents.

## 2.4 Evaluation Metrics

In research community, experimentation results are compared on the basis of previous results or a baseline. However, such comparison methods don't exist for all the problems. The major issues when evaluating results in NLP occur because of absence of standard data collection protocols [30].

In this thesis, the performance of the algorithms has been evaluated using four evaluation metrics, namely accuracy, F1-Score, precision, and recall. The use of confusion matrix made it easier to analyse and understand the prediction results. The description of these metrics is given below.

### 2.4.1 Accuracy and F1-Score

**Accuracy**

Accuracy in classification tasks is achieved using the accuracy_score metric which compares the set of labels predicted by the algorithm for a test set with its true labels. The mathematical formulation for the accuracy can be seen in the equation below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- TP represents True Positives, that is the classifier predicted the class as 1 when the true class is 1
- TN represents True Negatives, that is the classifier predicted the class as 0 when the true class is indeed 0.

- FP represents False Positives which means that the classifier predicted the class as 1 when the actual class is 0
- FN represents False Negatives which indicates that the classifier predicted the class as 0 when the actual class in 1.

**F1-Score**

Another common evaluation for classification tasks is using the precision and recall metrics. The F1-Score module from Scikit library allows us to get the F1-Score of the classifier. The mathematical equations for precision and recall can be seen below.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + TN}$$

Precision is the value that is received from the positive predictions of the algorithm. Recall on the other hand measures the ability of classifier to predict positive samples. Recall is also termed as sensitivity. Higher the recall value becomes, lesser is the chance of classifier missing any positive values.

The resultant of harmonic mean between the precision and recall is the F1-Score. The equation below calculates the F1-Score.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

It is the weighted average of the precision and recall values. If the F1-Score equals to 1, it is considered to be the best and 0 is considered the worst. The F1-Score in a classification task depends on the average parameters.

## 2.4.2 Confusion Matrix

Confusion matrix is method which allows researchers to visually analyse the classification results. For a binary classification (*in our case*) there are only two possible outcomes: True (Positives) and False (Negatives). Table 2.7 demonstrates a confusion matrix for a binary classification problem.

| | | **Predicted Class** | |
|---|---|---|---|
| | | Positive | Negative |
| **Actual Class** | Positive | True Positive (TP) | False Negative (FN) |
| | Negative | False Positive (FP) | True Negative (TN) |

***Table 2.7*** Confusion Matrix

If the FP and FN values from the matrices are as low as possible, then the classifier is thought to be trained well enough. During the experiments, if we want to figure out if the parameters are good enough for the classifiers, we can look at the confusion matrix. If the values for FN and FP are reducing that means the parameters are increasing the classifiers' learning capabilities.

# 3. EXPERIMENTS

This chapter contains the process journey for my thesis. In the first section, the project and its goals have been explained. It is followed by description of dataset used in this research. The third section contains information on the modules that are used in experimentation. The fourth section describes the experimental environment.

## 3.1 Project Information

The goal of this project was to help patent engineers find out relevant information from the OAs without the hassle of going through all the content. There were two main types of contents that were to be extracted: citation numbers and examiner's comments. Citations identified by the examining division are informed using the patent application publication number or patent number. Examiner's comments inform the applicant how his/her patent application cannot be granted because of the findings in the citations. Examiner's comments refer to specific sections in the citations which he sees describes the similarities against claims of the filed patent application.

As the project had to be developed from scratch, there were various minor objectives set in order to achieve the final goal. Firstly, research had to be done on how the PE separate the information in the OA received and how they were able to find out the relevant information from the documents. Secondly, the documents had to be researched thoroughly to find a way to extract citation numbers. After gathering enough information on how to classify the data and find citation numbers, the next step was to proceed with creating the dataset. A general classification workflow[49] was used in the research can be seen in Figure 3.1

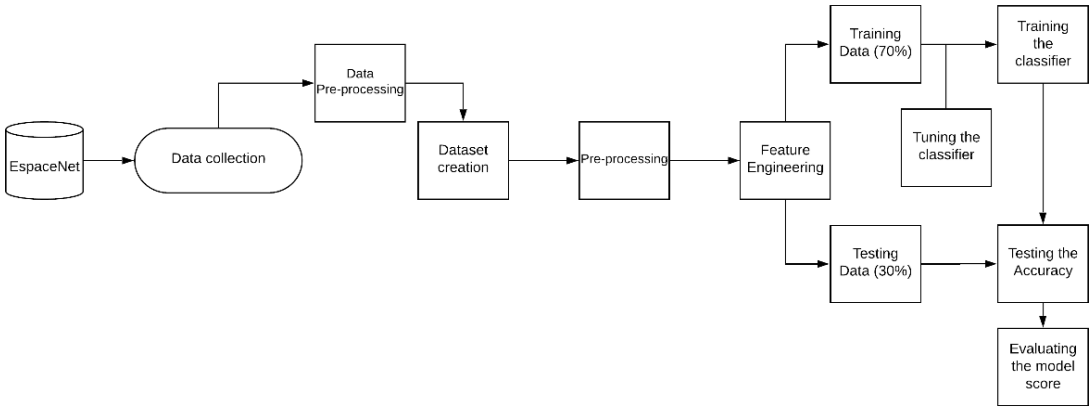


***Figure 3.1*** Research Workflow

## 3.2 Dataset

Various patent research datasets have been made publicly available for researchers to work. However, the patent process has multiple segments and an OA is one of the segments that has remained surprisingly unscathed. As a result, not a lot of office action research datasets are available for researchers, including dataset of office actions for European patents. Thus, to proceed with the project, a way had to be found to create the dataset from scratch.

**Data Collection**

It was found that the OAs are stored on the patent authorities' websites along with each application. Thus, OAs had to be retrieved from each application one by one and download it onto the system. Among various patent websites, Espacenet[1] was chosen for this project. There were two reasons for selecting Espacenet as a source of my data. Firstly, major focus in this project was the European patent application OAs. Espacenet is hosted by the European Patent Organization, that is it deals mostly with European patent applications. Secondly PEs prefer using Espacenet rather than any other patent database, which makes it a reliable source of information. All the process of collecting the OAs had to be done manually as the documents were stored as a part of application.

An original OA can be seen in Appendix A. The document demonstrates the complete picture of what is received by the PE. Figure 3.2 and Figure 3.3 illustrate the parts of an OA that were to be focused on in this research. Figure 3.2 shows us citation numbers that are mentioned by the examiner. The number written as "US 2014/177468 A1" refers to a published US patent application which, according to the examiner, contains similarities with the claims of filed application. Figure 3.3 mentions the examiner's comments while comparing the citation and the filed application. The comments from the examiner contain the paragraph numbers from the citation documents and a brief explanation of what claims are explained in those paragraphs.

As mentioned in Figure 3.3, the examiner states that content of the claims filed by the applicant already exist in 'document D1 in paragraph [0047]'. This information mentioned by the examiner proves that the applicant's claims are not novel and/or having no inventive step and they need to be amended in order to get a granted patent. Figure 3.4 depicts an example of a paragraph with irrelevant information. Unlike the previous figure, this one does not mention any information regarding the similarity of claims.

[1] The website that I used to collect the data can be viewed from this link: https://worldwide.espacenet.com/advancedSearch?locale=en_EP

1 **Cited documents**

Reference is made to the following documents; the numbering will be adhered to in the rest of the procedure.

D1 US 2014/177468 A1 (KAZMI MUHAMMAD [SE] ET AL) 26 June 2014 (2014-06-26)

D2 US 2015/004979 A1 (SZALAY ZOLTAN [HU] ET AL) 1 January 2015 (2015-01-01)

D3 WO 2011/069550 A1 (ERICSSON TELEFON AB L M [SE]; KLEINHENZ THOMAS [DE]; BACHL RAINER [DE]) 16 June 2011 (2011-06-16)

***Figure 3.2*** Citation numbers in an OA

The present application does not meet the requirements of Article 52(1) EPC because the subject-matter of claim **1** is not new within the meaning of Article 54(1) and (2) EPC.

**D1** discloses a method, comprising:

- measuring, by a user device, a wireless link between the user device and an access point to obtain a first set of measurement values (`Fig. 5, paragraph [0047], "circuitry that measures received signal strength or makes other received signal measurements");`

***Figure 3.3*** Examiner's comments in an OA (relevant)

The independent claims are not in the two-part form in accordance with Rule 43(1) EPC, which in the present case would be appropriate, with those features known in combination from the prior art (**D1**) being placed in the preamble (Rule 43(1)(a) EPC) and the remaining features being included in the characterising part (Rule 43(1)(b) EPC).

***Figure 3.4*** Examiner's comment in an OA (irrelevant)

2300 OAs were gathered from Espacenet. Generally, a few OAs are received for one patent application. Since a patent is granted in a few years, the number of OAs received by an organization depends on how many applications they file. Given this information, the number of applications filed, OAs received etc were researched upon and discussed with other team members to come up with an appropriate number of OAs to create the dataset. Once the appropriate number of OAs were collected to create the dataset, their structure was researched. New OAs received would be added to the dataset to increase the training data.

### Data Research

After collecting the OAs, the data was analysed manually to familiarize myself with the fashion of examiner's writings. The documents did not have a pre-defined template, so all examiner's wrote in different ways. Yet, the numbering was one element that was followed by everyone. Examiners would categorize different sections under different numbers and headings. After viewing through various documents, the structure was identified. During this stage various keywords and phrases that precisely differentiated the relevant sections of information were also identified.

### Creation of dataset

Most of the documents received by the PE are in the format of image PDFs or non-searchable PDFs. Processing these types of documents required character recognition, thus tesseract [42] module which provides the OCR engine to parse through the non-searchable PDFs was used. The response from this module comes in various formats like TEXT, PDF, TSV and hOCR. The data research led to the conclusion that relevant information can be segregated if the structure of OA can be specified. The hOCR was chosen as an output format. This option allowed me to convert the OAs into an XML file. Figure 3.5 shows how the PDF document is converted into a structured document with named classes and division. A larger XML output can be seen in Appendix C.

```
</div>
<div class='ocr_carea' id='block_1_3' title="bbox 221 186 1522 222">
 <p class='ocr_par' id='par_1_3' lang='eng' title="bbox 221 186 1522 222">
  <span class='ocr_line' id='line_1_3' title="bbox 232 186 1238 215; baseline 0 -8; x_size 21; x_descenders 4; x_ascenders 4">
   <span class='ocrx_word' id='word_1_14' title='bbox 232 186 276 215; x_wconf 96'>Date</span>
   <span class='ocrx_word' id='word_1_15' title='bbox 732 186 795 215; x_wconf 96'>Feuille</span>
   <span class='ocrx_word' id='word_1_16' title='bbox 1236 209 1238 211; x_wconf 0'>.</span>
  </span>
  <span class='ocr_line' id='line_1_4' title="bbox 221 205 1522 222; baseline 0 0; x_size 22.296295; x_descenders 5.2962961; x_ascenders 4">
   <span class='ocrx_word' id='word_1_17' title='bbox 217 201 287 232; x_wconf 23'>—$$</span>
   <span class='ocrx_word' id='word_1_18' title='bbox 221 205 1522 222; x_wconf 0'>sss</span>
   <span class='ocrx_word' id='word_1_19' title='bbox 1126 201 1204 232; x_wconf 0'>omande</span>
   <span class='ocrx_word' id='word_1_20' title='bbox 1214 201 1525 232; x_wconf 68'>no</span>
  </span>
 </p>
</div>
```

