AALBORG UNIVERSITET
ESBJERG

MASTER THESIS IN SOFTWARE CONSTRUCTION

AUTOMATIC E-MAIL CATEGORIZATION

Author:

Damian Bukszynski

Dbuksz11@student.aau.dk

Supervisor:

Dr. Daniel Ortiz-Arroyo

do@cs.aaue.dk

October 1, 2013

SUMMARY

Nowadays e-mails became the most important medium between individuals but also companies and various organizations and they settled down closely in almost any aspect of our everyday activity. E-mails are not just simple text information, they can also transport different kind of attachments. They can be archived and form a powerful, non-volatile source of knowledge and in some cases they can even constitute clear evidences in trials.

Maintaining mailboxes in a structured form is a challenging task. When incoming and outgoing correspondence have a low rate the task is relatively easy but as the rate is increasing the problem is getting more and more complicated and its solutions more and more time consuming. This process may be improved in a few ways. Most mailboxes allow for some helper options as Journaling to address and automate it at a basic level. However to achieve a really good organization level it is necessary to search for external tools such as machine learning methods.

Four machine learning algorithms have been implemented and their performance examined in this project. Also two additional methods based on combination of the results from the single classifiers have been implemented. Eventually all methods have been compared and the one which gives the best improvement to the e-mail classification process has been chosen.

This master thesis uses a part of the Enron e-mail collection [1] for training and testing phase. The best result achieved combination of single classifiers with F-measure equal to 0.7102.

The topics elaborated in the thesis, both the text and the software part, offer to the reader great knowledge about Information Retrieval, Machine Learning and related topics.

PREFACE

This MSc master thesis is the result of the last semester of my education in the Aalborg University Esbjerg at the department of Software Construction. The adventure with Information Retrieval started during the fourth semester of my education at the University and encouraged me to increase my understanding of machine learning algorithms which admittedly can find applications not only in e-mails but also in other fields of data processing.

The attached CD-ROM contains this report in electronic version, the Visual Studio Installer and full source code of the e-mail categorization program.

I would like to express my gratitude to Dr. Daniel Ortiz-Arroyo for his excellent advices and cooperation throughout this project. His wisdom and attitude showed me the right way how to keep my work organized and effective and eventually made the thesis saw the light of day .

Also I would like to thank my family, my mother Elżbieta, my father Zbigniew and my beloved sister Daria, for having given me a chance to study at the University and for their exceptional support and constant love. I have never felt alone. This work is dedicated to them.

CONTENTS

LIST OF FIGURES

CHAPTER 1 - INTRODUCTION

Since the early nineties of the last century, along with the popularization of the Internet as a widely accessible medium in which millions of people around the world can publish information without virtually limits, a growing interest in techniques for effective collecting and processing of information has been observing. The increasing number of documents (according to [2] there are several billion of them available in the network) and lack of well-defined data model make the task very difficult and time consuming, even using the latest state-of art computer technology.

The same demand applies to e-mails. In most cases mailboxes can be configured to meet certain needs but a human being is still the most important agent and a lot of tasks have to be performed manually. If the number of sent and received messages is small the activities do not take much time but with an increasing number of messages they become more and more tiring and inefficient.

This phenomenon is mostly observable in companies where the number of messages is usually huge. Two different solutions are commonly in use: the mentioned manual performance which requires skilled staff with necessary knowledge - this approach is called the expert knowledge, and deployment of algorithms belonging to the machine learning class.

The usage of techniques drawn from the machine learning and Information Retrieval algorithms attracts with the possibility of the process automation which in turn results in a significant time reduction and possible avoidance of human intervention. Currently, there are a lot of algorithms available which greatly simplify the task.

Communication through e-mails offer extensive flexibility of their application. E-mails are used in almost all areas of our life starting from regular activity at work, through shopping, advertising, logging in to various websites and services and ending with a private correspondence, just to give a few examples. With such a broad range of application it is quite often that mailboxes fill up very quickly, not only relevant messages but also with unimportant and often unwanted ones spam.

The method of dealing with the inboxes and their organization is discussed and implemented in the thesis. It can considerably simplify handling of e-mails basically by their classification according to one's preferences and then dispatching to appropriate folders.

The classification of e-mails is a hierarchical system of categories used to organize messages according to their content, so that any e-mail can be easy found. The following example illustrates a typical hierarchical structure of a mailbox:

- Mailbox
    - Category
        - Message

The hierarchical organization of the mailbox is commonly known to most people using it. In a properly prepared mailbox, people can easily navigate to find a desired message. However, the preparation has to be done manually by the mailbox's owner.

Sorting messages has always been a desirable phenomenon. Even before the computer era people used special compartments to sort the letters according to the sender, date, subject, or other specific features. With e-mail boxes the practise is very much similar, proper maintenance of the boxes and keeping e-mails in order facilitates working with the them.

Today there are many companies on the market providing mailbox solutions. The most popular ones are Gmail with, according to [3], 235 million users (data from June 2012) and Yahoo with, according to [4], 281 million users (data from December 2012). The number of users of these two worldwide companies is very impressive. Also there are many local, country level service providers.

E-mails addresses are composed of three fields put together according to the rules. The first field is an user identifier which can be selected at its own discretion, it is followed by the @ sign, and then by a domain name. An example e-mail address is provided below:

- John.Smith@example.com

E-mails address allow users to easily select the persons to whom the message should be delivered. Today it is common for a single user to possess a number of e-mail address. Mostly it is because they want to distinguish their company addresses from their privet ones. A lot of software developers want to address the need and offer full environmental management systems. The most known one is Microsoft Outlook that allows for efficient e-mail addresses management. In addition to the main task the system is fully integrated with a calendar or a task list.

Despite these amenities segregation of messages in the first phase has to be done manually. Users have to create their own folder structure, which may be extended without limits. However, over the time, they enrol in a number of services, and establish new contacts what makes, the management work very elongate. The result is that they lose a lot of time. Taking advantage of the methods of the Machine Learning family, the process can be simplified to minimum.

1.1. Objectives

The main objective of the thesis is to propose a solution to the problem of automatic classification of incoming e-mails. The task is very complex and broad.

a) Implementation of machine learning methods for email classification
b) Examination of the methods
c) Results comparison

1.2. Report organization

The thesis are composed of eight chapters which present theoretical and practical aspects of the subject. Chapter 2 presents issues related to classification of text in a variety of contexts, from the general definition of the classification problem to e-mail categorization. Chapter 3 deals with theoretical issues related to machine learning, and selected methods of this family. Information about popular methods of combining classifiers are presented in Chapter 4. Chapter 5 describes available literature on classification methods. Chapter 6 describes the email categorization

system model that has been chosen to be implemented in the work. Chapter 7 presents the experimental setup and discusses the results obtained. The final Chapter 8 summarizes the work and outlines possible further extensions to the current work.

CHAPTER 2 – TEXT CLASSIFICATION PROBLEM

There are many algorithms and methods in the field of the text categorization (TC). The categorization process is divided into different phases and each of them can be implemented with different methods. Learning is an inductive process which produces a classifier. The classifier learns from training data. The TC is based on categorizing documents into some predefined categories.

Training the classifier must be made on the basis of the structure of the documents prepared intellectually. This structure is called the document collection. The collections are organized into categories and each category contains a set of labelled documents. The collections in most cases consist of two sets: a training set and a test set. The purpose of the training set is to learn the classifier. The learning is based on the relevant behaviour among examples. The test set is used to verify the quality of the classifier.

2.1. Categorization of e-mails

The mailbox is a simple tool used basically for sending and receiving e-mails. If it has not been customized according to the user preferences all incoming messages go to the "Received" folder. The following steps have to be performed by the user to categorize them manually:

- Read the received message
- Decide what folder the message should be moved to
- If a proper pre-defined folder does not exist, create it
- Move the message to the folder

The procedure has to be repeated for each new message received. The task can be simplified by using specific mailbox's settings. If the user set his own preferences e-mails will be moved to proper folders automatically. However the decision is still made on the basis of the e-mail address. When a new e-mail is not recognized it goes to the "Received" folder.

2.2. Automatic e-mail categorization

Personalized mailbox gives a great opportunity for its application in data mining, information retrieval and machine learning. Of course a lot of effort has to be put as all messages have to be read before any further decision can be made.

Automatic categorization of documents may be implemented by means of intellect based taxonomy or algorithms of the machine learning family. Intellect based taxonomy is necessary to be used if a proper folder for new e-mails is not prepared or it is empty. For training to take effect, the classifiers have to be provided with data.

Machine Learning algorithms learn from examples called the training documents. These examples must be assigned to correct categories. Each category must be assigned its own examples of similar content to make distinguishing among the possible categories. The learning process is carried out using a feature called classifier. The classifier is able to classify previously unknown documents to one or more categories. Learning the classifier is performed in the inductive process called learning/training.

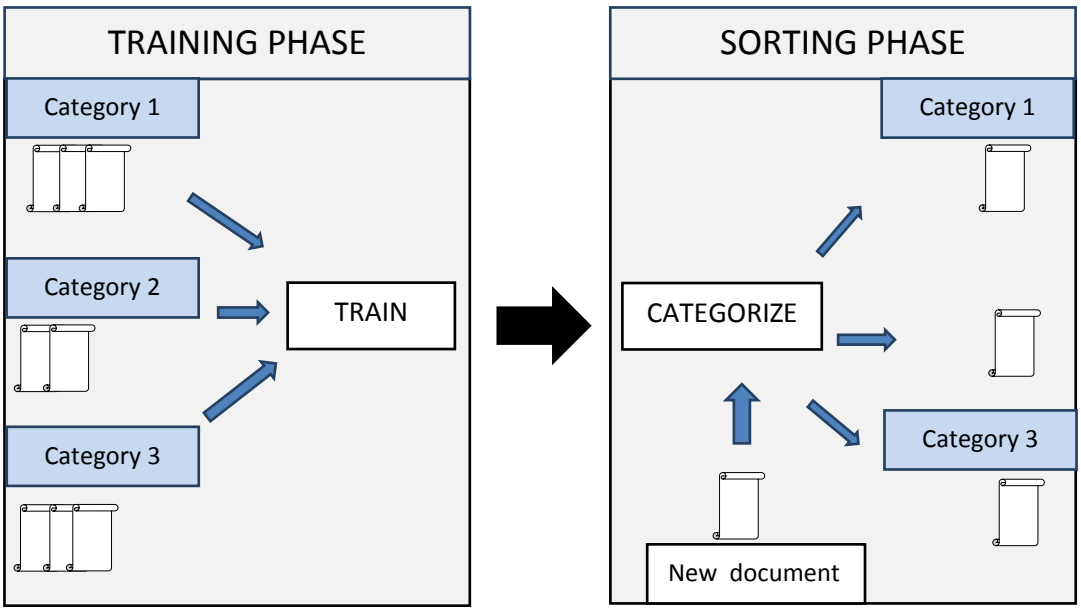

Figure 1 Training-sorting phase of categorization

The illustration above presents steps which have to be carried out in order to perform the categorization. The first one is to get data collections prepared in a form of indexed words, which constitute the input data. The next step is to learn the classifier. To achieve it a machine learning algorithm or a combination of weights taken from previously conducted training sessions can be used. The final step is to assign the new document to an appropriate category.

Machine learning techniques are aimed not only to problems associated with e-mails classification. Studies on the text categorization cover many areas including categorization of newspapers, websites and many other.

2.3. Data Mining

The data generated in an electronic form are growing at a very fast pace. According to[5], data produced double every 20 months. Although smaller than in the case of movies, websites, or blogs the phenomenon is also noticeable in the case of e-mail messages. However, all these cases share one common principle: with the increasing volume of data the corresponding problems of their organization increases proportionately. The data are mostly stored in databases, where they can be easily extracted from using queries. In many cases, the queries ensure adequate control over the data, but for more complex applications they are not sufficient. According to [5] "... data mining is the extraction of implicit, previously unknown, and potentially useful information from data". In other words, data mining is the extraction of relevant data from the database to find appropriate models that can have a positive impact on further data processing to eventually achieve accurate results. Discovered patterns do not necessarily lead to the process improvement. The patterns may be inaccurate or simply already known to the expert. Another issue is that the data mining algorithms should be able to deal with incomplete data collections.

Recently, a huge increase of interest in the field of Machine Learning (ML) can be observed [6]. This has resulted in a number of different algorithms which are used in many applications based on the pattern recognition. Variety of algorithms and their variations significantly affect research conducted by different groups.