***Figure 3.5*** XML output from Tesseract

The structured documents were researched through again to find out the appropriate names of divisions and classes that would surely divide the documents into the required paragraphs. Using the names of classes and divisions, all the words were combined to create statements which in turn created paragraphs. The assembled paragraphs were then stored in an Excel sheet with paragraph identification numbers. Once all the documents were processed, a raw dataset of almost 100,000 paragraphs was received. Figure 3.6 shows the paragraph identification numbers and its contents from the raw dataset. This dataset was further cleaned, pre-processed, and manually annotated.

| | | | |
|---|---|---|---|
| 124155 | **124153** | 999_24 | 2.2 It is not clear whether the step of "storing previously updated parameter values" implies a fixed sequence o |
| 124156 | **124154** | 999_25 | 2.3 Regarding claim 2, it is noted that the terms "coefficients" and "magnitude" are provided without their conte |
| 124157 | **124155** | 999_26 | 2.4 Similarly, the term "pass" in claims 3 and 4 is not clear, because it is provided without a context (JPEG2000 Tie |
| 124158 | **124156** | 999_27 | 2.5 The same objections apply to claims 7-10 mutatis mutandis. |
| 124159 | **124157** | 999_28 | 3 The present application does not meet the requirements of Article 52(1) EPC because the subject-matter of th |
| 124160 | **124158** | 999_29 | 3.1 Document D1 discloses a method comprising: |
| 124161 | **124159** | 999_30 | ° selecting a datastream among a first datastream and a second datastream, said first datastream and said secon |
| 124162 | **124160** | 999_31 | EPO Form 1703 01.91TRI |
| 124163 | **124161** | 999_32 | Datum Blatt Anmelde-Nr: |
| 124164 | **124162** | 999_33 | Date cf Form 1507 Sheet 3 nDplication 16 827 311.8 |
| 124165 | **124163** | 999_34 | Date Feuille . —$—$$ Demande no |
| 124166 | **124164** | 999_35 | ° |
| 124167 | **124165** | 999_36 | obtaining a context - decision pair from the selected datastream and an indication of the selected datastream (f |
| 124168 | **124166** | 999_37 | ° |
| 124169 | **124167** | 999_38 | using the datastream indication to select a set of registers containing parameter values relating to the selected |
| 124170 | **124168** | 999_39 | ° |
| 124171 | **124169** | 999_40 | providing the parameter values from the selected set of registers to arithmetic encoding to form updated param |
| 124172 | **124170** | 999_41 | ° storing previously updated parameter values to a set of registers indicated by a previous datastream indicatior |
| 124173 | **124171** | 999_42 | Therefore, the subject-matter of claim 1 is not new. For the independent claim 7, the above objection applies m |
| 124174 | **124172** | 999_43 | 4 The dependent claims do not appear to contain any additional features which, in combination with the feature |
| 124175 | **124173** | 999_44 | 4.1 As already mentioned, claim 2 is not clear with respect to what exactly is stored and where it is stored. Since |
| 124176 | **124174** | 999_45 | 4.2 Claims 3 and 4 do not disclose any features that would distinguish their subject-matter from D1. As is shown |
| 124177 | **124175** | 999_46 | 4.3 Claim 5 discloses the standard JPEG2000 passes, which are also disclosed inD1. |
| 124178 | **124176** | 999_47 | 4.4 The subject-matter of claim 6 is disclosed in D1 (e.g. see the sequence of passes in figure 14). |
| 124179 | **124177** | 999_48 | 45 The remaining features are defined in an analogous manner, and the above objections apply mutatis mutand |
| 124180 | **124178** | 999_49 | 5 The applicant is invited to consider the further comments below: |
| 124181 | **124179** | 999_50 | EPO Form 1703 01.91TRI |

***Figure 3.6*** Raw dataset

Cleaning and processing of the above shown raw dataset resulted in a total of about 40,000 relevant and irrelevant paragraphs. The difference in the relevant and non-relevant information in an OA is substantial, which was clearly visible in our processed dataset. The clean dataset had almost 30,000 irrelevant paragraphs with only 6000 relevant ones. Figure 3.7 represents the considerable difference in the numbers.

The label “0” represents the number of irrelevant paragraphs in our dataset while the label “1” represents relevant paragraphs. The PE must go through all the irrelevant information in the OA to retrieve a small amount of relevant information.

## 3.3 Research Modules

As mentioned earlier, the documents received by the CRs are in the non-searchable PDF format. For analysing the text from these documents, they had to be passed through an OCR engine. For my research the Tesseract OCR engine [43] was used as it is one of the most accurate[2] free OCR engines. Although the engine was developed in early 90s, it is still being used by the researchers in the character recognition domain like Vehicle tracking system [44], Meme Categorization [45] and a lot more. All this research

[2] A case study on various OCR engines. https://www.researchgate.net/publication/310645810_OCR_as_a_Service_An_Experimental_Evaluation_of_Google_Docs_OCR_Tesseract_ABBYY_FineReader_and_Transym

largely depends on Tesseract's result which makes it a reliable engine for our research. The architecture for the Tesseract engine, which can be seen in Figure 3.8, is taken from the case study of the OCR tool [46].

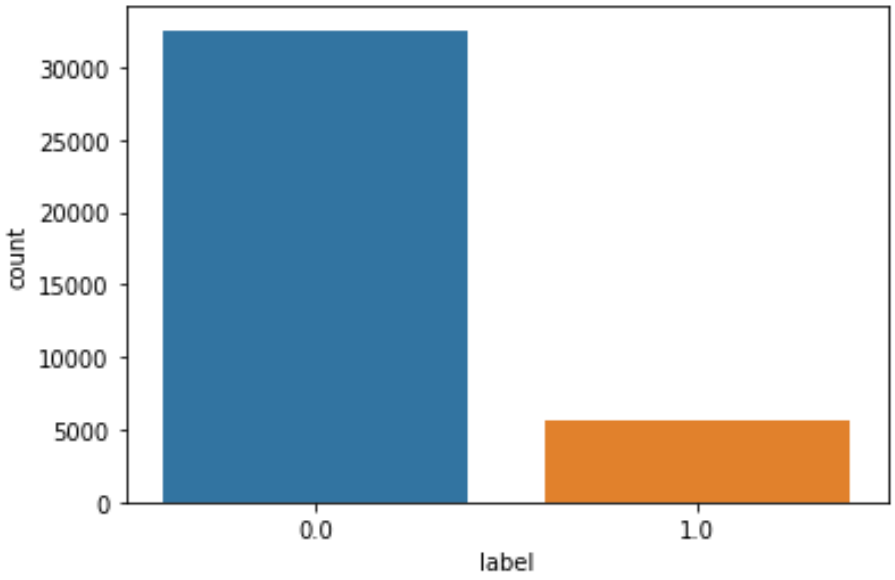


***Figure 3.7*** Number of relevant and irrelevant data rows

The text extracted from Tesseract is in the XML format which is then saved to create the XML version of the OA received. Since this XML version is readable, it is then processed using the BeautifulSoup [47] module in Python. Using the methods explained in the book *Website Scraping with Python* [47] the elements of XML files were read, parsed, and convert into the real-world structured text. Now that the real-world data was acquired, next stages of processing the data could be proceeded. According to the *Handbook of Natural Language Processing [48]*, Natural language Processing categorizes the language analysis into various stages. While analysing the real-world data, various stages/processes are very useful and important when we deal with the state-of-the-art algorithms. The analysis stages in NLP generally include tokenization, lexical analysis, syntactical analysis, pragmatic analysis which gives us the whole understanding of any statement. It is not always that all the steps are required to achieve the results.

For my research I was able to work around the tokenization and lexical analysis to create the appropriate workflow which helped me in feature engineering and classification stages.

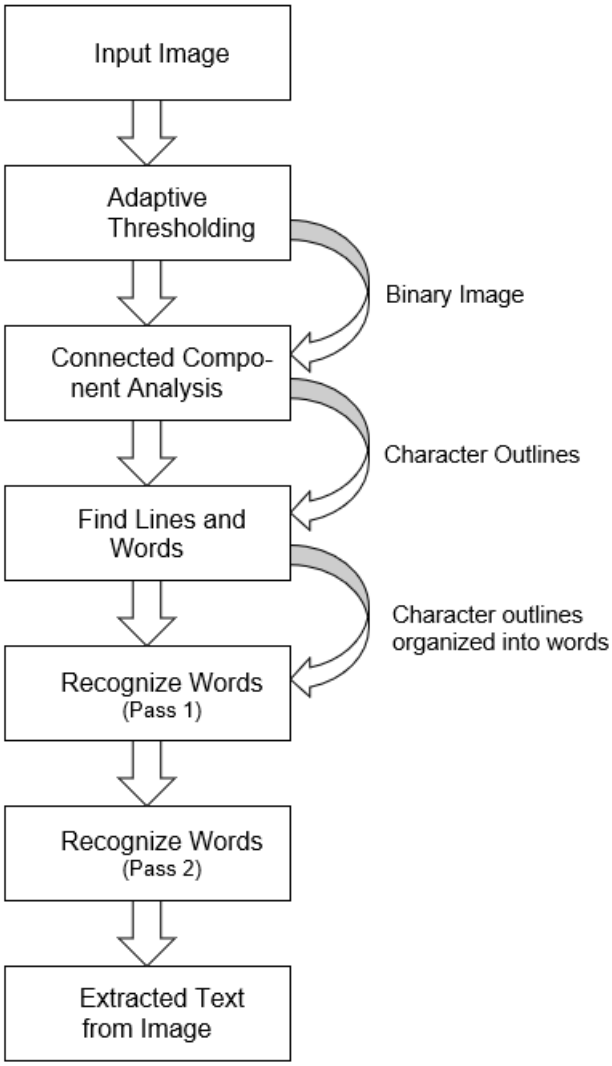


***Figure 3.8*** Architecture of Tesseract OCR

Along with tokenization, another basic technique was used to find out the relevant information from the OAs. As mentioned earlier, main focus of this research was on two parts of OAs: citation numbers and examiner's comments. For the citation number regular expressions were employed. According to the author [49], "a regular expression is a string containing a combination of normal characters and other metacharacters or meta sequences". Meta-characters and sequences are characters which can be utilized to match various patterns in the text. In the research, the pattern in the citation number like "US 2014/177468 A1" were recognized as a combination of alphanumeric characters in a specific pattern. After reading through multiple documents, the regex patterns were shortlisted to just three. The citation numbers recognised were cleaned of any noise and then matched to the regex pattern. The regex patterns used in the Python scripts are:

- "\d\d\d\d\d\d\d\d\d"
- "\w\w\d\d\d\d\d\d\d\d\d"

- “\d\s\d\d\d\s\d\d\d”

“ \d ” matched with digits, “ \w ” – alphabetic character and “ \s ” – spaces in corresponding positions. After processing the citation numbers, most convenient approach to work on paragraphs in the OAs had to be found.

The paragraphs were created by combining sets of words. While manually annotating the paragraphs in the dataset, various sets of keywords and phrases were found which made it easier to classify them. In the NLP terms, these phrases are referred to as N-grams, where “N” determines the number of words in a set. In the N-gram model, we directly try to figure out the probability of upcoming words based on n-1 previous words.

$$P(Wi/W\,i-1) = Count(Wi-1, Wi)/Count(Wi-1) \qquad (3.1)$$

The probability of words can be calculated using the aforementioned formula. N-gram models were the beginning of experimentation stage. Unigrams (1-gram), bi-grams (2-grams), trigrams (3-grams) and so on were used to create different sets of features which could be further fed to feature selection algorithms to get the most optimal features for training. Unlike humans, machines can just understand numbers fed to them and not words or characters. To create a feature set, these characters and words needed to be converted into a machine-readable format, that is to convert them into vectors and arrays of numbers. The most common techniques used to convert words or tokens into vectors are Count Vectorizer (CV) and Term Frequency and Inverse Document Frequency (TF-IDF) vectorizer.

Count Vectorizer is the most straight forward tool that counts the number of words in a document and assigns them as the weight of the word. CV has various other parameters like *n-grams* to create the n-gram features, *lowercase* to change all the information to lowercase etc., which can be modified while using the vectorizer and weighing the terms. The weighted words are then used as features by the algorithms to properly classify the information. Since CV focuses on the number of times a word has occurred, it can sometimes prove to be unreliable. The common terms like “The”, “I”, “is”, “a” etc. are among the most occurring words which allows them to get more weight values, but they are not very useful when providing actual meaning to the content of the document, hence the classifier can miss more important and interesting features. In order to adjust the weights in the corpus of words and get better features, the TF-IDF vectorizer was chosen.

TF-IDF gives the optimal weight for a word as it compares the number of times a word has appeared with the number of documents it has appeared in. The results from TF-IDF are created from two separate matrices [50] which can be seen from the equation below:

$$TF - IDF = TF(t, d) * IDF(t, d) \quad (3.2)$$

The frequency term is denoted as *TF (t, d)* and is calculated as a summation of number of times the term '*t*' appears in the document '*d*'.

$$TF(t, d) = \sum_{x \in d} f(x, t) \quad (3.3)$$

*IDF* (*t, d*) denotes the inverse document frequency and provides us with the information of how many documents the term "*t*" appears in. The inverse frequency is calculated using the average and the logarithms [50].

$$IDF(t, d) = 1 + \log \frac{T}{(1 + df(t))} \quad (3.4)$$

Here, the total number of documents substitute the term "*T*" and the number of documents with the term "*t*" present in them replaces "*df(t)*". The product of TF (t, d) and IDF (t, d) will compute the final TF-IDF metric. Similar to its counterpart CV, TF-IDF also has various parameters like *n-grams, stop words* etc., which are tuned for acquiring the resulting weighted metric. Once the feature metric is achieved, we can feed this metric to the classification algorithms after performing a few feature selection methods.

## 3.4 Classification Models

A lot of research has already been done on text and patent classification using various Scikit algorithms, multilayer perceptrons (MLPs), convolutional neural networks (CNNs) and so on. As mentioned earlier, the main obstacle for this research was creation of the dataset. After creating the dataset, a general text classification approach shown in Figure 3.1 was followed.

The word embedding models TF-IDF and CV discussed in the previous section were majorly used for feature creation. Other than that, four classification models namely Gaussian Naive Bayes, Random Forest, AdaBoost and XGBoost were used. These four models were chosen as they used theorems and techniques distinct from each other to classify the text.

Gaussian Naïve Bayes works on Bayes theorem [51]. Naive Bayes works on the probabilistic functions. Based on the probability of the features occurring in a paragraph, the classifier would try to predict if the new paragraph is relevant or not.

Random Forest algorithm works on the foundation of decision trees, that is the features from the training data are divided into tree like structures to decide which data would be categorized to different classes. Table 3.1 presents representation of the features that were identified while researching the cleaned data.

| Features | Label |
|---|---|
| Feature A | 1 |
| Feature B | 1 |
| Feature C | 1 |
| Feature D | 0 |
| Feature E | 0 |

***Table 3.1*** Features used to label the training data

Figure 3.9 explains the internal processing done in the Random Forest classifier. T1 and T2 are two trees in the classifier, and they decide the labels of the features in the shown way.

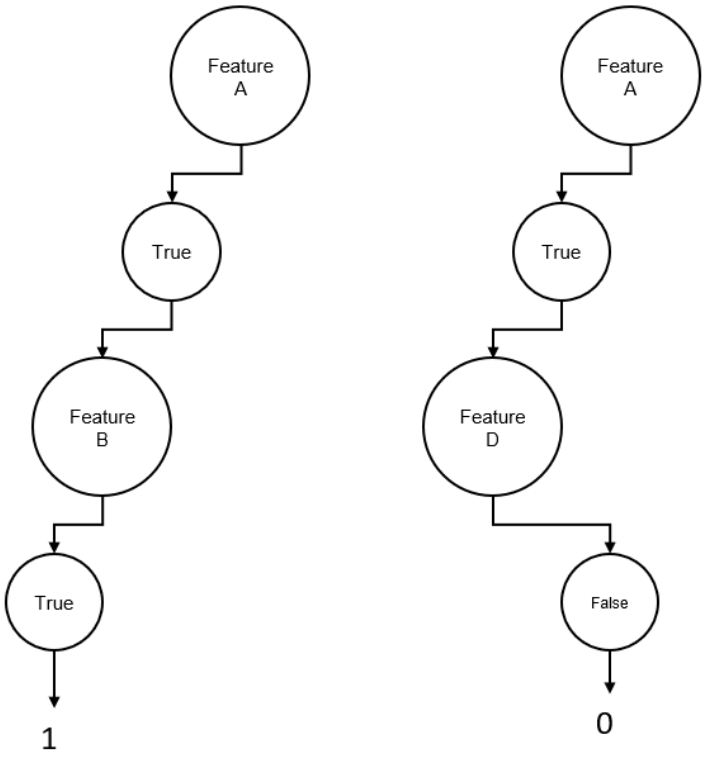


***Figure 3.9*** Two example trees from Random Forest classifier

In both trees, the first feature was classified as true, that is it was relevant, but the second feature for the trees decided if the complete information was relevant or not. Similarly, the algorithm works for all the features and data provided.

The AdaBoost algorithm works by fitting the classifier on the original dataset and then the algorithm creates multiple classifiers and start training them on the dataset in iterations. The first classifier is trained on the original dataset with some weights being set to the features. The copies of classifiers are then fit with adjusted weights of the features so that the final classifiers would be able to learn the best features. Figure 3.10 shows the architecture of the AdaBoost classifier and how it is fitted.

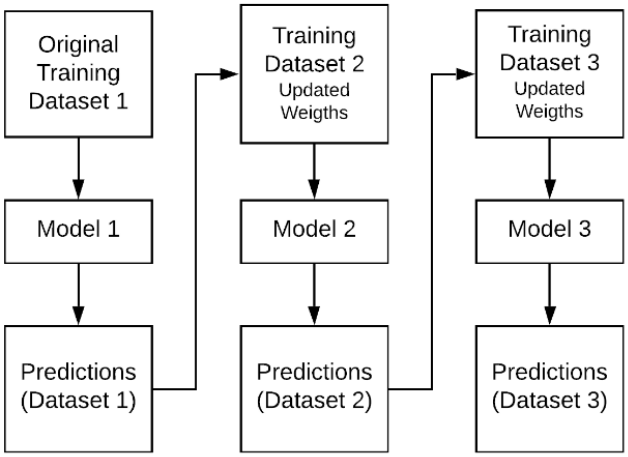


***Figure 3.10*** Training process in the AdaBoost and XGBoost algorithms

XG boost classifier works on gradient boosting technique to be highly effective, efficient and flexible. It also provides parallel tree boosting which makes it more effective than the Random Forest and Ada Boost models.

## 3.5 Setting up the Experiments

### 3.5.1 Data Pre-processing

For data pre-processing, OA was converted into the readable XML format and provided as an input to the scripts. The task was to classify the information in the OA as relevant and irrelevant, still the OA contained information that would not contribute to any classification task. The data had to be cleaned of the stop words, punctuation marks and other noise such as headers and footers of the OAs. They consisted of different information related to the application filed which were completely irrelevant for classification. In this case, the advantage was that these unnecessary elements were always written in the same way. Figure 3.11 and Figure 3.12 show examples of how the header and footer in an OA look like.

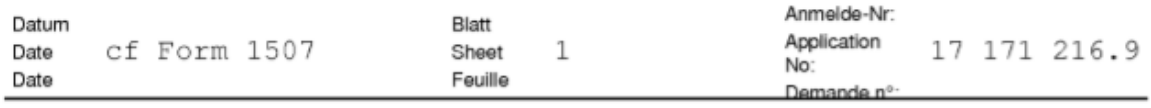


***Figure 3.11*** Header of an OA

***Figure 3.12*** Footer of an OA

The dataset was cleared of stop words and punctuation marks using the NLTK module, but the headers and footers were cleaned using the conditional statements. After a thorough check of the dataset and information in it, the hyperparameters of various models were tuned and the ones which gave the best results were opted. The steps performed in pre-processing can be seen in the architecture below:

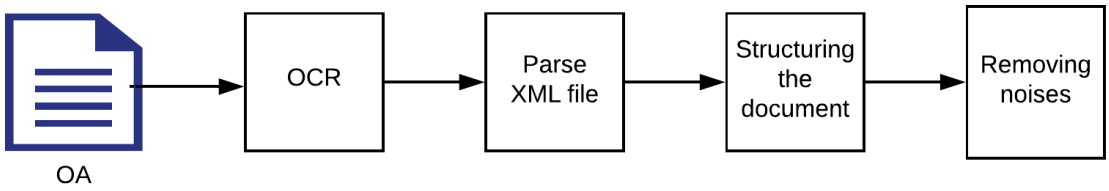


***Figure 3.13*** Pre-processing steps involved

## 3.5.2 Experimental Environment

The experimental results largely depend on the environment they are performed in. Variation in processing units, RAM, GPU, development environment etc. can alter the results in different ways. For example, training of the Gaussian Naive Bayes classifier would be faster on an i7 core processor than the one used in the experiments. Detailed information about the environment is given in Table 3.2. Spyder was chosen as the development environment and python as the programming language. Spyder is one of the most common scientific environments that is being used by researchers, scientists, and other professionals in the field of machine learning. It features various components such as variable explorer, profiler, editor, IPython console, debugger etc., which helped me to analyse the Python scripts efficiently. Since it supports various machine learning libraries and NLP modules, it became a lot easier to implement the models into Python scripts.

| Experimental Environment | Tool |
|---|---|
| Operating System | Windows 10 |
| Programming language | Python 3.7 |
| RAM | 24 GB |
| Processor | Intel i5 8th gen |
| GPU | Nvidia RTX 1050 |
| Spyder | 4.1.2 |

***Table 3.2*** Environment

# 4. EVALUATION

This chapter contains the description of feature engineering, hyperparameters tuning, training of the classification models, the results achieved and discussion.