To achieve its objectives the Machine Learning, in most cases, uses two sets of data: the first one is used for learning and is commonly referred to as the training data, The second one is used for testing the found pattern and is commonly called the test set. Machine learning in its assumptions can be divided into two distinct techniques: learning with a teacher called supervised learning and learning without a teacher called unsupervised learning.

Page [8] shows that "the aim of the supervised learning is to learn a mapping from the input to an output whose correct values are provided by a supervisor." In other words, the supervised learning involves finding an appropriate model based on the inputs and unsupervised learning based on the data without prior preparation.

Kononenko, Bratko and Kukar in [7] divide the machine learning algorithms into three main groups:

- Artificial Neural Networks
- Pattern recognition
- Inductive learning

Of course, in the scope of each group, there are various methods that can be used in different fields such as: text classification, healthcare and medical data analysis, business, information retrieval from database [6], optical character recognition(OCR), image and video recognition or even games, just to list a few most popular ones.

2.4. Variants of Data Mining

According to [9] the area of data mining applications became vast and the methods can be applied to different kind of documents. The approach to organizing documents uses classification, clustering, association and sequence discovery. These methods employed against documents with similar content can greatly simplify and speed up the process.

2.4.1. Categorization

The categorization is a method based on the idea of learning with the teacher and involves assigning documents (or parts thereof) to one of the predefined categories. These categories are usually static ones (also called classes of the decision class) which means that they are defined at the beginning based on the analysis of the contents of several documents - in other words, these are the types of documents that we have. New texts received are classified into one of these predefined categories.

To perform the classification process it is necessary to use a reference collection of human prepared documents. Usually the document collection is divided into two classes: the training one and the test one. Both can contain documents divided further into subcategories.

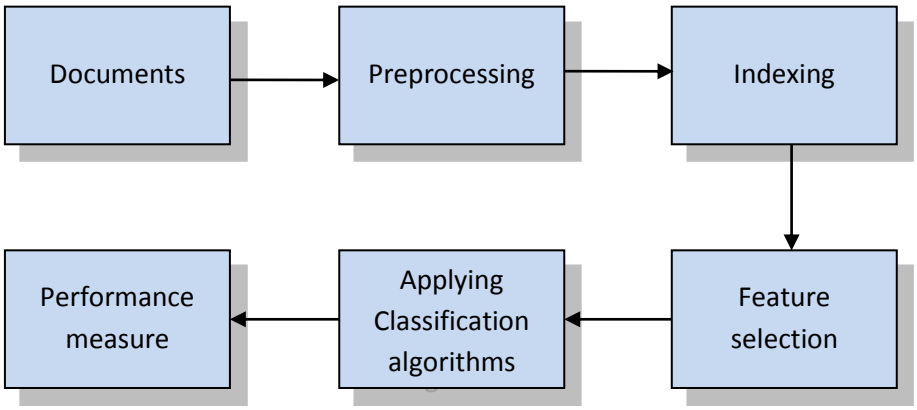

Figure 2 Steps of classifying documents

2.4.2. Clustering

The clustering is a method representing a process of learning without the teacher (unsupervised learning). Though it uses neither a predefined category structure nor a training set it can, show the expected relationship between documents. However this method requires identification of some measures, description of grouped objects, and definition of a threshold value, which would determine the degree of similarity between these objects. The purpose of the grouping is to define classes of objects (clusters). Object within a class should be, as much as possible, similar to each other

(should have similar features) and should as much as possible differ from objects of other classes.

2.4.3. Text mining

The text mining is a technique based on the extraction of relevant patterns from text documents. The technique is also known as Knowledge Discovery from Text (KDT). The text mining is based on the methods of data mining and Natural Language Processing (NLP) which enable analysing of text collections. The KDT offers a wider range of functionality than just finding information through word processing, mining knowledge and understanding of individual documents. Text Mining applications applied to digital text data, allow for the patterns and relationships identification and results visualization.

2.4.4. Unstructured data sources.

Data for analysing the Text Mining algorithms can origin from various external and/or internal sources. Very valuable source of external data are social services with thousands of posts, comments, feedback, etc. Minutes from conversations with customers, e-mails, business documents such as contracts and offers, publications, transcripts of call-centre, descriptions of insurance claims, police notes, open-ended questions in surveys, etc. are examples of internal sources of data.

2.4.5. Association

The *association* is a method based mainly on finding the right relationship between records in the database. The main areas of its application are healthcare and the NLP based marketing. Besides, the method has found its applications images comparison or online shopping - for example, if someone bought a product x what is the probability that he/she will purchase the products y.

2.5. Document collection

Categories have to be predefined before the classification process can be applied. There are many documents sets with different contents available on the Internet, for example WIPO-Alpha [10] published the data collection related to the patent categorization. The categorization is related not only to text. It can be related to video, stock markets, health care etc. Example data collections can be downloaded from the Internet. Alternatively data collections can be created on our own. This approach however may be a tough task if the data set is going to be big. In such case, the categories should be prepared first and then relevant documents put inside.

In majority of cases the document collections are divided into two sets:

- training set
- test set

The training set is used to construct a classifier. The test set is prepared to evaluate the classifier. Size of the data sets is one of the issues related to the process preparation. Authors in [11] strongly recommend splitting these two sets in proportion 2/3 for the training set and 1/3 for the test set. In some cases the classifier can be overloaded e.g. trained too much. Obviously in such case such the system will work but the trained function will not be able to recognize documents which are not very similar to the ones of the training set. For this reason it is necessary to have a function or functions which would be able to determine if the classifier was trained correctly.

During training the classifier it is impossible to predict when process should be finished. It may lead to complicated situations. If the classifier is undertrained or over-trained it may give wrong results. Using another document collection working as a validation set prevents such situation.

To make the learning algorithms brought satisfactory results, the training set should incorporate as many documents as possible. In such cases the learning process slows down but the learned hypotheses usually have better accuracy.

The training set should be composed of a large number of examples. The collections of documents used in the thesis are divided into three sets, the first one contains 417 a second 4104 and the third 8241 e-mails. Each set is divided into 25 categories. The classifiers quality grows proportionally with the size of the collection. The best results have been achieved by using a combination of classifiers and large collection of documents and the best F-measure achieved is equal to 0.7102 .

2.6. Enron email collection

It is possible to find on the Internet a lot of different kind of text document collections. However, as the objective of the thesis is related to email categorization, I choose the Enron dataset collection to work with.

Enron was a company founded in 1985. In 2001 the company fell down because of the financial scandal. For years its e-mail collections have been used as evidences in the court. Then, these messages have been released to the public. That big data sets have been used as an excellent material for researchers and scientists in the fields of natural text processing, information extraction and machine learning. Originally this collection was used by William Cohen [4]. It contained 517.431 messages belonging to 150 users, mostly senior management.

The table below illustrates an excerpt from the original Enron database which were used in the thesis. Each collection contains 25 categories. Each employee has been assigned from one to six categories. There are numbers of e-mails belonging to each category depends on the collection. The summary shows that the collections used to carried out my research contained from 417 to 8241 of emails.

TRAINING SET		
Label	Number of category	Number of E-mails
Collection1	25	417
Collection2	25	4104
Collection3	25	8241

Figure 3 Training set used in the work

The graph below shows the relationship between the number of e-mails in the training set and the number of categories.

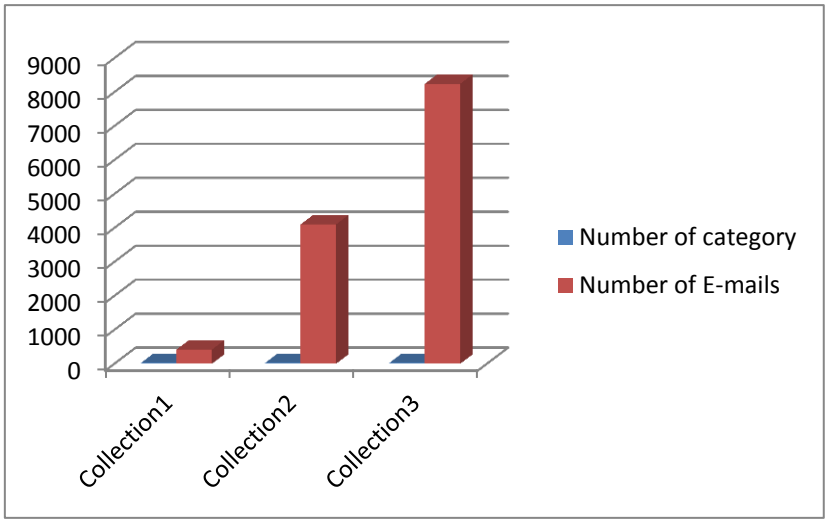

Figure 4 Relationship between categories and number of emails

The collection of documents includes also the test set, which has the same structure as the training set and differs in number of e-mails only. The test set is composed of 168 e-mails. Chapter 7, Experimental setup and results, reports the precision, recall and F1-measures which are calculated by the developed software and discusses results.

TEST SET

Label	Number of category	Number of E-mails
Back	6	37
Farmer	4	17
Kaminski	6	34
Lokay	4	31
Sanders	4	37
Williams	1	11
		168

Figure 5 Test set used in the work

Figure 7 presents the relationship between the amount of emails in the test set and the training set. The same structure allows to test the quality of learned hypothesis during learning/testing phase.

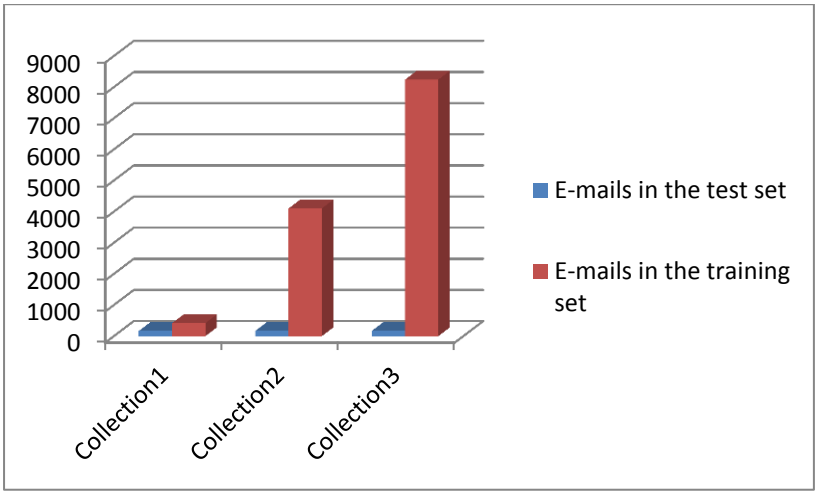

Figure 6 Relationship between categories and number of emails

2.7. Document representation

Analysis and manipulation of natural language documents requires transformation of text data into a comprehensible for the computer numerical form which can be further processed using standard data mining algorithm. There are no strict rules defining which features should be taken into account during transformation. Below there is a proposal how to accomplish it:

- feature extraction
- feature selection
- selection of the document representation (a logical view of the document)

2.7.1. Feature extraction

The first step of preparing data for the categorization process is called pre-processing. Its purpose is to generate a list of keywords (terms), which sufficiently describe the documents to be analysed. This list is called the dictionary.

First, a set of the training documents are passed to the parsing process which eliminates, according to the requirements, unwanted characters such as whitespace characters, digits etc. Then the documents are transformed to a set of keywords, the semantics is considered. These operations usually remove stop words (articles,

prepositions etc. with little or none influence on the decision process) and stem the words. The most common algorithm is Porter algorithm. Another, well-known, is Snow-ball algorithm which might be used in variety of languages. The exact course of this process depends on the language in which the document has been written.

2.7.2. Feature selection

The previous step of the preparation phase was designed to eliminate irrelevant keywords i.e. words which carry little on none information about a category to which the processed documents should belong to. Next, the elaborated list of words has to be converted to the numerical values. The following methods can be applied to accomplish the task:

1. Term Frequency (TF) - frequency of a keyword in the collection of documents
2. Term frequency-inverse document frequency (TF-IDF)

Words with the highest index scored are meant to be important. The whole list of words is passed to the next processing phase which is a classification algorithm.

Term Frequency is a number of occurrences of a word in a single category and it is a component for calculating a more complex measure which is TF-IDF.

The latter is determined as the logarithm of the ratio of the total number of documents and the number of documents containing the term.

To avoid dividing by zero the denominator is usually adjusted by adding 1 to it.