## 4.1 Experimental results

**Model Training**

In the research, two word embedding models and four classification models were trained. The batch size was created using a train test split with the split value of 0.3, that is 26,000 paragraphs were used in the training set while 11,500 were used in the testing set. The word embedding models used were CV and TF-IDF. After fitting the models with the training set, different numbers of features were received which can be seen in Table 4.1.

| Embedding | Parameter | Number of features |
|---|---|---|
| Count Vectorizer | Unigrams | 18874 |
| | Bigrams | 265170 |
| | Trigrams | 458098 |
| | 4-grams | 543458 |
| | Bigrams and Trigrams | 723268 |
| TF-IDF vectorizer | Unigrams | 18874 |
| | Bigrams | 266570 |
| | Trigram | 451979 |
| | 4-grams | 543458 |
| | Bigrams and trigrams | 723268 |

***Table 4.1*** Total number of features

As seen in the table above, there were a good number of features but using all of them resulted in overfitting the classification models. For overcoming this obstacle, the feature selection technique was used to experiment with the different percentile of features. Memory constraints also played a vital role in deciding the percentage of features to be used. For baseline results, all the features mentioned in Table 4.1 were experimented on using Gaussian Naive Bayes. After experimenting with different permutations of feature selection and n-gram selection in the naive Bayes model, a baseline for the experiment was defined. The complete set of baseline results can be seen in Appendix B. Table 4.2 contains the best result from the naive Bayes classifier. Once a baseline was set, the remaining classifiers were tuned and the results were studied. For hyper tuning the models, Grid Search module was used. Random Forest, AdaBoost and XGBoost were the remaining classifiers used in this research.

| Embedding | Features used (% of total) | Accuracy |
|---|---|---|
| Count Vectorizer | 14210 Unigrams (75) | 88.89 |
| | 5277 Bigrams (2) | 90.99 |
| | 4613 Trigrams (1) | 93.17 |
| | 5411 4-grams (1) | 93.14 |
| | 5411 bi & trigrams (1) | 91.82 |
| TF-IDF vectorizer | 1858 Unigrams (10) | 87.78 |
| | 17248 Bigrams (10) | 87.54 |
| | 23255 Trigrams (7) | 86.76 |
| | 29411 4-grams (7) | 86.99 |
| | 25418 bi & trigrams (5) | 87.05 |

***Table 4.2*** Baseline results

Random Forest was chosen for the second experiment. Random Forest had multiple parameters such as, '*min_samples_leaf*' that tells us how many minimum leaf nodes should be present in the tree, or '*criterion*' identifying the quality of the data. These were tuned in order to obtain the best results. The parameters for Random Forest are noted in Table 4.3.

| Parameter | Value |
|---|---|
| Criterion | entropy |
| Min_samples_split | 2 |
| Min_samples_leaf | 1 |
| N_estimators | 100 |
| random_state | 0 |

***Table 4.3*** Hyperparameters for Random Forest

Similarly, AdaBoost and XGBoost had different hyperparameters that had to be tuned to achieve optimal results. Tables 4.4 and 4.5 demonstrate the parameter values for both models.

| Parameter | Value |
|---|---|
| N-estimators | 200 |
| Learning rate | 0.5 |
| Algorithm | SAMME.R |
| random_state | 2 |

***Table 4.4*** Hyperparameters for AdaBoost

| Parameter | Value |
|---|---|
| Learning rate | 0.5 |
| Booster | gbtree |
| N_estimators | 200 |
| Gamma | 0.5 |
| random_state | 0 |

***Table 4.5*** Hyperparameters for XGBoost

**Experimental Results**

As Gaussian Naïve Bayes set up a baseline, Random Forest model was experimented with to obtain better results. Numerous hyperparameters were tried before receiving satisfying results. Using the parameters shown in Table 4.4, better accuracy results in classifying the data were achieved. The results for Random Forest classification can be seen in Table 4.6.

| Embedding | Features used (% of total) | Accuracy |
|---|---|---|
| Count Vectorizer | 948 Unigrams (5) | 94.51 |
| | 13203 Bigrams (5) | 95.90 |
| | 9018 Trigrams (2) | 94.67 |
| | 10818 4-grams (2) | 93.22 |
| | 14422 bi & trigrams (2) | 95.73 |
| TF-IDF vectorizer | 1888 Unigrams (10) | 94.28 |
| | 13215 Bigrams (5) | 95.91 |
| | 46163 Trigrams (10) | 94.47 |
| | 27375 4-grams (5) | 93.41 |
| | 14485 bi & trigrams (2) | 95.65 |

***Table 4.6*** Results from Random Forest

It was clearly visible that the accuracy increased substantially after using the Random Forest algorithm. Now the features received from the Random Forest algorithm were compared to the manual features that used to annotate the data. The features used in unigrams and 4-grams were considerably different than those used while manual labelling. Because unigrams and 4-grams did not provide any promising features, the research was continued using bi-grams and trigrams for the remaining models. The topmost features received from the RF model can be seen in Figure 4.1.

The x-axis represents the names of the features from the model whereas the y-axis represents their importance, or the entropy values as perceived by the model. The features with higher entropy values are considered to be better by the classifier. Although, the values may appear to be very small, they were considered the most important features by the model.

From Random Forest and Gaussian Naïve Bayes, it was deduced that the unigrams and 4-grams are not that useful in classifying the paragraphs. The bigrams, trigrams, and their combinations were used as features for the remaining AdaBoost and XGBoost algorithms. The results from AdaBoost and XGBoost can be seen in Tables 4.7 and 4.8. Various experiments performed using different percentile of features yielded different result. The results shown in Tables 4.7 and 4.8 display the best of them.

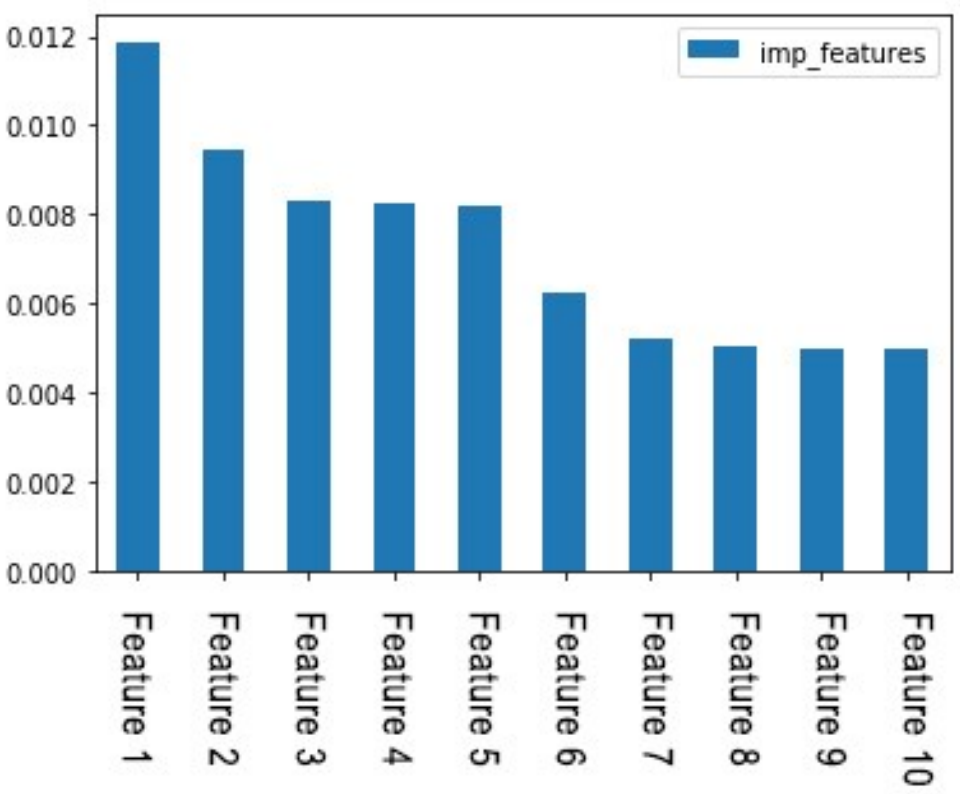


***Figure 4.1*** Importance of features from Random Forest model

| Embedding | Features used (% of total) | Accuracy |
|---|---|---|
| Count Vectorizer | 36287 Bigrams (15) | 95.23 |
| | 45155 Trigrams (10) | 94.28 |
| | 7183 bi & trigrams (1) | 95.77 |
| TF-IDF vectorizer | 39723 Bigrams (15) | 95.61 |
| | 13670 Trigrams (3) | 94.44 |
| | 21661 bi & trigrams (3) | 95.69 |

***Table 4.7*** Results from AdaBoost

| Embedding | Features used (% of total) | Accuracy |
|---|---|---|
| Count Vectorizer | 13670 Bigrams (5) | 96.18 |
| | 9104 Trigrams (2) | 94.87 |
| | 21596 bi & trigrams (5) | 96.39 |
| TF-IDF vectorizer | 13388 Bigrams (5) | 96.30 |
| | 4578 Trigrams (1) | 94.78 |
| | 14560 bi & trigrams (2) | 96.50 |

***Table 4.8*** Results from XGBoost

Compared to the baseline, results achieved from Random Forest, AdaBoost and XGBoost were significant. Since the models' results from different feature percentiles are

already shown in previous tables, Table 4.9 reflects the best results from them along with their F1 scores.

| Embedding | Classification Model | Accuracy | *F1 Score* |
|---|---|---|---|
| Count Vectorizer | Random Forest | 95.90 | 0.846 |
| | AdaBoost | 95.77 | 0.827 |
| | XGBoost | 96.39 | 0.865 |
| TF-IDF vectorizer | Random Forest | 95.91 | 0.850 |
| | AdaBoost | 95.69 | 0.842 |
| | XGBoost | 96.50 | 0.866 |

***Table 4.9*** Best results for classification

The variation in the accuracy and F1-scores of the three models was trivial. XGBoost had the best accuracy with 96.5% with the F1-score of 0.866. The Random Forest outperformed the AdaBoost by 0.14 %. The F1-scores for Random Forest and AdaBoost were 0.85 and 0.827, respectively. The adoption of different embedding models also made an impact on the accuracy of the models.

Table 4.9 shows the contrast in the accuracy of both the embedding models. Even though the deviation in the accuracy of both the embedding models is marginal, it still proves that using different embedding model does influence the classification accuracy of the model. TF-IDF model weights the features differently than the CV model which makes an impact when the model tries to learn the significant features. As observed from the Table 4.9. TF-IDF also has better results than CV in both accuracy and F1-scores. The prediction results from all the models were investigated in the experimentation as well. The confusion matrices for all the embedding and classification models with best results were calculated. Figure 4.2 and 4.3 show the confusion matrices from the Random Forest model for CV and TF-IDF embedding models. The confusion matrices from AdaBoost can be seen in Figure 4.4 and 4.5. The best results for both embedding models were achieved from the XGBoost classifier. Figure 4.6 and 4.7 show the confusion matrices from the experiments performed using XGBoost classifier. Most of the mis-classified information in the confusion matrices can be seen in the relevant classes. The Random Forest classified 131 and 100 relevant paragraphs as irrelevant while predicted 1315 and 1317 paragraphs correctly using CV and TF-IDF models respectively. Similarly, AdaBoost classified 114 and 128 relevant paragraphs as irrelevant and classified 1282 and 1349 precisely. XGBoost provided with the best prediction numbers with 112 and 173 relevant paragraphs as irrelevant, at the same time classifying 1359 and 1451 paragraphs correctly. From all the confusion matrices we can see that there were a lot of relevant paragraphs that were being classified wrongly. One possible explanation of this can be the unbalanced training set which can be seen in Figure 3.6.