2.7.3. Recall, precision, F1-measure

Besides knowledge about the classifiers i.e. about their advantages and drawbacks, and how they work, it is also necessary to have methods of assessing their quality. In most cases, the measures listed below are sufficient, but sometimes, depending on actual needs, more sophisticated methods have to be used.

Recall and precision are one of the simplest measures of the classifiers quality. They have been derived from the classical Information Retrieval and adapted to the

machine learning and particularly to the classification of text documents. Precision is a measure of the probability that a randomly selected document will be assigned to a category, which will coincide with expert knowledge. Recall is the probability that for the random document, which should be placed in the specified category, such decisions have been taken. Measures of these probabilities are calculated with the formulas:

$$precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$recall = \frac{TP}{TP + FN} \tag{2.2}$$

where:

TP - true positive

FP - false positive

TN - true negative

FN - false negative

Recall and precision should not be considered separately, calculating only one of them does not allow for precise determination of the other. Usually the measures are inversely proportional and increasing one of them results is decreasing the other. Although a system classifying all documents into categories with 100% precision could be prepared without any problems, its recall would be extremely low. In most cases it is desired to get the best possible of both measures which results in the need for a trade-off and determination of a combined measure which would take into account both of them.

One of the combined measures is the F-measure which is defined as follows:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.3}$$

2.8. Representation of the document

This is the last step of the document pre-processing. The final result is representation of a document in the form of an n-element vector of attributes, where n is the number of keywords obtained at the end of the feature selection step. The entire collection of documents can thus be presented in the form of m x n matrix ***a*** (m is the number of documents in the collection), the element a_{ij} represents the weight of the j-th term in the i-th document. The weight values depend on the selected the mode of the document representation. The easiest way is a binary representation which, assigns weights equal to one or zero. Zero is set when the term is not present in the document and one, when it occurs one or more times. Unfortunately, this type of representation does not take into account the fact that some words carry more information than the other. A representation without this drawback is the vector representation, where each element of the vector is a positive real number representing amount of information carrying out by a term in the document.

The collections used in the thesis throughout the process of the data preparation contained, depending on a collection, from 2085 to 6873 of terms. The best results were achieved by getting rid of stopwords and applying a stemming algorithm. Extraction of features on the basis of the subject and the content of each email is an additional function.

Collection	Raw training set	Remove Stopwords	Using steemer	Stopwords + steemer
Collection1	9479	2873	7039	2085
Collection2	20652	5021	18902	4090
Collection3	31569	8621	27648	6873

Figure 7 Number of features in the collection

CHAPTER 3 - MACHINE LEARNING

Machine Learning(ML) attracts the attention of people for its continually widening fields of application. Before the rise of machine learning a lot of research were conducted on intelligent systems. It is commonly accepted that the research in areas such as artificial intelligence, expert systems and cognitive science had a greatest impact on the development of machine learning. Authors in [5] give the simplest definition of machine learning: “the machine is learning the task T on the basis of experience E, where the quality is P and with increasing of experience E also improves the quality of the task T, measured using the P” [29]. In other words, the machine learning classifier is constructed in inductive learning process. At the time when a new document is classified, the classifier recognize relevant features of the new document and based on them classifier compares it with a set of training documents.

The main task of the classifier is to estimate as accurately as possible the function of the relationship between the new document and the category in which it should be found. This function is called decision function. In some cases, new document can belong to more than one category. For example, we can determine the first three categories to which the new document matches.

There are many methods in the field of machine learning methods. Most of them are supervised ML methods. Hereafter the most common methods will be described, and differences between the algorithms will be presented.

3.1. Statistical classification

Statistical classification is a kind of statistical algorithm which assigns statistical observations to classes, based on the attributes (characteristics) of these observations.

Formally, this problem can be summarized as follows: for a given data set $\{(x_1, y), \dots, (x_n, y)\}$ find a training classifier $h: X \to Y$ which assigns the set to an

object class. For example, for the problem of spam filtering, x_i is a certain representation of the message and the y_i takes value "spam" or "not spam".

Learning Machine in its range includes a number of algorithms which may be used in the field of text categorization. Of course, they differ from one another in terms of time learning, precision and implementation. Available literature shows that ML algorithms become increasingly popular for tasks related to text categorization.

The study conducted on the Enron case in [13] argues that the best ways to achieve the highest precision are:

- Naive Bayes (NB)
- Wide-margin Winnow
- Support Vector Machine (SVM)

Other works on text categorization show that methods such as Neural Network and Decision Tree also give good results. The author in [14] has studied different variants of the Winnow algorithm and ultimately proved its superiority over the SVM one.

Hereafter high performance text categorization methods are presented. The methods: Neural Network, Naive Bayes, Winnow and Support Vector Machine have been implemented in this work.

3.2. Artificial Neural Networks

Constant interest in artificial neural networks have been being observing since their invention. They find application in various areas like finance, medicine physics and other. Neural networks are used wherever there is a task of classification, prediction or control. A few features which make artificial neural networks so popular are:

- Power - neural networks can be used for modelling very complex non-linear functions. Thus, they can be very useful in a case where the linear approximation cannot be used. Neural networks also allow to control complex multi-dimensional problems where using of other methods of non-linear modelling is very difficult.

- Ease of use - Neural networks can construct models themselves. A learning algorithm with a set of examples must be run in order to achieve this goal. Only knowledge on how to prepare input data, what type of neural network has been chosen and how to interpret results is needed.

3.2.1. Applications of neural networks

Categorization of text documents and more specifically e-mails is the main application of neural network described in the thesis. However hereafter a few other examples of interesting applications are also provided.

- Disease recognition - measuring health of a patient requires taking into account a number of indicators (heart rate, levels of various substances in the blood, blood pressure, etc.). Neural networks are able to recognize complicated relationships among the factors and conclude an appropriate therapy.
- Financial time series forecasting rapidly changing prices is a common phenomenon in the stock market. With neural network methods it is possible, to certain extents, to predict these variations.
- Credit rating - providing a customer's data like education, age, current job, etc. neural networks are able to predict his/her creditworthiness.
- Machine condition monitoring - for example, neural network can learn the correct sound of the machine. As soon as the sound is different a warning signal can be generated.

3.2.2. Biological Inspiration

The first neural networks were created thanks to research conducted in the field of artificial intelligence. The main objectives of those works were to map the biological nervous systems. The first success in this area was the creation of an expert systems which could conclude what decision should be taken. Further works aimed on preparing a fully intelligent system based on the human brain.

The anatomical studies indicate that the human brain is built up of approximately ten billion nerve cells linked together in a complex network. On average, one nerve cell (neuron) has several thousand calls. Neuron has a branched structure of many inputs called dendrites which collect all the incoming signals. The result is an output signal which leaves the cell via the axon. This signal can be sent to one or many nerve cells. Axons and dendrites are connected to other neurons via synapses. A neuron which was triggered by a synapse trips to the active state.

3.2.3. Artificial neural networks

As mentioned earlier in this chapter the inspiration for artificial neural networks came from neurobiology. The simplest representation of a neuron is a perceptron. It is the smallest unit that occurs in neural networks. Due to the fact that the perceptron is constructed as a linear function it allows to perform only linear categorization of the input values.

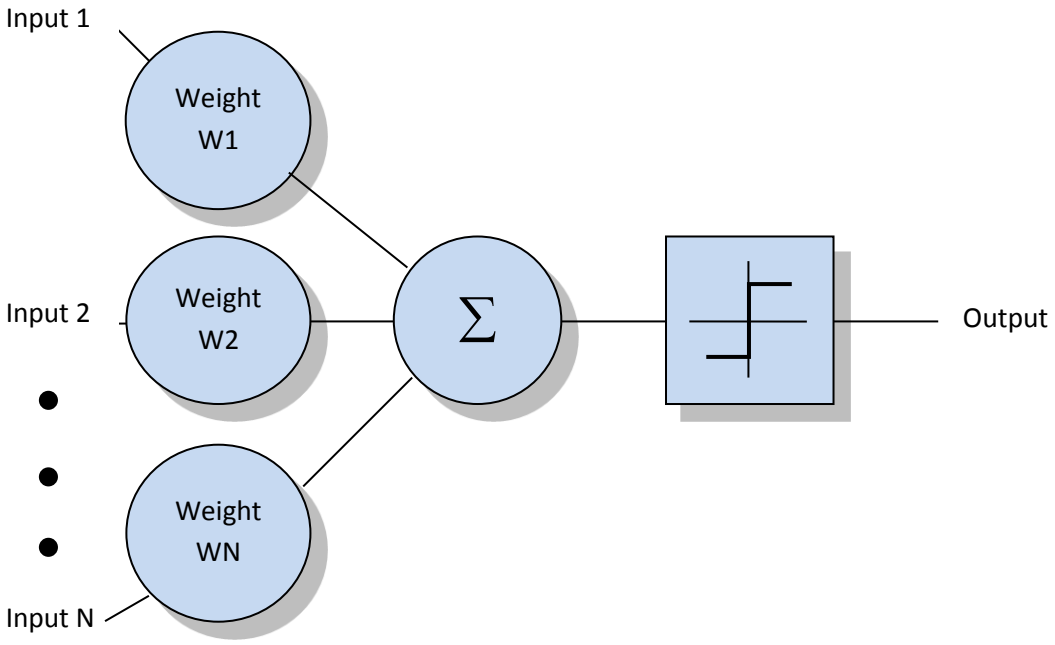

Figure 8 Linear Perceptron

Perceptron as the smallest unit of an artificial neural network works in a simple way. It calculates the sum of the weighted inputs and compares the result with a threshold value.

$$o(x_1,\ldots,x_n)=\begin{cases}1 & if \sum_{i=0}^{n} w_i \cdot x_i > 0\\ -1 & otherwise\end{cases} \quad (3.1)$$

If the sum of the weighted inputs is greater than the threshold the output is 1 which means that the document is classified as correct. If opposite, the output is -1 which means that the document should not be included in the category.

During training, the classifier should seek to obtain the best possible solution for a given objective. In other words, training should learn the classifier how to choose the most suitable vector. Sometimes the examples included in the collection of documents are not linearly separable. If it is the case additional rules have to be applied. The best solution of the problem, very often used for network learning, is using the delta rule which is a special case of the backpropagation algorithm.

In each learning cycle the gradient descent algorithm compares actual values with expected ones. If the error $E(\vec{w})$ is too big the algorithm updates the actual values to minimize the error.

$$E(\vec{w}) \equiv \frac{1}{2}\sum_{d\in D}(t_d - o_d)^2 \quad (3.2)$$

D – set of training examples t_d

o_d – output for the training example *d*

The above described perceptron is able to classify examples according to the linear decision function. In most cases it is not sufficient to solve a given problem. More complex networks use nonlinear decision surfaces. A sigmoid is an unit which enables to obtain better results by using a differentiable threshold function. The structure of the artificial neuron remains the same.

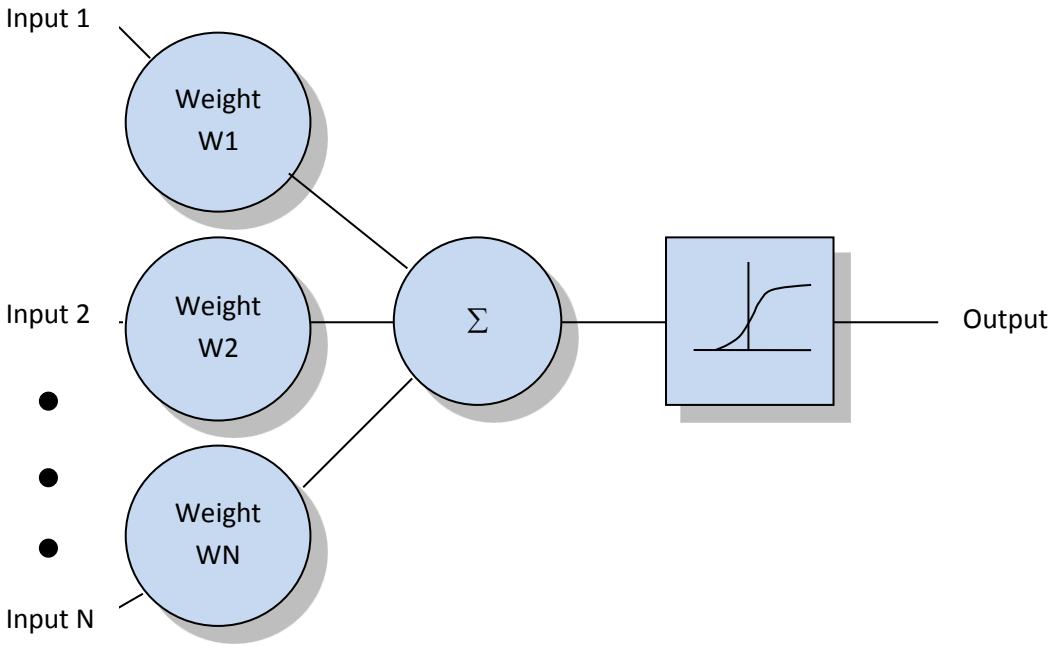

Figure 9 Sigmoid Perceptron

First the sum of the weighted inputs is computed.

$$net = \sum_{i=0}^{n} w_i \cdot x_i \tag{3.3}$$

Then the result is passed to the threshold function which is a sigmoid defined as

$$\sigma(net) = \frac{1}{1+e^{-net}} \tag{3.4}$$

Backpropagation algorithm by far dominated the learning methods for unidirectional multilayer perceptron. The method name reflects its principle of operation, which consists of "transfer" of the amount of error in the output compared to the expected result, in the direction from the output layer to the input layer (and thus reverse to the direction of information flow).

The cycle of the backpropagation learning method consists of the following steps:

1. Determining a response of the output and hidden layers of the neuron to a given input.
2. Determining the amount of error in the output compared to the expected result and sending it back to the input layer

3. Adapting the weights.

Backpropagation algorithm defines a procedure of updating the weights in the network by using a multi-gradient optimization method. The process is based on minimization of measurement error (objective function), which is defined as the sum of squared errors at the outputs of the network (the objective function).

$$E = \frac{1}{2}\sum_{k=1}^{m}\left(z_k(t) - y_k(t)\right)^2 \tag{3.5}$$

Alternatively, the weights can be updated after all training data have been presented, then the number of inputs should be considered in the function.

Steepest descent rule can be used to minimize the mean square error:

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} \tag{3.6}$$

Development of (3.6) results in the following formula which depend on the number of updates:

- for the output layer:

$$\Delta w_{kj}^{out} = \eta\left(z_k - y_k^{out}\right)\frac{df}{du_k^{out}} x_j^{out} \tag{3.7}$$

- for hidden layers:

$$\Delta w_{ji}^{1} = \eta\sum_{k=1}^{m}\left(z_k - y_k^{out}\right)\frac{df\left(u_k^{out}\right)}{du_k^{out}} w_{kj}^{out} \frac{df\left(u_j^{1}\right)}{du_j^{1}} x_i^{out} \tag{3.8}$$

3.3. Naive Bayes classifier

Naive Bayesian classifier is a simple probabilistic classifier which is based on the assumption that the predictors (independent variables) are mutually independent. It is called ‘naive’ or sometimes ‘independent feature model’ as the assumption is usually unrealistic. The probability model of the classifier is based on the Bayes' theorem.

- Naive Bayes probabilistic model
- Parameter estimation
- The design of the probabilistic model classifier

3.3.1. Naive Bayesian probabilistic model

The Naive Bayes model may be expressed by a conditional model (3.9) where C is a class and F1, … F2 are features belonging to the class.

$$p(C \mid F_1, \ldots, F_n) \tag{3.9}$$

If the class C includes a small number of inputs, the above formula can be used sufficiently. However, as the number of inputs is increasing and their values are getting bigger values, the basic model is more and more insufficient. Then the model needs to be reformulated in order to be of practical use. Using Bayes' theorem:

$$p(C \mid F_1, \ldots, F_n) = \frac{p(C)p(F_1, \ldots, F_n \mid C)}{p(F_1, \ldots, F_n)} \tag{3.10}$$

The most important in this equation is the counter. Due to the fact that the parameter C has no influence on the denominator, in fact, the denominator is a constant.

The numerator is equivalent to the joint probability model

$$p(C, F_1, \ldots, F_n) \tag{3.11}$$

which can be written using the conditional probability:

$$
\begin{aligned}
& p(C, F_1, \ldots, F_n) \\
& = p(C)p(F_1, \ldots, F_n \mid C) \\
& = p(C)p(F_1 \mid C)p(F_2, \ldots, F_n \mid C, F_1) \\
& = p(C)p(F_1 \mid C)p(F_2 \mid C, F_1)p(F_3, \ldots, F_n \mid C, F_1, F_2) \\
& = p(C)p(F_1 \mid C)p(F_2 \mid C, F_1)p(F_3 \mid C, F_1, F_2)p(F_4, \ldots, F_n \mid C, F_1, F_2, F_3)
\end{aligned}
\tag{3.12}
$$

Assuming that each feature F_i is conditionally independent of every other features Fj for $j \neq i$ which means

$$
p(F_i \mid C, F_j) = p(F_i \mid C) \tag{3.13}
$$

the joint model can be expressed as

$$
\begin{aligned}
& p(C, F_1, \ldots, F_n) = p(C)p(F_1 \mid C)p(F_2 \mid C) \cdot \ldots \cdot p(F_n \mid C) = \\
& = p(C)\prod_{i=1}^{n} p(F_i \mid C)
\end{aligned}
\tag{3.14}
$$

This means that under the above independence assumptions, the conditional distribution over the class variable *C* can be rewritten as

$$
p(C \mid F_1, \ldots, F_n) = \frac{1}{Z} p(C)\prod_{i=1}^{n} p(F_i \mid C) \tag{3.15}
$$

where Z is a scaling factor dependent only on $F_1, \ldots, F_n$. Models of this form are easier to implement, as they decomposes the model into a class called "prior" $p(C)$ and independent probability distribution $p(F_i \mid C)$. If there are *k* classes and if the model $p(F_i)$ can be expressed by means of *r* parameters, then the corresponding naive Bayes model has $(k-1)+nrk$ parameters. In practice usually *k=2* (binary classification) and *r=1* (Bernoulli variable as a feature) and the total number of parameters of naive Bayes model is $2n+1$ where *n* is a number of binary features used.

3.3.2. Parameter estimation

In the case of supervised learning, parameters of a probabilistic model have to be estimated. As the assumption has been made that features are independent it is sufficient to estimate the previous class and further features of the model independently. This can be accomplish by using the maximum a posteriori probability method (MAP), Bayesian inference or other parametric estimation procedure.

3.3.3. Construction of a probabilistic model classifier.

Assumption that the features of the model are independent leads to naive Bayesian probabilistic model and naive Bayesian classifier which describes a decision-making rule. The general rule is to extract the most probable hypothesis. The corresponding classifier is defined as follow:

$$classify(f_1,..., f_n) = \arg\max_c p(C = c)\prod_{i=1}^{n} p(F = f_{i_i} \mid C = c) \qquad (3.16)$$

On the basis of this description and formulas described in this subchapter it can be concluded that the classifier is suitable for incremental learning and can easily be stored in the database

3.4. Winnow Algorithm

In this chapter a family of algorithms based on Winnow concept will be described [30]: Positive Winnow, Balanced Winnow and the Modified Balanced Winnow. A common assumption has been made that the incoming example x_t is a vector of positive weights. The assumption is often fulfilled by the NLP methods, where *x* are based on frequency of words. Usually the TF-IDF (term frequency – inverse document frequency) numerical statistic is used which computes weights of the terms. The thesis uses a classifier based on the simple frequency of words.

Learning phase is a training process which is performed in a sequence of trials. Initially the algorithm makes a prediction and then it receives a feedback which is

used for updating the weights vector. Winnow [21] algorithm is constructed of 3 parameters: promotion α , demotion β and threshold.

3.4.1. Positive Winnow

Positive Winnow keeps all the feature weight vectors in the feature collection. In the beginning the weights vector is assigned positive values. Characteristic of the parameters is as follows:

- promotion parameter $\alpha > 1$
- demotion parameter β, where $0 < \beta < 1$
- threshold $0th > 0$

The winnow algorithm predicts 1 for a document x if:

$$\sum_{j=1}^{m} w_j \cdot x_j > \theta \tag{3.17}$$

where xj is the $j-th$ feature of document x and w_j is the *j*-th weight. If a mistake is made the algorithm updates its hypothesis according to the rules:

- if the correct prediction should be 1, and the classifier predicts 0, then the weights of features which achieved the threshold are promoted.
- if the correct prediction should be 0, and the classifier predicts 1, then the weights of features which achieved the threshold are demoted.

3.4.2. Balanced Winnow

The Balanced Winnow algorithm is constructed to keep only two weights w^+ and w^- for each feature from the collection. The final weight is a difference between these two values. The classifier predicts 1 for a given document x if

$$\sum_{j=1}^{m}\left(w_j^+ - w_j^-\right)\cdot x_j > \theta \qquad (3.18)$$

if a mistake is made the algorithm updates the weights of features which achieved the threshold according to the rule:

- if the correct prediction should be 1, and the classifier predicts 0, then the weights of features which achieved the threshold for weights w^+ are promoted and for weights w^- are demoted.
- if the correct prediction should be 0, and the classifier predicts 1, then the weights of features which achieved the threshold for weights w^+ are demoted and for weights w^- are promoted.
- It has been shown that this classifier can effectively learn any linear threshold function and works relatively well when linear separation areas do not exist. Theoretical analyses show that the algorithm behaves correctly even in the case of hype and irrelevant words. Additionally, due to its increment nature, it provides drift method of separation (approximated function). As it does not require complex calculations, it is well suited for use with databases. Moreover, it has built-in support for incremental learning where modification of weights can be performed any time after incorrect classification of the document.

3.5. Support Vector Machine

Nowadays Support Vector Method (SVM - Support Vector Machines) is the most popular methods used in data mining. It specifies a method of construction of a hyperplane or hyperplanes separating considered objects. The boundary between the objects does not reflect positions of all objects but only those which are in its close vicinity (coordinates of these points define the so-called supporting vectors). In the absence of the possibility of sharing a hyper-plane made immersion of the objects to a larger number of dimensions in such a way that new points were characterized by a linear separability. If there is no possibility to construct a hyperplane (the objects are

not linearly separable because of their quantity) then the objects should be mapped to a higher dimensional space what may make the separation easier.

The SVM plays an important role in text mining as it can be adapted to different methods of representation of textual information. The SVM has been successfully used to classify documents in the application using both frequencies of the words [16], [15] representation of complex structures [17], [18]

In case of a problem including a lot of classes, learning takes place by dividing the problem into a number of binary sub-problems and observing one class vs. all others.

Hyperplanes for linearly separable classes can be defined as:

$$H_1 : x_1 w + b \geq +1 \ \ for \ \ y_i = +1 \tag{3.20}$$

$$H_2 : x_1 w + b \leq +1 \ \ for \ \ y_i = -1 \tag{3.21}$$

The margin between the hyperplanes can determined with the following formula:

$$m = \frac{2}{\|w\|} \tag{3.22}$$

The objective of Support Vector Machine is maximizing the margin m while keeping the hyperplanes definitions (3.20) and (3.21) unchanged. The hyperplanes separate two classes. The Support Vectors lay on the decision-making hyperplanes.

Learning the classifier requires minimization of the objective function which in this case is the task associated with the Quadratic Programming (QP). In the past when the QP was introduced, it required very intensive processing - proportional to the square of the number of training documents. Recently developed algorithms divide the main task into several subtasks, then each of them is processed separately and eventually the results are combined. This attitude speedups the total processing time significantly. The algorithm used to solve linearly separable problems can now be

extended to support non-linear separation. This can be achieved by introducing mild hyper-plane boundaries or transforming the original vectors to higher dimensions.

The SVM classifier with learning algorithm modified to use the QP can handle incremental learning The classification process is well suited for use with databases. Unfortunately, the learning phase cannot be said the same.

CHAPTER 4 – AGGREGATION OF CLASSIFIERS AND VOTING SYSTEM

Current methods of aggregation of classifiers use different subsets of the training set and construct several classifiers, eventually combine the results of individual classifiers. Such systems are often referred to as the voting systems. The word "combination" highlights the metod used which means that a linear combination of individual classifiers is used.

The most common variant of this strategy includes algorithms such as voting, bagging, boosting and fuzzy methods, which are discussed below.