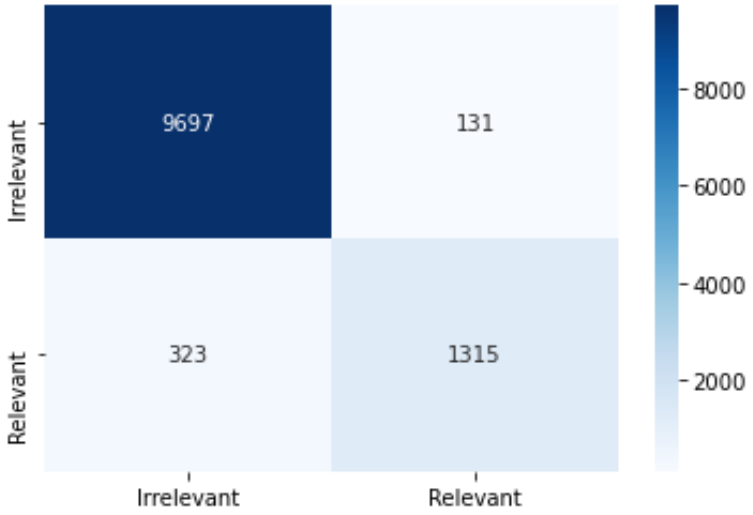


***Figure 4.2*** Confusion Matrix from Random Forest (CV)

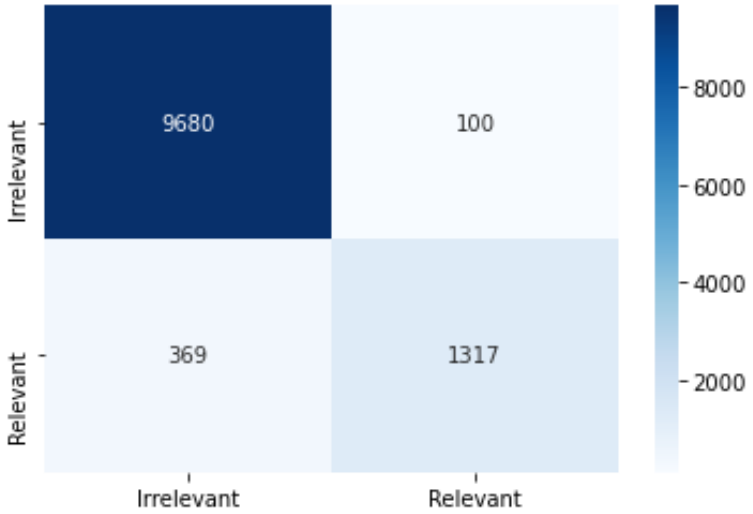


***Figure 4.3*** Confusion Matrix from Random Forest (TF-IDF)

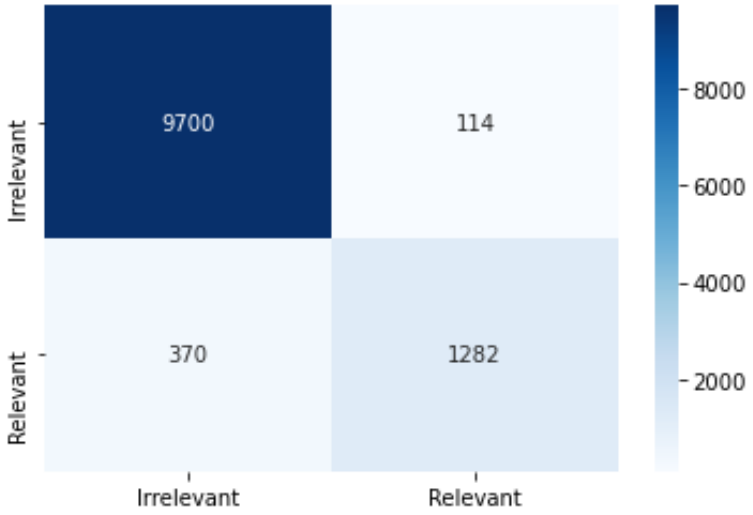


***Figure 4.4*** Confusion Matrix from AdaBoost (CV)

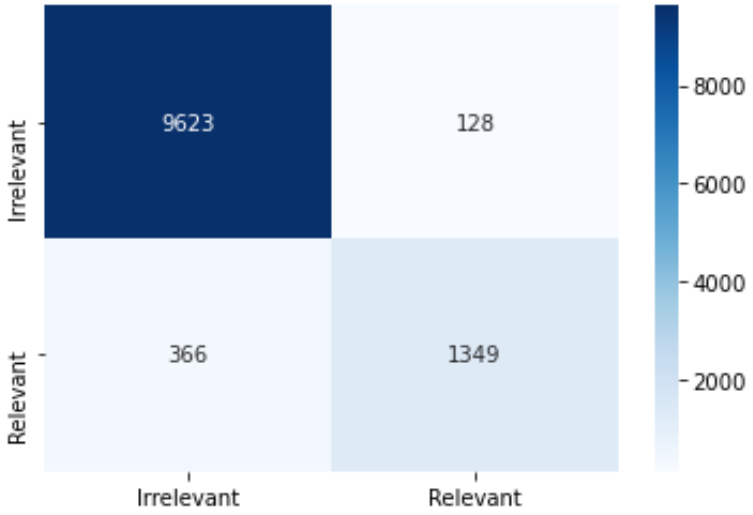


***Figure 4.5*** Confusion Matrix from AdaBoost (TF-IDF)

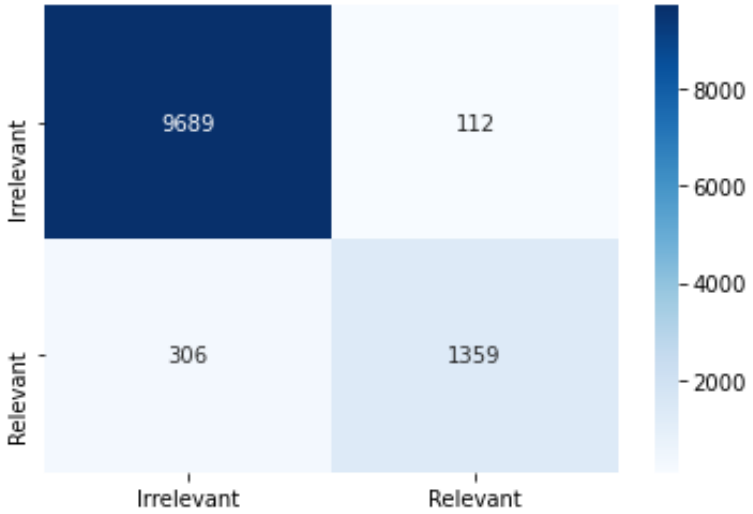


***Figure 4.6*** Confusion Matrix from XGBoost (CV)

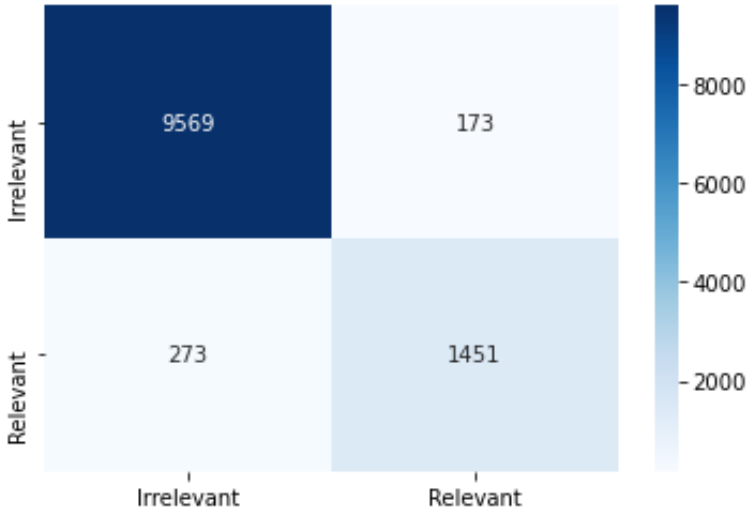


***Figure 4.7*** Confusion Matrix from XGBoost (TF-IDF)

## 4.2 Discussion

The incorrectly classified paragraphs from all the models were meticulously analysed. The analysis indicated that the paragraphs with contradictory features were wrongly classified. For example, below is a paragraph which is classified as relevant by the XGBoost model but is in fact an irrelevant paragraph.

“4.1.6 The subject-matter independent Claim 1 therefore involve inventive step vis-a-vis disclosure D1 combination D2 Claim 1 thus allowable according Articles 52 (1) 56 EPC”

The paragraph contains the features “A” and “B” which may deem this as relevant, but the whole paragraph does not contain any information about the examiner’s comments or other relevant information that might be useful for the PE. In this case the model has wrongly classified the paragraph on the basis of a few n-grams in the paragraph. Even though XGBoost gave the best accuracy among the three models, there are times when its result is incorrect. Currently, the models use only n-grams as their feature to classify the paragraphs in the OA. If n-grams are combined with other features such as length of the paragraphs, frequency of words etc., it might be possible to improve the results. Since OAs are written by law experts, the terminology and language used in them is clean and precise. Syntactic analysis from OAs might also provide some insight on the meaning of the statements by the examiner. Syntactical analysis can then help provide further features and improve the classification of the models. Similarly, the dataset can also be modified to perform classification based on the sentiments of the examiners. Features in NLP can be the number of tokens, length of tokens, relationship between the tokens, length of a specific type of tokens etc. These features are decided after researching the data. The features that can help reach the best results are then chosen for model training. In different language processing problems, the best results are achieved after reaching a harmony between the quality of features and the quantity of features. There may be situations which would give the best results from using just one feature and results would degrade on using more features while in other cases it can be completely opposite. In this research, one of the biggest setbacks was the size and unevenness of the dataset. This unevenness of the dataset led to variation in F1 scores of both relevant and irrelevant classes. The classification reports for the classes also showed a slightly lower rate for relevant class than the irrelevant class.

**Future Work**

This thesis is devoted to information extraction from office actions using NLP techniques. There are a few limitations to this research. The first and foremost is the dataset size. Limited resources and manpower allowed me to create a small dataset which is a huge

limitation to this research. Addition of new OAs will increase the size of dataset which will facilitate the implementation of new embedding techniques and machine learning algorithms. Because the OAs contain a lot more irrelevant information than the relevant one, the number of OAs added should be substantial. It should be overseen that the number of relevant paragraphs is also increased in the dataset to balance out the irrelevant ones. Another limitation to this research is the type of OAs. Currently, the research focuses on just European OAs. The OAs are classified also on the basis of the regions such as the USA, APAC, China etc. Different examiners send OAs from different regions and their structure can vary substantially. These OAs can also be studied in future. This research can be further developed into a commercial application which can be used by different PEs, easing their daily tasks. Since there is a lot of software to help work with patent applications and processes, this application can even be integrated into some software to improve its results. Another way to overcome the setbacks of the dataset is to find out more features such as length of paragraphs, frequency of keywords etc. and even the combination of features to improve the accuracy.

The dataset can be analysed for sentiment classification. The relevant and irrelevant paragraphs seem to have different sentiments. The dataset can then be labelled and classified into different sentiments. This research can also be approached as sentiment classification.

# 5. CONCLUSION

In this thesis, a new dataset for European OAs was created. This dataset was further processed to classify the information from OAs into relevant and irrelevant classes. The performance of three different types of classification algorithms (Random Forest, AdaBoost, XGBoost) was evaluated with two word embedding models (CV, TF-IDF). Moreover, the misclassified information was closely analysed to find out the reasons for misclassification.

**European OA Dataset**

The OA dataset used in this research was created for the classification of the information from OAs. Lack of a ready-made OA database and manpower resulted in a small research dataset. Given more resources, this dataset can be further increased in size for the implementation of state-of-the-art methodologies such as CNN, MLPs etc. The labelling of new OAs added to the dataset would still require human interference.

**Comparison of Classification Algorithms' Performance**

Since this was a newly generated dataset, the Gaussian Naïve Bayes classifier was first used to set a baseline for future experiments. Random Forest, AdaBoost and XGBoost algorithms were experimented using different number of features. XGBoost with the TF-IDF embedding gave the best accuracy as 96.5% with an F1-Score of 0.866. The Random Forest classifier had the best accuracy as 95.91% and an F1-Score of 0.850 with TF-IDF model. AdaBoost was able to achieve the best accuracy, 95.77% using CV embedding, but had a better F1-Score of 0.842 in the TF-IDF model.