4.1. Voting

One of the easiest and most popular way of combining classifiers is majority voting [19, 20]. The result is a binary classifier which takes true if the majority of votes is true. Each weak classifier is involved in the decision to classify N input vector. This method involves taking a final decision in accordance with the number of votes cast by each classifier for each class C, so assigning X o the class that receive a majority of votes. When working with data sets that contain more than two classes, the winner class is the one which received the highest number of positive votes. To accomplish the voting task, a number of criteria should be taken into account [26]. A major problem with usage of this method is determination of the accuracy threshold values for selecting individual classifiers. The author in [27] suggests that the individual classifiers which have not obtained the accuracy above 50% should not be taken into account. This option is often omitted, so that the final vote could give underestimated result [28].

4.2. Fuzzy max operator

Fuzzy logic plays a big role in the family of classifying different types of text documents. Possibility of applying it to aggregation of results allows to achieve better results. The results are based on dynamic selection of weights of individual classifiers. The fuzzy logic family contains a lot of operators which can be used in the

classification task, such as: AIVA, OWA or fuzzy max operator, which is part of OWA. Only the last one has been implemented in this work.

The fuzzy max operator works with a combined sum of all classifier. This combining process is performed with finite training sets. During testing, values of individual classifiers are calculated for each new e-mail and it is assigned to the category of the highest value scored.

4.3. Bagging

The bagging meta-algorithm, by using a standard training set $T = \{t_1, t_2, ..., t_m\}$, and classification algorithm $class(\ )$, creates a set of classifiers $class(\ ) = \{class_1(\ ), class_2(\ ), ..., class_k(\ )\}$, where a classifier $class_i(\ )$ is a randomly selected sample of T with replacement. The size of each sample is the same as the cardinality of the training set, so that some of the documents may be repeated in the collections T and some may not be taken into account.

A new document x is analysed by each classifier of the set $class(\ )$. The result is a set of identifiers $I : \{I_1, ... I_k\}, I \in K$, where K is a collection of the set I and determines the class which the document x should be assigned to.

4.4. Boosting

Boosting is a general and effective method for creating an efficient classifier by combining many weak classifiers. The most common algorithm which implements the boost method is AdaBoost algorithm. In contrary to the majority of classification algorithms, AdaBoost uses a binary classification (two classes). It is expected that combining a number of weak classifiers will give at least 50% better results than individual classifiers can do.

Simple decision trees are the most chosen for this role - in some cases a tree consisting only of a single node may be sufficient.

Initially all objects from the training set are assigned weights $\omega_i = \frac{1}{N}$, where i is a particular object number and N is number of all objects in the collection. Then a new training set C_i is formed of randomly selected *h* objects of the training set and C_i is used to train the first weak classifier K_i. A classification error for the created classifier K_i is determined as the sum of the weights E_i of wrongly classified objects and the scaling weights α_k are calculated according to (4.1)

$$a_k = \frac{1}{2}\ln\left(\frac{1-E_k}{E_k}\right) \tag{4.1}$$

Based on α_k new weights for all objects in the training set can be calculated according to formula (4.2

$$\omega_i = \omega_i \cdot \begin{cases} e^{-\alpha_k} & \text{for correctly classyfied objects} \\ e^{\alpha_k} & \text{for incorrectly classyfied objects} \end{cases} \tag{4.2}$$

After new values have been determined, the weights are being normalized. Each iteration of the algorithm creates a new learning subset C_k having weights attached to individual objects. Objects with larger weights are more likely to get into a subset of the learner. In this way, weaker classifiers which were previously misclassified are better prepared in terms of $\mathrm{E_k}$ for classification (difficult cases). For each weak classifier trained K_k the algorithm determines E_k, α_k and a set of weights for the objects in the training set. The algorithm is executed subsequently $N-1$ times.

Classification of an unknown object is to classify target objective functions T by trained classifiers K_k by aggregating their responses $f_k(x)$ being multiplied by the corresponding scaling factors a_k:

$$a\,choosen\ class = \sum_{k=1}^{T} a_k f_k(x) \tag{4.2}$$

CHAPTER 5 - RELATED WORK

This chapter is dedicated to related work in the field of text categorization (TC). In this work it was not possible to consider all aspects of the field. Described issues are most relevant to the topic of the study. The first section relates to the automatic categorization of e-mail messages in the following topics are discussed work on automatic text categorization, and sample applications.

5.1. Automatic Categorization of e-mails

Automatic categorization of e-mails is a very large area in which a lot of people doing research on various collections of e-mails. The two most popular collections publicly released are Enron Corpus and SRI Corpus. The first one contains over half a million e-mail messages from Enron company. The second collection is SRI research project, which contains the messages that belong to scientists and people associated with the project

In a study conducted by Ron [31] the author is working on both data sets. The report described classifiers which were divided into two groups: generative and discriminative classification approach. Tests were conducted with four classifiers: MaxEnt, Naive Bayes, SVM and Winnow. For experiments with the Enron Corpus only messages of six people who had the highest number of e-mails were taken into account. All non-topical folders such as "all_documents" or "contacts", were omitted. From the SRI research project the author chose, as in the case of Enron Corpus, only people who had a lot of themed folders and categories. Training and test sets were distributed in the incremental way based on time-stamp. The results presented were calculated with accuracy. They show that the Support Vector Machine achieved the highest accuracy in the Enron Corpus, which varied, depending on the category, between 94.6 and 56.4. The other algorithms gave similar but nevertheless worse results than the Support Vector Machine. In the case of SRI Corpus the results with the highest accuracy were similar to the highest accuracy results of SVM. Depending on the category, Winnow and MaxEnt won the second or the third place. In both cases,

Naive Bayes achieved the worst accuracy which, depending on the category, varied between 32.0 and 75.0. Winnow with the weighted sum of all categories taken into account, reached the accuracy of 55.08. The next were, in ascending order, MaxEnt with 70.87 and SVM with of 71.52 what was the best result.

Fabrizio Sebastiani [32] describes several methods based on using indexed words from documents. Such methods, called “controlling vocabulary”, greatly improved the process of putting documents into right categories. A few examples where such approach works well are: articles in newspaper which have no titles or the titles are wrong. By analyzing content of an article it is possible to recognize the class where the article should be assigned to. Maurice de Kunder in [32] described also an approach called Text Filtering. Nowadays this approach has been successfully applied in a very broad range of tasks, such as: Spam Filtering, Junk Advertisement or Adult Offers. Another very important topic discussed in [32] is Word Sense Disambiguation (WSD). The term WSD refers to the fact that one word might have more than one meaning. This is very important issue that must be tackled when a document is quite small and there are a lot of stop words in its content. These kind of documents are prone to incorrect classification. WSD has been applied in many tasks but the most important ones are those in test processing related with linguistics and computational approaches. WSD is very helpful in word choice selections, spelling correction, part of speech tagging and other similar tasks. Manco in [3] distinguished three types of text representation used in emails:

- Unstructured text
- Semi structured text
- Numeric data

Manco also examined semi-structured categorization. He proved that information contained in fields as “to”, “from” or “keywords” should be treated differently from unstructured fields such as “e-mail content” or “subject”. Information contained in the thread is the most important text in the e-mails. Semi-structured text is processed in advance. In addition, each e-mail containing numerical data can be used to improve the quality of classification. For example the number of recipients, message size or its

length can be combined in one processing step of classification and give decent results.

5.2. Automatic Text Categorization

Mathiassen in [35] introduced the work associated with an increase in the quality of classifiers based on the WIPO-Alpha Document Collection. In his work, three different structures were used to the preparation of document collections, depending on the characteristics: "... a) how features are indexed, b) how features are Represented, and c) how the process of feature reduction is performed." Representation of documents is completed in three phases: Reduction dimensionality, Indexing Feature and Feature Weighting. Preparing the collection includes the removal of stop-words and stemming. Features are weighted using two methods: frequencies term and term frequency - inverse document frequency. Four tests are conducted on the individual classifiers KNN, LLSF, Neural Network and Winnow, and combined results of these classifiers are presented. The combined classifiers user are Binary Voting, Weighted Classifier Combination, Dynamic Classifier Selection and Adaptive Classifier Combination.

The results obtained in the test phase are presented with the recall precision and F1 measure . The results obtained by using the combined methods yield better results as compared with the individual classifiers.

In the work [34] about re-examination methods in the field of text categorization author examines five different classifiers: the k-Nearest Neighbours, the Neural Networks, the Last-squares linear fit the Naive Bayes classifier and the Support Vector Machine. Document collection chosen for test is Routers-21578 corpora. All unlabelled documents were eliminated from this corpus. Each category included at least one document in the training set as well as in the test set used for examination. Though the selection of learning collection was carried out in full compliance with the supervised approach, the process resulted in 90 categories in the training set and test. 82% of the categories had less than 100 documents and 33% had less than 10

documents. Evaluation of the effectiveness of the system was provided by using recall precision and F1 measure.

Vapnik in [36] implemented SVM - for solving two-class recognition problem. The structure of this method is based on the Structural Risk Minimization [36,37]. The idea is to find the best decision surface which separates data points between two classes. However, for high dimensional space this model cannot be applied.

Two methods can be used to solve problems where linear separation is not possible. The first one is soft Margin hyper-plane. The second one is mapping the original date vectors to a higher dimensional space where the new features contains interaction terms of the original features, so the data can be separated linearly [36,37,38]. The presented results show that SVM has the highest F1 measure at the level of 0.85, next is KNN with 0.85 and then NB with 0.79. Author concludes that SVM and KNN classifiers are significantly better than other ones, and NB is far worse.

Mailboxes structure varies depending on user's preferences and evolves constantly during everyday usage. In studies [39], the authors are conducting research on maintaining the right boxes without much user intervention. The hierarchical structure of folders has been used to build the system called eMailSift. The system uses Naive Bayes approach introduced in [40]. The system takes into account the structure and content of e-mails. Removal of all stopwords reduces 40% of features. The standard numerical statistic TF-IDF were used for words weighting. The precision of results which was is 65%.

5.3. Examples of usage

In the initial period of NLP growth, automatic document categorization was perceived as a specialized technique applicable only to organizing collections of documents e.g. in libraries or in collections of legal information. Such tasks which can be described as classic applications, keep being essential for developers dealing with categorization of documents. However, with development of the Internet, a lot of new problems

related to the processing of documents emerged, for which the automatic classification can be applied.

A typical issue with automatic categorization of documents is repository. Such a repository - regardless of whether it is a library or a set of legal or scientific documents must have a logical structure which allows not only for searching for specific documents, but also for finding related documents, and eventually, for assessing complete sets of documents. To accomplish this goal, the most common approach is to divide the repository into thematic groups which contain publications of similar topics.

While usage of the structured sets can be quite comfortable, their creation can be cumbersome. An automatic categorization system can be used directly to support the task. At first a project repository with thematic groups is created. Then a set of training examples is created by means of manual selection of the most representative of each group documents. Then automatic assignment of the remaining documents can be performed.

Of course, after this process and depending on the quality of the categorization system, a number of documents is classified incorrectly. Therefore it is necessary to check the resulting repository manually. Categorization system can be used not only to initial structuring a set of documents, but also to its maintenance which usually means adding new documents (less frequently – adding new thematic groups). For this purpose the entire contents of the repository can be considered as a training set. Despite the performance of the system must be controlled manually, thanks to the classifier, the inspector does not even need to know the structure of the repository (as opposed to an expert who would perform manual assignment of the document), nor the thematic groups having been defined. The inspector's task is merely to verify if the document is consistent with the content of the system group.

Another application of categorization systems is related to identification of writers. If information about the author is not given or plagiarism is suspected then searching through all documents and verifying their contents may be required.

Identification of authors can be performed in the same manner as discussed before, by assigning documents to topics. In this case, the training collection consists of classes which group documents created by different authors. At least two conditions must be met for the identification to be performed effectively. Firstly, the repository should contain a big number of documents created by the putative author of the analyzed text, practically this method can be applied only to very large collections of documents. Secondly, an n-gram text representation is desired to reflect characteristic styles of individual authors.

Most searches on the internet can be classified into one of two groups. The first one is searching for a single, well-defined web page (or, more generally, a single network resource) e.g. a colleague homepage, a particular company official website or a scientific publication with the title and author name given.

The second group is not related to a specific documents or even to the structure of the World Wide Web. The purpose of these kind of searching is to find information on a given subject, neither source of information nor its presentation is important. Searching information about the planets of the solar system is an example. Diverse objects can be obtained as a result, be it text documents with information about the planets and solar system, three-dimensional VML models or photos of planets posted by NASA.

What distinguishes the two groups is that the former treats the web as an unstructured "black box" or even as an autonomous expert system while for the latter, the structure of the network is important.

Modern search systems offer pretty good possibilities of constructing queries to the document text layer, but it is not enough to search for multimedia documents which even the simplest web pages are. It means that the type of the website is of quall importance. Also appearance of a web page is important and may be remembered but its contents not e.g. the web site of New York Times is often browsed for general news rather than for an exact one.

CHAPTER 6 – E-MAIL CATEGORIZATION MODEL

This chapter presents data on e-mail categorization model. The model is based on the knowledge gained in Chapters 3 and 4. It is not a copy of any existing system. The entire computer program has been implemented in Microsoft Visual Studio using C#. The data have been obtained from MySQL database in the form of an XML file.

Figure 11 depicts all stages of the program. In the first stage a collection of documents in the form of XML files is taken. In the second stage, four classifiers are trained with the same collection of data. Stages 3 and 4 integrate the results obtained with the individual classifiers.

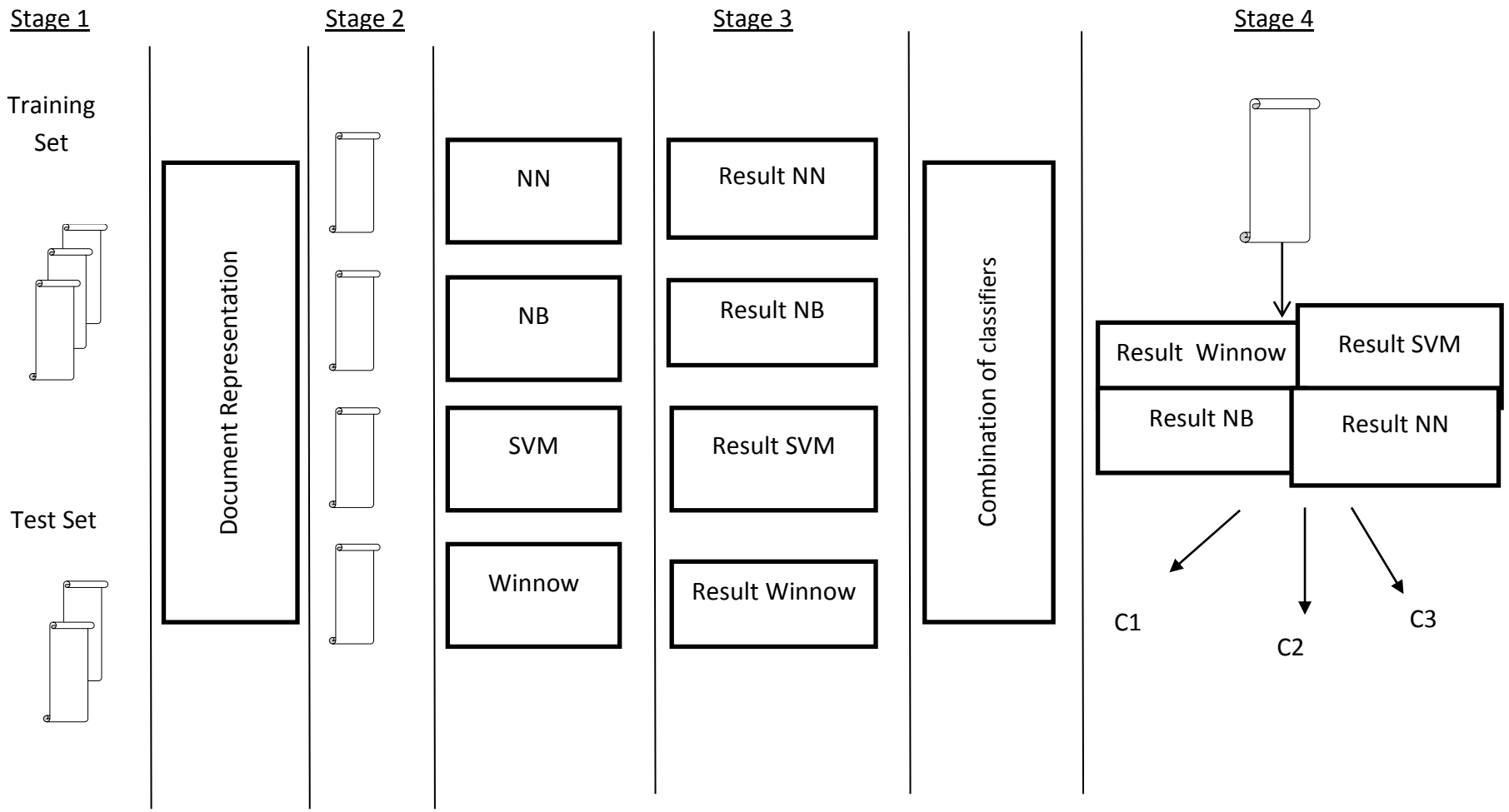

Figure 10 Model of E-mail Categorization program

Test sets are excluded from the training process. Testing is done by calculating the accuracy of the classifier.

To make the data understandable to the computer the following steps are performed:

- Removing stopwords
- Stemming
- Removing digits
- Putting to the lowercase

Figure 12 depicts steps which are necessary to prepare a data collection.

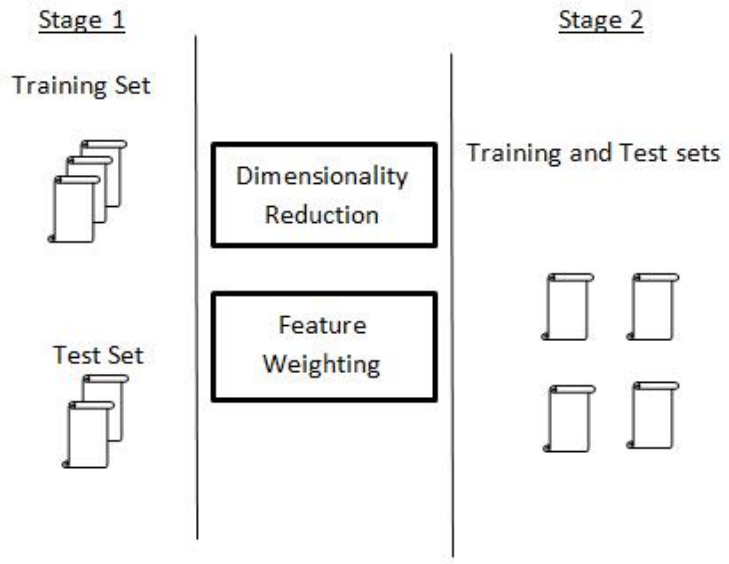

Figure 11 Document collection representation

6.1. Document Collection

Each email message consists of several sections, see Figure 13. All sections are stored in an XML document. Both the training set and the test set have the same structure. For the training process only e-mail body and subject sections are taken into account. According to a user's preferences one of them or both can be considered. The best results are achieved when both sections are taken for the learning process.

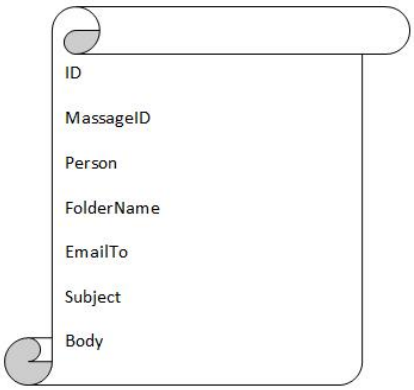

Figure 12 E-mail sections

6.2. Design and implementation

Four classes are used for a document representation: Category, Email, Email Collection and Term. All classes are kept in the *Model* folder.

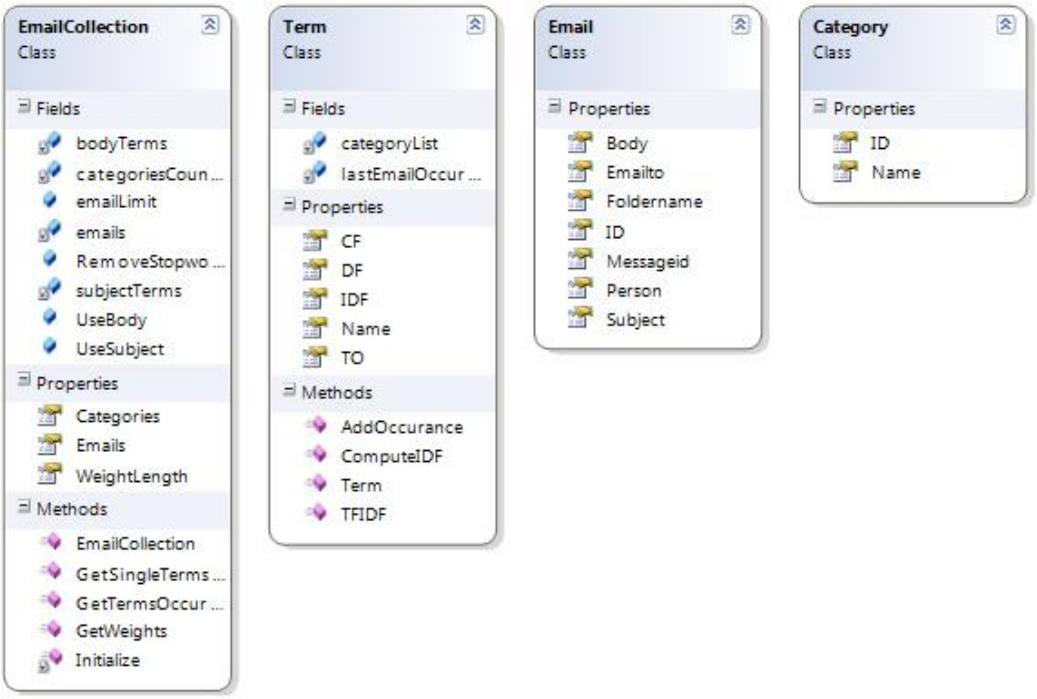

Figure 13 Representation of documents

The *Category* class includes all the categories which are present in the xml document. In the program the categories are represented as: category *Id* and category *name*. The *name* category contains the structure of the mailbox folders created by Enron's

employees according to their own preferences. The *Email* class includes all sections which occur in every single e-mail: *Id*, *Person*, *EmailTo*, *Subject*, *MassageID*, *Body*, *FolderName*. The *Foldername* section used is also created by Enron's employee. The *Term* class contains weights generated for all words taken from *e-mails*. The weights are generated by means of either the term frequency method or frequency or term frequency - inverted document frequency, depending on actual needs. The *EmailCollection* class is responsible for creating a collection of documents. At this stage methods responsible for downloading collection of all sections of each email are called. This class also includes methods responsible for pre-processing.

The *Input* folder contains *EmailReader*, *PorterSteemer*, *TextProcessor* and *XmlSanitizingStream* classes which are responsible for preparing the input data. The *EmailReader* class is used to read all columns of data from an XML file. The *TextProcessor* class is responsible for preparing the data for conversion to a digital form. All sections which have been used in the learning stage are taken into account in this stage. Using the Porter algorithm and removing unnecessary characters and words gives a significant reduction in dimension. A supporting trick is to bring all the words to lowercase.

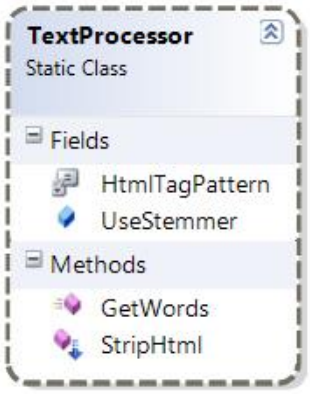

Figure 14 Text Processor

6.3. Machine Learning algorithms

Each classifier is constructed separately but inherits from the *Classifier* class. It gives the possibility to use the methods that are contained therein. The most important

methods of the *IClassifier* class are *Train* and *Classify*. Two additional methods are implemented: *IsLearningComplited* and *ClassificationScore*. The *Train* method accepts the entire training set which is in the form of a dictionary containing terms along with their numerical values. The *Classify* method is responsible for retrieving new e-mails to be classified to categories. This is performed incrementally .

The *IsLearningComplited* is invoked after each training method to check if the learning has been done successfully. The *ClassificationScore* combines all the results obtained during the classification processes into the precision, recall and F1-measures.

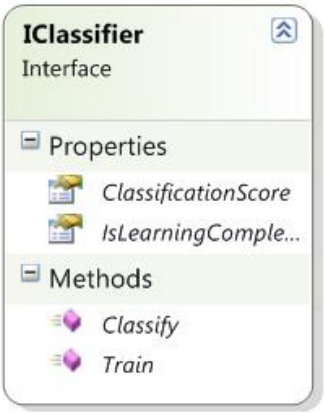

Figure 15 IClassifier

6.3.1. Neural Network

The *NeuralNetworkClassifier* class is built using the *NeuroBox* library which provides support for rapid implementation. The *Config* field allows to adjust parameters of the network learning. They may be configured in the program.