**Embedding Model Comparison**

Through the experimentation in my thesis, it becomes clear that word embedding had an influence in classification of information. Even though the accuracy scores of classification models for both the embedding models were in a spitting distance, TF-IDF always outperformed CV when it came to F1-Scores.

**Misclassified Office Action Information**

From the analysis, it was clear that the unbalanced dataset played a critical role in misclassification of information from the three models. Because the amount of irrelevant information in the dataset was 5 times larger than that of relevant information, the models were not able to properly learn all the features from relevant paragraphs. If the number

of relevant paragraphs is increased in the dataset, current models can learn more information from relevant paragraphs and improve the classification scores. The results can also be improved by adding more features. The results shown above are from current feature set which includes bigrams, trigrams and their combination. These n-grams can be related with the length of relevant paragraphs, frequency of words, length of words etc. tom improve the results. If the syntactical analysis is performed on the given dataset, the meaning of statements in the OAs can reveal more features which can also be combined with the n-gram feature set resulting in improved classifiers. Moreover, the dataset can also be worked on for sentiment classification.

# REFERENCES


[1] World Intellectual Property Organization. What is Intellectual Property. Geneva, Switzerland: WIPO. Available from: https://www.wipo.int/about-ip/en/

[2] United States Patent and Trademark Office. Responding to Office Actions. 26 July 2019 [2 Jan 2020]. Available from: https://www.uspto.gov/patents-maintaining-patent/responding-office-actions

[3] Moens M-F. Information Extraction: Algorithms and Prospects in a Retrieval Context. Vol. 21, Information Extraction: Algorithms and Prospects in a Retrieval Context. Dordrecht: Springer Netherlands; 2006.

[4] Helmers L, Horn F, Biegler F, Oppermann T, Müller K-R. Automating the search for a patent's prior art with a full text similarity search. PloS one. 2019;14(3):e0212103–.

[5] Chiarello F, Cimino A, Fantoni G, Dell'Orletta F. Automatic users extraction from patents. World patent information. 2018; 54: 28–38.

[6] Ernst H. Patent information for strategic technology management. World patent information. 2003;25(3):233–42.

[7] Bowonder B, Racherla JK, Mastakar NV, Krishnan S. R&D Spending Patterns of Global Firms. Research-Technology Management. 2005 Sep 1;48(5):51–9.

[8] World Intellectual Property Organization. What is a Patent. Geneva, Switzerland: WIPO. Available from: https://www.wipo.int/patents/en/

[9] Rimai D. Patent engineering : a guide to building a valuable patent portfolio and controlling the marketplace . Hoboken, New Jersey: Scrivener Publishing; 2017.

[10] Altınel B, Can Ganiz M, Diri B. A corpus-based semantic kernel for text classification by using meaning values of terms. Engineering applications of artificial intelligence. 2015 Aug;43:54–66.

[11] Pinheiro RH., Cavalcanti GD., Tsang IR. Combining dissimilarity spaces for text categorization. Information sciences. 2017 Sep;406-407:87–101.

[12] Wang D, Wu J, Zhang H, Xu K, Lin M. Towards enhancing centroid classifier for text classification—A border-instance approach. Neurocomputing (Amsterdam). 2013 Feb 4;101:299–308.

[13] Petroni F, Piktus A, Fan A, Lewis P, Yazdani M, De Cao N, et al. KILT: a Benchmark for Knowledge Intensive Language Tasks. 2020 Sep 4;

[14] Prabhu Kaliamoorthi. Advancing NLP with Efficient Projection-Based Model Architectures. Google. 2020 September 21. Available from: https://ai.googleblog.com

[15] Liu X, Zhou Y, Wang Z. Recognition and extraction of named entities in online medical diagnosis data based on a deep neural network. Journal of visual communication and image representation. 2019 Apr;60:1–15.

[16] ChanJong Im, DoWan Kim, Thomas Mandl. Text Classification for Patents : Experiments with Unigrams, Bigrams and Different Weighting Methods. International JOURNAL OF CONTENTS. 2017;13(2):66–74.

[17] Alderucci D, Sicker D. Applying Artificial Intelligence to the Patent System. Technology and innovation. 2019 Aug 5;20(4):415–25.

[18] Luo L, Yang Z, Yang P, Zhang Y, Wang L, Wang J, et al. A neural network approach to chemical and gene/protein entity recognition in patents. Journal of cheminformatics. 2018;10(1):1–10.

[19] Helmers L, Horn F, Biegler F, Oppermann T, Müller K-R. Automating the search for a patent's prior art with a full text similarity search. PloS one. 2019;14(3):e0212103.

[20] Chiarello F, Cimino A, Fantoni G, Dell'Orletta F. Automatic users extraction from patents. World patent information. 2018;54:28–38.

[21] Mirończuk MM, Protasiewicz J. A recent overview of the state-of-the-art elements of text classification. Expert systems with applications. 2018;106:36–54.

[22] Pinheiro RH., Cavalcanti GD., Tsang IR. Combining dissimilarity spaces for text categorization. Information sciences. 2017 Sep;406-407:87–101.

[23] Altınel B, Can Ganiz M, Diri B. A corpus-based semantic kernel for text classification by using meaning values of terms. Engineering applications of artificial intelligence. 2015 Aug;43:54–66.

[24] Goldberg Y, Hirst G. Neural network methods in natural language processing . San Rafael: Morgan & Claypool Publishers; 2017.

[25] Xie F, Wu X, Zhu X. Efficient sequential pattern mining with wildcards for keyphrase extraction. Knowledge-based systems. 2017;115:27–39.

[26] Chang M, Poon CK. Using phrases as features in email classification. The Journal of systems and software. 2009;82(6):1036–45.

[27] Manning CD, Raghavan P, Schütze H. Introduction to information retrieval . Cambridge: Cambridge University Press; 2008.

[28] Jurafsky D, Martin JH. Speech and language processing : an introduction to natural language processing, computational linguistics and speech recognition . 2nd ed. Upper Saddle River (N.J.): Prentice Hall; 2009.

[29] SPARCK JONES K. A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. Journal of documentation. 1972;28(1):11–21.

[30] Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, Brown. Text Classification Algorithms: A Survey. Information (Basel). 2019;10(4):150–.

[31] Rocchio, J.J. Relevance feedback in information retrieval. In The SMART Retrieval System: Experiments in Automatic Document Processing; Englewood Cliffs: Prentice-Hall, NJ, USA, 1971; pp. 313–323.

[32] Cox, D.R. Analysis of Binary Data; Routledge: London, UK, 2018.

[33] Vapnik, V.; Chervonenkis, A.Y. A class of algorithms for pattern recognition learning. Avtomat. Telemekh 1964, 25, 937–945.

[34] Jiang, S.; Pang, G.; Wu, M.; Kuang, L. An improved K-nearest-neighbor algorithm for text categorization. Expert Syst. Appl. 2012, 39, 1503–1509.

[35] Morgan, J.N.; Sonquist, J.A. Problems in the analysis of survey data, and a proposal. J. Am. Stat. Assoc. 1963, 58, 415–434.

[36] Liu S-H, Liao H-L, Pi S-M, Hu J-W. Development of a Patent Retrieval and Analysis Platform – A hybrid approach. Expert systems with applications. 2011;38(6):7864–8.

[37] Park H, Ree JJ, Kim K. Identification of promising patents for technology transfers using TRIZ evolution trends. Expert systems with applications. 2013 Feb 1;40(2):736–43.

[38] Roh T, Jeong Y, Yoon B. Developing a Methodology of Structuring and Layering Technological Information in Patent Documents through Natural Language Processing. Sustainability (Basel, Switzerland). 2017 Nov 17;9(11):2117–.

[39] Helmers L, Horn F, Biegler F, Oppermann T, Müller K-R. Automating the search for a patent's prior art with a full text similarity search. PloS one. 2019;14(3):e0212103–.

[40] Chiarello F, Cimino A, Fantoni G, Dell'Orletta F. Automatic users extraction from patents. World patent information. 2018;54:28–38.

[41] Kim H, Joung J, Kim K. Semi-automatic extraction of technological causality from patents. Computers & industrial engineering. 2018 Jan;115:532–42.

[42] Github. tesseract. 7 March 2007 [17 July 2020]. Available from: https://github.com/tesseract-ocr/tesseract

[43] Smith R. An Overview of the Tesseract OCR Engine. In IEEE; 2007. p. 629–33.

[44] Lakshmanan L, Vora Y, Ghate R. Deep Learning Based Vehicle Tracking System Using License Plate Detection And Recognition. 2020 May 10;

[45] Amalia A, Sharif A, Haisar F, Gunawan D, Nasution BB. Meme Opinion Categorization by Using Optical Character Recognition (OCR) and Naïve Bayes Algorithm. In IEEE; 2018. p. 1–5.

[46] Patel C, Patel A, Patel D. Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study. International journal of computer applications. 2012 Oct 20;55(10):50–6.

[47] Hajba GL. Website Scraping with Python: Using BeautifulSoup and Scrapy. 1st ed. Website Scraping with Python. Berkeley, CA: Apress L. P; 2018.

[48] Indurkhya N, Damerau FJ. Handbook of natural language processing. 2nd ed. Boca Raton, Florida: Chapman & Hall/CRC; 2010.

[49] Stubblebine T. Regular expression pocket reference. 2nd ed. Sebastopol, California: O'Reilly; 2007.

[50] Kaur S, Kumar P, Kumaraguru P. Automating fake news detection system using multi-level voting model. Soft computing (Berlin, Germany). 2019;24(12):9049–69.

[51] Budhiraja SS, Mago V. A supervised learning approach for heading detection. Expert systems. 2020;37(4).

[52] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV. XLNet: Generalized Autoregressive Pretraining for Language Understanding. 2019 Jun 19;

[53] Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018 Oct 10;

[54] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019 Jul 26;

[55] Hill BM. Posterior Distribution of Percentiles: Bayes' Theorem for Sampling From a Population. Journal of the American Statistical Association. 1968 Jun 1;63(322):677–91.

[56] Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada 14–16 August 1995; Volume 1, pp. 278–282

[57] Freund, Y.; Kearns, M.; Mansour, Y.; Ron, D.; Rubinfeld, R.; Schapire, R.E. Efficient algorithms for learning to play repeated games against computationally bounded adversaries. In Proceedings of the 36th Annual Symposium on Foundations of Computer Science, Milwaukee, WI, USA, 23–25 October 1995; pp. 332–341.

# APPENDIX A: OFFICE ACTION DOCUMENT

| Datum Date Date | cf Form 1507 | Blatt Sheet Feuille | 1 | Anmelde-Nr: Application No: Demande n°: | 17 171 216.9 |
|---|---|---|---|---|---|

The examination is being carried out on the **following application documents**

**Description, Pages**

1-21 as originally filed

**Claims, Numbers**

1-15 as originally filed

**Drawings, Sheets**

1/9-9/9 as originally filed

Reference is made to the following documents; the numbering will be adhered to in the rest of the procedure.

D1 WO 2010/128909 A1 (ERICSSON TELEFON AB L M [SE]; FRENGER PAAL [SE]; BALDEMAIR ROBERT [SE]) 11 November 2010 (2010-11-11)

D2 WO 2013/113988 A1 (NOKIA CORP [FI]; NOKIA SIEMENS NETWORKS OY [FI]) 8 August 2013 (2013-08-08)

D3 US 2014/334390 A1 (LINDHOLM JARI OLAVI [FI] ET AL) 13 November 2014 (2014-11-13)

D4 WO 2014/168574 A1 (ERICSSON TELEFON AB L M [SE]) 16 October 2014 (2014-10-16)

Abbreviations: RS = reference signal

1 With respect to **claim 1,** as the examiner understands the claim with reference to the clarity objection above, document **D1** discloses (any references in parentheses applying to this document) a method, comprising:
a receiving, by a target BS, a request to begin transmitting reference signals for mobility measurements, the request being received from a source BS serving a UE *(page 15 lines 13-17)*; and
b in response to the request, deciding whether or not to transmit the

reference signals for mobility measurements *(page 15 line 20, disclosing that the second BS sends RSs, which is seen as the second BS deciding based on the received request to send RSs)*.

The document **D1** thus discloses **all** features of **claim 1** and the subject-matter of said claim is therefore **not novel** (Articles 52, 54(1) and (2) EPC) with respect to D1, and hence not allowable.

It should furthermore be noted that even if the applicant intended to argue novelty of **claim 1**, based on a slightly different interpretation of the features of said claim and those disclosed in document **D1**, the subject-matter of **claim 1** would still **not involve an inventive step** (Article 52(1) and 56 EPC) having regard to the disclosure of **D1** and the **normal knowledge of a person skilled in the art** of wireless communication, considering that **D1** is directed to achieving the same object, i.e. to enable power saving (see page 2 line 26), and provides the same type of solution as present claim 1.

It is to note, that if the applicant considers that claim 1 differs from D1 in terms of the target BS "deciding", then what the "deciding" is based on, should have been defined in the claim 1. Merely "deciding" is considered derivable from D1, as set out above.

2 Independently of the above objection with respect to document D1, **claim 1** does **not** fulfil the requirements of neither **novelty** (Articles 52, 54(1) and (2) EPC) **nor inventive step** (Article 56 EPC) with respect to document **D2,** since said document also discloses the claimed solution, see par. 0049, disclosing that the serving cell may cause the second cell to begin transmitting the cell specific reference signal.

3 The above inventive step objection regarding claim 1 applies also to independent **claim 10**, which defines a method for transmitting a request, which is inter-related to the method for receiving the defined in independent claim 1. As a consequence the subject-matter of **said claims 1 and 10** is also **neither novel** (Article 54(1) and (2) EPC) **nor does it involve an inventive step** (Article 56 EPC).

4 The above objection regarding claim 1 applies also to independent **claims 14 and 15**, which define an apparatus and computer program, respectively, for performing the method defined in independent claims 1 or 10. As a consequence the subject-matter of **said claims** is also **neither novel** (Article 54(1) and (2) EPC) **nor does it involve an inventive step** (Article 56 EPC).

5 The **dependent claims 2-9, 11-13** do, when interpreted with the aid of the description, **not** contain any additional features which, in combination with the features of any claim to which they refer, **involve an inventive step** (Article 56 EPC) for the reason that the subject-matter of said claims **either** is in principle directly derivable from the disclosure of document **D1**, **D2, D3, D4 or** represents minor design details which are based on the normal knowledge of a person skilled in the art of wireless communication.

Particularly, the additional subject-matter of claim:

- 2, handover of the user equipment is impending, is obvious from D1 or D2;

- 3, 11, RS density or periodicity, is known from D3, par. 0033-0034 or from D4, page 5 lines 11-18;

- 4, determining a candidate beam set to enable efficient discovery of the target base station, provides no unexpected effect since it is well known in the art to send RS on selected beams, and is thus a mere design choice by the skilled person starting from D1;

- 5, increasing RS density or periodicity for service flows serving high reliability traffic, provides no unexpected effect since it is well known in the art that increasing the density results in better performance, and is thus a mere design choice by the skilled person starting from D1;

- 6, 7, changing RS signal density or periodicity depending on mobility state, is known from D4, page 5 lines 11-18;

- 8, 12, receiving a notification of an impending handover of the user equipment and determining a candidate beam set, provides no unexpected effect since it is well known in the art to send RS on selected beams, and is thus a mere design choice by the skilled person starting from D1;

- 9, 13, sending a non-acknowledgement message rejecting the request from the source base station, is an obvious measure by the skilled person starting from D1.

6 The following points regard defects in form or content.

6.1 The **independent claims** are not in **two-part form** in accordance with Rule 43(1) EPC, which in the present case would be appropriate, with those features known in combination from the pre-published prior art D1 being placed in the preamble and with the remaining features being included in the characterising part. The independent claims should therefore be redrafted.

It is to note that non-compliance with Rule 43(1) EPC should be well-founded and can only be permitted when it is clear from the **description** which features defined by the independent claim are already known in combination from the prior art, see EPO Guidelines F-IV, 2.3.2.

6.2 To increase their intelligibility, all the claims should include **reference signs in parentheses** where features shown in the **drawings** are referred to, Rule 43(7) EPC. This applies to both the preamble and characterising portion, EPO Guidelines F-IV, 4.11.

6.3 In order to meet the requirements of Rule 42(1)(b) EPC, the **relevant prior art** disclosed in document D1 should have been acknowledged by reference and briefly discussed in the introductory part of the **description** (EPO Guidelines F-II, 4.3), preferably in such a way that the **inventive merit** of what is claimed can be readily understood.

6.4 It is to note that **no** amendments filed during the procedure may **extend beyond** the content of the **application as filed**, Article 123(2) EPC.

Further, any **amendments** filed during the procedure should be **clearly identified** and the passages of the **application as filed** on which these amendments are based should be **indicated** (Rule 137(4) EPC).

# APPENDIX B: GAUSSIAN NAÏVE BAYES RESULTS

| Model name | Features used (% of total) | Accuracy |
|---|---|---|
| Count Vectorizer | 18815 Unigrams (100) | 88.33 |
| | 14210 Unigrams (75) | 88.89 |
| | 9446 Unigrams (50) | 88.61 |
| | 4642 Unigrams (25) | 88.94 |
| | 2783 Unigrams (15) | 88.47 |
| | 1870 Unigrams (10) | 88.04 |
| | 939 Unigrams (5) | 88.20 |
| | 26541 Bigrams (10) | 87.86 |
| | 13065 Bigrams (5) | 89.47 |
| | 5277 Bigrams (2) | 90.99 |
| | 23060 Trigrams (5) | 91.96 |
| | 9208 Trigrams (2) | 92.39 |
| | 4613 Trigrams (1) | 93.17 |
| | 26968 4-grams (5) | 92.76 |
| | 16323 4-grams (3) | 92.06 |
| | 5411 4-grams (1) | 93.14 |
| | 25178 bi & trigrams (5) | 89.01 |
| | 16323 bi & trigrams (3) | 91.11 |
| | 5411 bi & trigrams (1) | 91.82 |
| TF-IDF vectorizer | 18681 Unigrams (100) | 86.95 |
| | 14159 Unigrams (75) | 87.47 |
| | 9378 Unigrams (50) | 87.75 |
| | 4730 Unigrams (25) | 87.68 |
| | 2840 Unigrams (15) | 87.47 |
| | 1858 Unigrams (10) | 87.78 |
| | 948 Unigrams (5) | 86.95 |
| | 25306 Bigrams (15) | 87.09 |
| | 17248 Bigrams (10) | 87.54 |
| | 8385 Bigrams (2) | 86.39 |
| | 23255 Trigrams (7) | 86.76 |
| | 16933 Trigrams (5) | 86.16 |
| | 10142 Trigrams (3) | 86.11 |
| | 6715 Trigrams (2) | 86.37 |
| | 29411 4-grams (7) | 86.99 |
| | 20899 4-grams (5) | 86.14 |
| | 12408 4-grams (3) | 86.01 |
| | 4243 4-grams (1) | 85.71 |
| | 25418 bi & trigrams (5) | 87.05 |
| | 15242 bi & trigrams (3) | 86.08 |
| | 5057 bi & trigrams (1) | 85.62 |

# APPENDIX C: PART OF XML OUTPUT FROM OA

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title></title>
  <meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
  <meta name='ocr-system' content='tesseract v5.0.0-alpha.20191030' />
  <meta name='ocr-capabilities' content='ocr_page ocr_carea ocr_par ocr_line ocrx_word ocrp_wconf'/>
 </head>
 <body>
  <div class='ocr_page' id='page_1' title='image "C:\Users\GHOST\AppData\Local\Temp\tess_uey52aov.PNG"; bbox 0 0
   <div class='ocr_carea' id='block_1_1' title="bbox 232 112 1233 148">
    <p class='ocr_par' id='par_1_1' lang='eng' title="bbox 232 112 1233 148">
     <span class='ocr_line' id='line_1_1' title="bbox 232 112 1233 148; baseline 0 -16; x_size 21.279999; x_desce
      <span class='ocrx_word' id='word_1_1' title='bbox 232 128 294 144; x_wconf 96'>Datum</span>
      <span class='ocrx_word' id='word_1_2' title='bbox 732 112 775 148; x_wconf 96'>Blatt</span>
      <span class='ocrx_word' id='word_1_3' title='bbox 1110 112 1233 142; x_wconf 87'>Anmelde-Nr:</span>
     </span>
    </p>
   </div>
   <div class='ocr_carea' id='block_1_2' title="bbox 232 147 1511 340">
    <p class='ocr_par' id='par_1_2' lang='eng' title="bbox 232 147 1511 340">
     <span class='ocr_line' id='line_1_2' title="bbox 232 147 1511 190; baseline 0 -15; x_size 35.054966; x_desce
      <span class='ocrx_word' id='word_1_4' title='bbox 232 159 276 175; x_wconf 86'>Dae</span>
      <span class='ocrx_word' id='word_1_5' title='bbox 338 155 374 176; x_wconf 87'>cf</span>
      <span class='ocrx_word' id='word_1_6' title='bbox 397 156 476 176; x_wconf 95'>Form</span>
      <span class='ocrx_word' id='word_1_7' title='bbox 500 154 572 176; x_wconf 95'>1507</span>
      <span class='ocrx_word' id='word_1_8' title='bbox 731 159 788 175; x_wconf 80'>Sheet</span>
      <span class='ocrx_word' id='word_1_9' title='bbox 850 155 863 175; x_wconf 74'>1</span>
      <span class='ocrx_word' id='word_1_10' title='bbox 1110 147 1217 190; x_wconf 0'>npplication</span>
      <span class='ocrx_word' id='word_1_11' title='bbox 1278 155 1310 175; x_wconf 60'>17</span>
      <span class='ocrx_word' id='word_1_12' title='bbox 1337 154 1389 175; x_wconf 73'>172</span>
      <span class='ocrx_word' id='word_1_13' title='bbox 1416 154 1511 176; x_wconf 95'>763.9</span>
     </span>
     <span class='ocr_line' id='line_1_3' title="bbox 232 184 1238 222; baseline 0 -15; x_size 25; x_descenders 6
      <span class='ocrx_word' id='word_1_14' title='bbox 232 184 276 217; x_wconf 96'>Date</span>
      <span class='ocrx_word' id='word_1_15' title='bbox 732 184 795 217; x_wconf 93'>Feuille</span>
      <span class='ocrx_word' id='word_1_16' title='bbox 1111 205 1119 222; x_wconf 0'>P</span>
      <span class='ocrx_word' id='word_1_17' title='bbox 1140 188 1142 190; x_wconf 0'>:</span>
      <span class='ocrx_word' id='word_1_18' title='bbox 1225 206 1238 213; x_wconf 24'>°.</span>
     </span>
     <span class='ocr_line' id='line_1_4' title="bbox 242 310 1328 340; baseline 0.001 -6; x_size 26; x_descender
      <span class='ocrx_word' id='word_1_19' title='bbox 242 310 250 334; x_wconf 96'>1</span>
      <span class='ocrx_word' id='word_1_20' title='bbox 358 314 403 335; x_wconf 96'>The</span>
      <span class='ocrx_word' id='word_1_21' title='bbox 414 314 560 335; x_wconf 96'>examination</span>
      <span class='ocrx_word' id='word_1_22' title='bbox 572 314 589 335; x_wconf 96'>is</span>
      <span class='ocrx_word' id='word_1_23' title='bbox 599 314 665 340; x_wconf 96'>being</span>
      <span class='ocrx_word' id='word_1_24' title='bbox 674 314 757 335; x_wconf 96'>carried</span>
      <span class='ocrx_word' id='word_1_25' title='bbox 766 315 805 335; x_wconf 96'>out</span>
      <span class='ocrx_word' id='word_1_26' title='bbox 813 318 841 335; x_wconf 96'>on</span>
      <span class='ocrx_word' id='word_1_27' title='bbox 852 314 889 335; x_wconf 96'>the</span>
      <span class='ocrx_word' id='word_1_28' title='bbox 897 314 1017 340; x_wconf 96'>following</span>
```