The NeuroBox provides a comprehensive library for customization of the training parameters according to one's preferences so that allows to improve quality of the classifier. Hereafter basic functionality of the library is described. Adaptation of the parameters are generally referred to regularization which prevents over-fitting.

6.3.1.1. Manhattan Propagation – optimization for backpropagation

Backpropagation algorithm is by far the dominant technique for training neural networks. The incremental behavior allows to change the weights in each cycle. However, the problem associated with this technique is that the calculated partial derivatives may be too big or too small. The Manhattan Propagation training algorithm is a simpler version of the Resilient Propagation one. Both are designed to optimize the results obtained during the learning network. The Manhattan method uses partial derivatives to identify the sign only to be used for updating the weight matrices. Updating the weight consists of adding or subtracting a constant weight to of the matrix, value of which depends on the network being trained. The process starts from higher values of weights and, if necessary, they are reduced.

6.3.1.2. Learning rate and momentum

Speed of the backpropagation learning algorithm depends on the parameters learning rate which is the number of weight updating cycles that must be performed to reach a local minimum. In most cases, if the searched value is far from the minimum, the learning rate should be large with small momentum. If the value is getting close to the minimum then the learning rate should be gradually reduced with momentum increased. If the weights are adjusted correctly then the training error can be used to determine the end of the workout. If the error is not changing with successive cycles then the learning rate should be reduced.

6.3.1.3. Weight decay and flat spot elimination

The weight decay is another method used for preventing over-fitting. The method introduces a penalty term to the error function. The learning process starts with high initial values of weights and successively decays them. As the weights are not needed for reducing the error function they become smaller and smaller and eventually they are eliminated. The weight decay method is used for regularization which is to punish large weights in the neural network.

Another method used for regularizing the network is using a flat spot elimination value. The value is a constant which has to be added to the derivative of the activation function. This enable for passing flat spots of the error surfac. Typical values are between $_0$.1 – 0.25, most often 0.1 is used.

The *NeuralNetworkClassifier* uses a broad range of possible solutions from the NeuroBox library.

The Neural Network implemented in the thesis allows to adjust the number of hidden layers in the network and the number of nodes in each layer. The *Classify* and the *Train* fields are inherited from the *IClassifier* class.

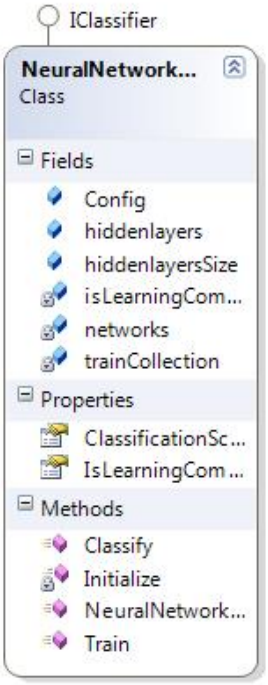

Figure 16 Neural Network Classifier

The training method used for the network learning is based on the backpropagation algorithm. The process can be illustrated by the following pseudocode:

```
public void Train()
{
   for (int i = 0; i < trainCycles; i++)
   {
      foreach (training example x)
      {
         PushUnboundInput(x);
         PushUnboundTraining(MainClass of x);
         TrainCurrentPattern();
      }

      float score = ValidateAsBestInClass();
      if (score > bestInClassValScore)
      {
         bestInClassValScore = score;
      }
   }
```

The training is carried out by a user defined number of cycles. In each cycle, the algorithm propagates learning error backward. In each cycle, the algorithm attempts to adjust the weights to obtain the results as close as possible to the ideal.

6.3.2. Naive Bayes

The second classifier implemented in the program is *NaiveBayesClassifier*. Construction of the classifier and technical details have been discussed in the Chapter 3. The classifier determines the probability of belonging of an e-mail to a category. The implementation of the algorithm does not use any additional libraries of the ML family.

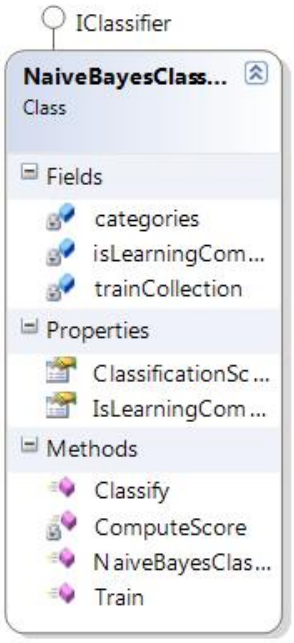

Figure 17 Naive Bayes Classifier

6.3.3. Support Vector Machine

The third classifier implemented in the program is *SuppurtVectorMachineClassifier*. Construction of the classifier and technical details have been discussed in the Chapter 3. The classifier operates by performing linear separation between classes. In contrary to the Neural Networks which use a dictionary of indexed features the classifier uses an incremental approach: one class vs. all the others in turn. Learning is finished at when all classes have been examined. The final result of the classification process is always a single category.

Methods inherited from the *IClassifier* class provide learning and testing algorithms in the same way as is done for other individual classifiers.

The algorithm is implemented with OpenCL Library which is a powerful tool designed for multi-thread calculations. It includes many useful classes such as: *LinearAlgebra*, *DifferentalEquations*, *Fouriertransform* or *MachineLearning*. The program implemented with the thesis uses only the last of these items. The library provides a multi-SVM training and adjustment of parameters of learning (ConfigSVM). Parameters made available by the library are: Lambda Regularization parameter,

selection a type of the kernel functions, Numerical Tolerance and MaxPasses Times of iteration cycles without changing the Alpha parameter.

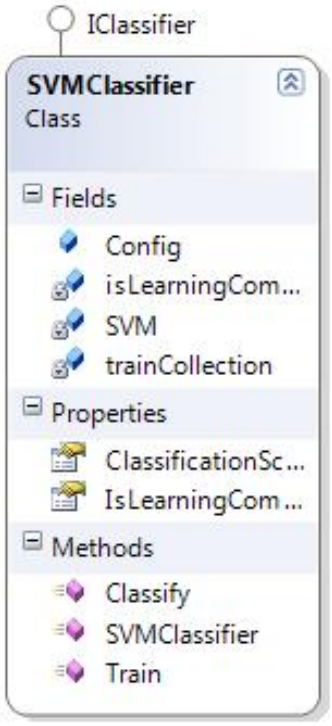

Figure 18 SVM Classifier

6.3.4. Winnow

The fourth classifier implemented in the program is *WinnowClassifier*. The construction of the classifier and technical details have been discussed in the Chapter 3. Winnow is a linear classifier which tries to separate linearly positive and negative examples in the classes.

To simplify the implementation the WinnowClassifier does not use any external library. The implementation allows to set the Alpha and Threshold parameters which have been discussed in the Chapter 3.

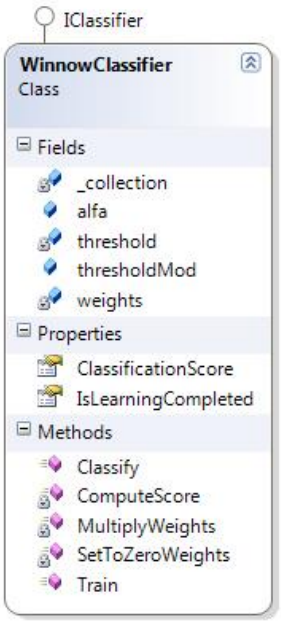

Figure 19 Winnow Classifier

6.3.5. Aggregation of Methods

Two methods of combining classifiers have been implemented in this work:

- Binary Voting
- Fuzzy max aggregation

Combined classifiers themselves does not take part in training of individual classifiers. Instead, results of the individual classifiers are retrieved and processed.

Binary voting is based on the results of trainings conducted by individual classifiers. Each workout is considered to be one single result. The final result is determined by a majority of votes. Results achieved by the classifiers must be normalized to the binary form. Each new e-mail should be assigned to the category that has obtained the highest number of votes.

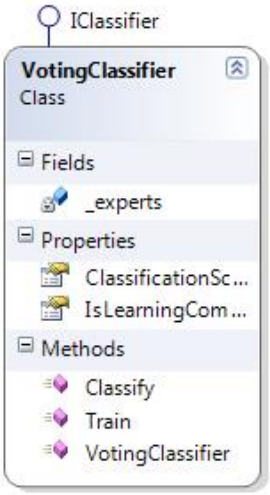

Figure 20 Voting Classifier

The *VotingClassifier* inherits from the IClassifier class and is implemented without any reference to additional libraries.

Fuzzy max aggregation is another method of combining the results. Like the voting classifier the fuzzy max does not participate in individual classification processes. During testing, the method operates on results obtained by individual classifiers. Due to the fact that the corpus contains 25 categories, the classifier retrieves 25 results from individual classifiers. The results are normalized to a float number between 0 and 1. The SVM classifier gives results in the binary form 0 or 1 so the results are presented as [0001000 ...] where 1 means that the category won. The result with the biggest number is the winner.

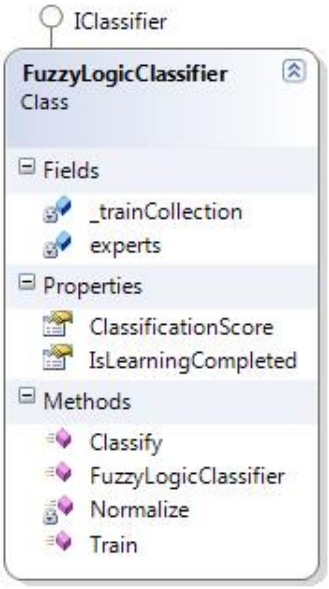

Figure 21 Fuzzy Logic Classifier

Figure 22 shows the implementation of the fuzzy operator max method. Again, the implementation does not use any additional library. The classifier inherits from the IClassifier class so that the methods such as *Classify* and *Train* are available in this class. Additional methods such as *Normalize* and *FuzzyLogicClassifier* allow for the classification without interference with training phases of individual classifiers.

CHAPTER 7 – EXPERIMENTAL SETUP AND RESULTS

This chapter describes the experiments performed, the test setups used and the results obtained for:

- Neural Network
- Naive Bayes
- Support Vector Machine
- Winnow
- Voting
- FuzzyMax

Each method was evaluated using precision, recall, F-measure. Each experiment was performed with the same databases so that the results could be compared. All experiments used the same data collection which were divided into the training and the testing sets. Each set contained the same number of categories.

7.1. Document Representation

The representation of documents always starts from creation a list of features from the training set. Removal of stopwords and stemming give significantly better results though these options can be omitted in the program. Although each e-mail in the database is composed of a number of sections, only the „subject" and „body" sections have been used in the experiments. The „foldername" section which contains names of folders created by the mailbox owners is considered as a set of categories which are divided according to their owners. Each mailbox owner is represented in the database as a "person".

7.2. Feature Reduction

The collections of documents having been used in the thesis is divided into three collections. The first one contains only 417 e-mail messages, the second one contains 4104 e-mails and the last ones contains 8241 e-mails. Though the set is composed of

subject and body sections only, without any other modifications, the smallest collection contains 9479 words, the medium one contains 20652 words and the biggest one contains 31596. The body set itself contains 8725 words. The large number of words increases chances of correct learning significantly. While choosing combinations of words from the body and subject sections, prior removal of stopwords and stemming turned out to be the most efficient solution which resulted in the final set of 2085 words for the smallest collection, 4090 words for medium one, and 6873 words for the biggest one.

7.3. Neural Network

A few parameters are used for the Neural network learning. The parameters having been used in the thesis were at the rate 0.1.

Learning network with one hidden layer forced and 100 training cycles gave worse results than learning without any hidden layer and with the same number of cycles.

Figure 23 depicts results of learning with one hidden layer and the *body* and *subject* sections considered for the smallest collection.

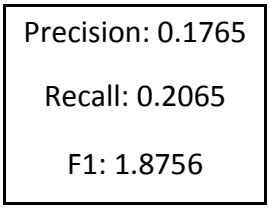

Figure 22 Neural network – result with one hidden layer

Learning without hidden layers and with 100 cycles and with Manhattan propagation gave slightly better results, see Figure 24.