```
</div>
<div class='ocr_carea' id='block_1_3' title="bbox 240 416 487 443">
 <p class='ocr_par' id='par_1_3' lang='eng' title="bbox 240 416 487 443">
  <span class='ocr_line' id='line_1_5' title="bbox 240 416 487 443; baseline 0 -6; x_size 27; x_descenders 6; x_ascenders
   <span class='ocrx_word' id='word_1_31' title='bbox 240 416 397 443; x_wconf 96'>Description,</span>
   <span class='ocrx_word' id='word_1_32' title='bbox 408 416 487 443; x_wconf 96'>Pages</span>
  </span>
 </p>
</div>
<div class='ocr_carea' id='block_1_4' title="bbox 258 479 713 505">
 <p class='ocr_par' id='par_1_4' lang='eng' title="bbox 258 479 713 505">
  <span class='ocr_line' id='line_1_6' title="bbox 258 479 713 505; baseline 0 -6; x_size 26; x_descenders 6; x_ascenders
   <span class='ocrx_word' id='word_1_33' title='bbox 258 480 309 500; x_wconf 96'>1-29</span>
   <span class='ocrx_word' id='word_1_34' title='bbox 510 484 537 500; x_wconf 96'>as</span>
   <span class='ocrx_word' id='word_1_35' title='bbox 546 479 655 505; x_wconf 94'>originally</span>
   <span class='ocrx_word' id='word_1_36' title='bbox 664 479 713 500; x_wconf 96'>filed</span>
  </span>
 </p>
</div>
<div class='ocr_carea' id='block_1_5' title="bbox 240 596 465 622">
 <p class='ocr_par' id='par_1_5' lang='eng' title="bbox 240 596 465 622">
  <span class='ocr_line' id='line_1_7' title="bbox 240 596 465 622; baseline 0.004 -5; x_size 25; x_descenders 4; x_ascer
   <span class='ocrx_word' id='word_1_37' title='bbox 240 596 335 622; x_wconf 96'>Claims,</span>
   <span class='ocrx_word' id='word_1_38' title='bbox 346 597 465 618; x_wconf 96'>Numbers</span>
  </span>
 </p>
</div>
<div class='ocr_carea' id='block_1_6' title="bbox 258 660 713 686">
 <p class='ocr_par' id='par_1_6' lang='eng' title="bbox 258 660 713 686">
  <span class='ocr_line' id='line_1_8' title="bbox 258 660 713 686; baseline 0 -6; x_size 37.25; x_descenders 6.5; x_asce
   <span class='ocrx_word' id='word_1_39' title='bbox 258 660 309 681; x_wconf 94'>1-15</span>
   <span class='ocrx_word' id='word_1_40' title='bbox 510 664 537 681; x_wconf 96'>as</span>
   <span class='ocrx_word' id='word_1_41' title='bbox 546 660 655 686; x_wconf 96'>originally</span>
   <span class='ocrx_word' id='word_1_42' title='bbox 664 660 713 681; x_wconf 95'>filed</span>
  </span>
 </p>
</div>
<div class='ocr_carea' id='block_1_7' title="bbox 240 778 469 805">
 <p class='ocr_par' id='par_1_7' lang='eng' title="bbox 240 778 469 805">
  <span class='ocr_line' id='line_1_9' title="bbox 240 778 469 805; baseline 0.004 -6; x_size 26; x_descenders 4; x_ascer
   <span class='ocrx_word' id='word_1_43' title='bbox 240 779 369 805; x_wconf 96'>Drawings,</span>
   <span class='ocrx_word' id='word_1_44' title='bbox 379 778 469 800; x_wconf 96'>Sheets</span>
  </span>
 </p>
</div>
<div class='ocr_carea' id='block_1_8' title="bbox 258 841 713 867">
 <p class='ocr_par' id='par_1_8' lang='eng' title="bbox 258 841 713 867">
  <span class='ocr_line' id='line_1_10' title="bbox 258 841 713 867; baseline -0.002 -5; x_size 27; x_descenders 6; x_asc
   <span class='ocrx_word' id='word_1_45' title='bbox 258 841 387 862; x_wconf 90'>1/10-10/10</span>
   <span class='ocrx_word' id='word_1_46' title='bbox 510 846 537 862; x_wconf 96'>as</span>
   <span class='ocrx_word' id='word_1_47' title='bbox 546 841 655 867; x_wconf 94'>originally</span>
   <span class='ocrx_word' id='word_1_48' title='bbox 664 841 713 862; x_wconf 96'>filed</span>
  </span>
```

```
  <span class='ocr_line' id='line_1_11' title="bbox 240 978 1504 1011; baseline 0 -7; x_size 32; x_descenders 7; x_ascen
   <span class='ocrx_word' id='word_1_49' title='bbox 240 980 255 1003; x_wconf 96'>2</span>
   <span class='ocrx_word' id='word_1_50' title='bbox 360 978 510 1004; x_wconf 96'>Reference</span>
   <span class='ocrx_word' id='word_1_51' title='bbox 522 979 543 1004; x_wconf 96'>is</span>
   <span class='ocrx_word' id='word_1_52' title='bbox 556 979 635 1004; x_wconf 96'>made</span>
   <span class='ocrx_word' id='word_1_53' title='bbox 646 981 672 1004; x_wconf 96'>to</span>
   <span class='ocrx_word' id='word_1_54' title='bbox 684 979 728 1004; x_wconf 97'>the</span>
   <span class='ocrx_word' id='word_1_55' title='bbox 739 978 867 1011; x_wconf 96'>following</span>
   <span class='ocrx_word' id='word_1_56' title='bbox 878 979 1048 1008; x_wconf 96'>documents;</span>
   <span class='ocrx_word' id='word_1_57' title='bbox 1060 979 1105 1004; x_wconf 95'>the</span>
   <span class='ocrx_word' id='word_1_58' title='bbox 1117 979 1271 1011; x_wconf 96'>numbering</span>
   <span class='ocrx_word' id='word_1_59' title='bbox 1282 979 1325 1003; x_wconf 96'>will</span>
   <span class='ocrx_word' id='word_1_60' title='bbox 1339 979 1373 1004; x_wconf 96'>be</span>
   <span class='ocrx_word' id='word_1_61' title='bbox 1384 979 1504 1004; x_wconf 96'>adhered</span>
  </span>
  <span class='ocr_line' id='line_1_12' title="bbox 358 1024 800 1057; baseline 0 -7; x_size 32; x_descenders 7; x_ascen
   <span class='ocrx_word' id='word_1_62' title='bbox 358 1027 383 1050; x_wconf 96'>to</span>
   <span class='ocrx_word' id='word_1_63' title='bbox 396 1025 417 1049; x_wconf 96'>in</span>
   <span class='ocrx_word' id='word_1_64' title='bbox 430 1025 475 1050; x_wconf 96'>the</span>
   <span class='ocrx_word' id='word_1_65' title='bbox 487 1027 540 1050; x_wconf 96'>rest</span>
   <span class='ocrx_word' id='word_1_66' title='bbox 551 1024 577 1050; x_wconf 96'>of</span>
   <span class='ocrx_word' id='word_1_67' title='bbox 587 1025 632 1050; x_wconf 96'>the</span>
   <span class='ocrx_word' id='word_1_68' title='bbox 644 1025 800 1057; x_wconf 96'>procedure:</span>
  </span>
 </p>
</div>
<div class='ocr_carea' id='block_1_10' title="bbox 370 1086 1449 1158">
 <p class='ocr_par' id='par_1_10' lang='eng' title="bbox 370 1086 1449 1158">
  <span class='ocr_line' id='line_1_13' title="bbox 370 1086 1449 1119; baseline 0 -8; x_size 31.5; x_descenders 5.5; x_
   <span class='ocrx_word' id='word_1_69' title='bbox 370 1087 408 1111; x_wconf 26'>D1</span>
   <span class='ocrx_word' id='word_1_70' title='bbox 457 1086 500 1112; x_wconf 20'>=US</span>
   <span class='ocrx_word' id='word_1_71' title='bbox 510 1087 751 1112; x_wconf 88'>2011/283223</span>
   <span class='ocrx_word' id='word_1_72' title='bbox 721 1082 756 1123; x_wconf 81'>A1</span>
   <span class='ocrx_word' id='word_1_73' title='bbox 766 1087 950 1119; x_wconf 90'>(VAITTINEN</span>
   <span class='ocrx_word' id='word_1_74' title='bbox 960 1086 1079 1112; x_wconf 91'>SAMPO</span>
   <span class='ocrx_word' id='word_1_75' title='bbox 1092 1086 1230 1112; x_wconf 96'>TUOMAS</span>
   <span class='ocrx_word' id='word_1_76' title='bbox 1239 1087 1289 1119; x_wconf 74'>[F1]</span>
   <span class='ocrx_word' id='word_1_77' title='bbox 1309 1087 1340 1111; x_wconf 96'>ET</span>
   <span class='ocrx_word' id='word_1_78' title='bbox 1348 1087 1399 1119; x_wconf 95'>AL)</span>
   <span class='ocrx_word' id='word_1_79' title='bbox 1415 1088 1449 1111; x_wconf 96'>17</span>
  </span>
  <span class='ocr_line' id='line_1_14' title="bbox 455 1133 688 1158; baseline -0.004 0; x_size 30.351351; x_descenders
   <span class='ocrx_word' id='word_1_80' title='bbox 455 1133 610 1158; x_wconf 96'>November</span>
   <span class='ocrx_word' id='word_1_81' title='bbox 617 1134 688 1158; x_wconf 96'>2011</span>
  </span>
 </p>
</div>
<div class='ocr_carea' id='block_1_11' title="bbox 370 1212 1303 1244">
 <p class='ocr_par' id='par_1_11' lang='eng' title="bbox 370 1212 1303 1244">
  <span class='ocr_line' id='line_1_15' title="bbox 370 1212 1303 1244; baseline 0 -7; x_size 45.75; x_descenders 7.5; x
   <span class='ocrx_word' id='word_1_82' title='bbox 370 1212 408 1237; x_wconf 84'>D1</span>
   <span class='ocrx_word' id='word_1_83' title='bbox 457 1212 705 1237; x_wconf 24'>US2007/073475</span>
   <span class='ocrx_word' id='word_1_84' title='bbox 713 1212 751 1237; x_wconf 90'>A1</span>
```