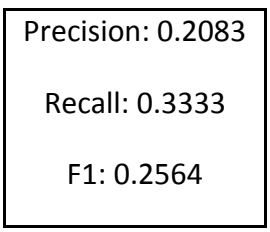

Figure 23 Neural network – result with Manhattan propagation

Next tests use the bigger collections and the Manhattan propagation without hidden layers, with the body and subject sections considered. This setup of the first test showed better results .

Figure 25 depicts results of learning on the medium training set. As it can be seen the results are getting better with more terms used for training.

Precision: 0.4536
Recall: 0.4932
F1: 0.4725

Figure 24 Neural network – result of the medium collection

Figure 26 presents results of learning with the biggest data collections. They are significantly better than the results obtained with the smaller collections.

Precision: 0.6454
Recall: 0.6845
F1: 0.6644

Figure 25 Neural network – result of the biggest collection

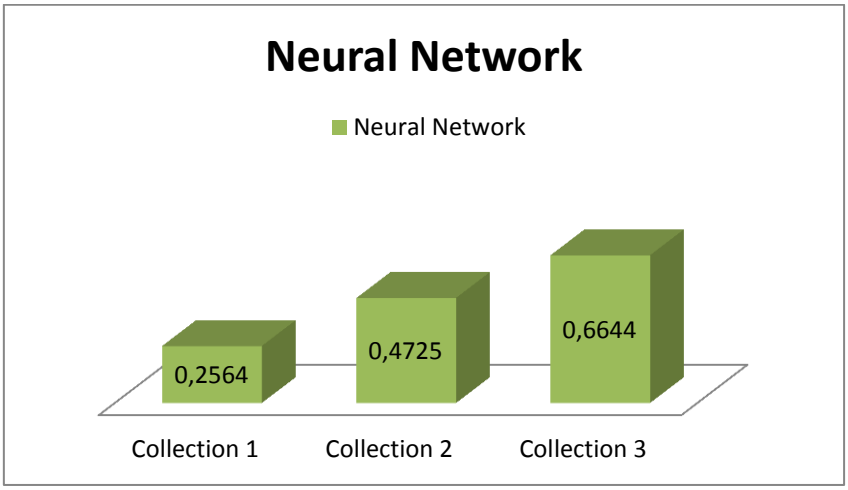

Figure 26 Neural Network - comparision of result

The tests clearly show that the more terms is used for learning the better the F1-measure achieved is. As all collection were taught using the backpropagation algorithm the results can be compared on a common plane. Only the first result used optimization but it did not bring the expected results. This could happen because of too high or too low dimensionality of categories in the training set.

7.4. Naive Bayes

Naive Bayes classifier turned out to be of good precision as well. Among all individual classifiers it won the second location.

Figure 27 presents results of Naive Bayes classifier trained with the smallest collection. The results achieved in this test were a very low level. This was because of the amount of documents in the training set.

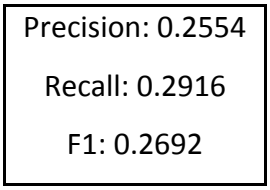

Figure 27 Naive Bayes – result of the smallest collection

Figure 28 depicts results of Naive Bayes classifier trained with the medium collection. The result of F1-measure equal to 0.4005 reached with the medium collection showed a significant improvement.

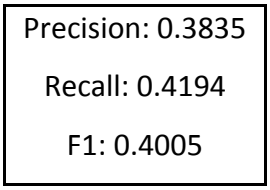

Figure 28 Naive Bayes – result of the medium collection

Figure 29 presents results of Naive Bayes trained with the biggest collection. When the number of documents in the collection increased the results obtained achieved a satisfactory level. Each collection was trained in the same way so the results could be compared.

Precision: 0.6358

Recall: 0.6607

F1: 0.6479

Figure 29 Naïve Bayes – result of the biggest collection

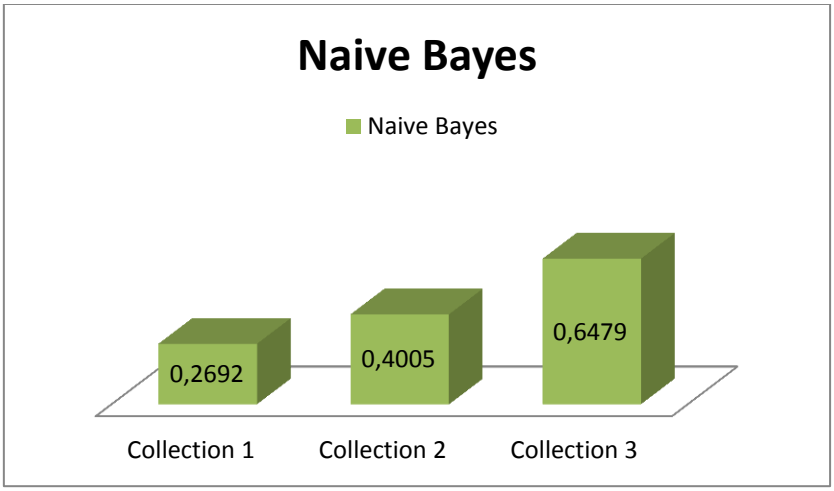

Figure 30 Naive Bayes - comparision of results

7.5. Support Vector Machine

Figure 30 presents results of SVM trained with the smallest collection.

Precision: 0.1365

Recall: 0.1533

F1: 0.1466

Figure 31 SVM – result of the smallest collection

Figure 31 depicts results of SVN trained with the medium collection.

Precision: 0.2269

Recall: 0.2569

F1: 0.2408

Figure 31 SVM – result of the medium collection

Figure 32 depicts results of SVN trained with the biggest collection.

Precision: 0.3108
Recall: 0.3634
F1: 0.3349

Figure 33 SVM– result of the biggest collection

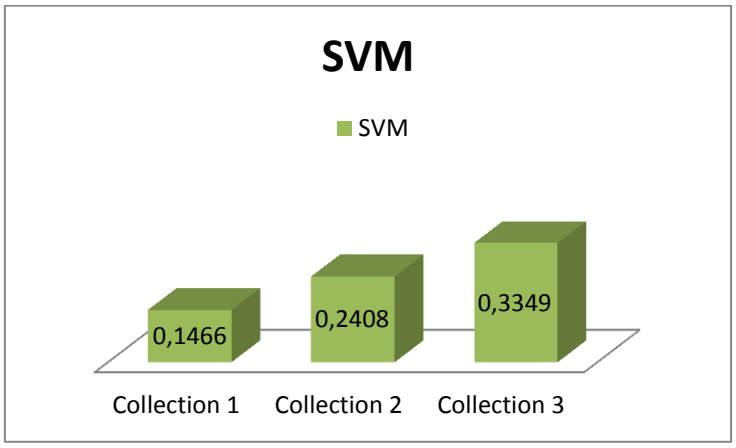

Figure 32 SVM - comparision of results

7.6. Winnow

Figure 33 depicts results of Winnow classifier trained with the smallest collection

Precision: 0.3636
Recall: 0.2598
F1: 0.2962

Figure 34 Winnow – result of the smallest collection

Figure 34 depicts results of Winnow classifier trained with the medium collection.

Precision: 0.4365
Recall: 0.4458
F1: 0.4415

Figure 33 Winnow – result of the medium collection

Figure 35 depicts results of Winnow classifier trained with the biggest collection.

Precision: 0.6164

Recall: 0.6428

F1: 0.6292

Figure 34 Winnow – result of the biggest collection

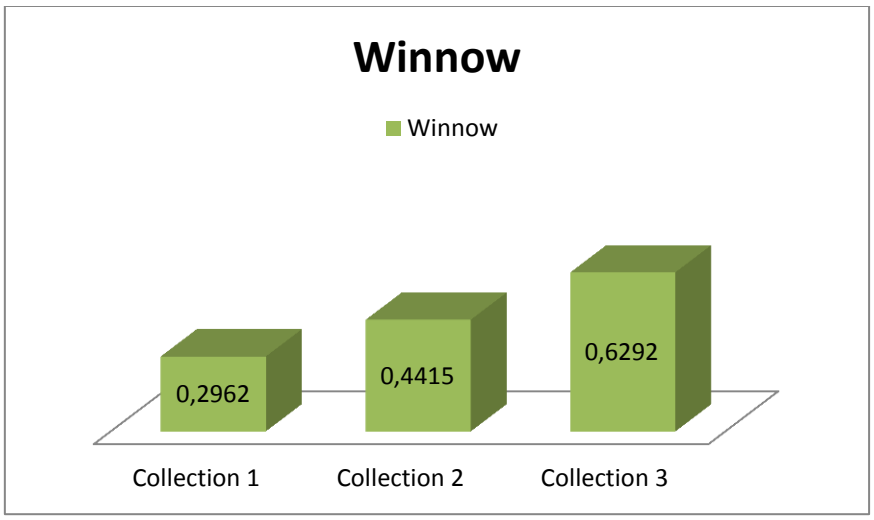

Figure 35 Winnow - comparision of results

7.7. Voting

Formula (7.1) is used to calculate improvment of the combined results.

$$\text{Improvement} = \frac{F1(\text{combination}) - F1(\text{single best classifier})}{F1(\text{single best classifier})} \qquad (7.1)$$

This measure is applied to the voting and the FuzzyMax classifiers.

Figure 36 shows results of Voting trained with the smallest collection.

Precision: 0.3333

Recall: 0.4166

F1: 0.3703

Improvement: 6.29%

Figure 36 Voting – result of the smallest collection

Figure 37 shows results of Voting trained with the medium collection.

Precision: 0.4765

Recall: 0.5165

F1: 0.4956

Improvement: 5.04%

Figure 37 Voting – result of the medium collection

Figure 38 shows results of Voting trained with the biggest collection.

Precision: 0.6632

Recall: 0.7083

F1: 0.6849

Improvement: 3.15%

Figure 38 Voting – result of the biggest collection

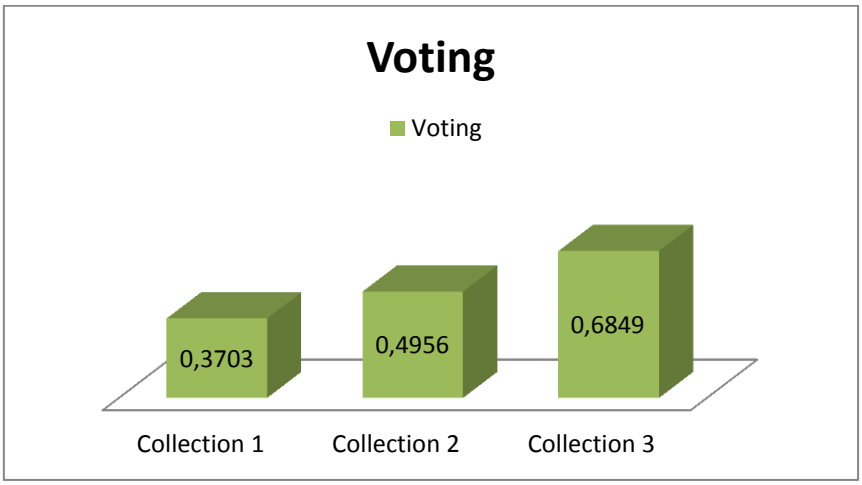

Figure 39 Voting - comparision of results

7.8. Fuzzy Max

Figure 39 depicts results of Fuzzy Max trained with the smallest collection

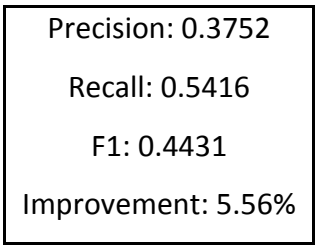

Figure 40 FuzzyMax – result of the smallest collection

Figure 40 depicts results of Fuzzy Max trained with the medium collection

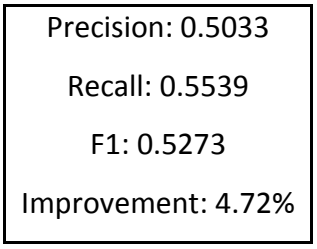

Figure 41 FuzzyMax – result of the medium collection

Figure 41 depicts results of Fuzzy Max trained with the biggest collection.

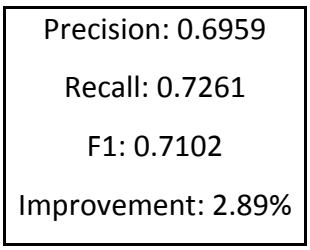

Figure 42 FuzzyMax – result of the biggest collection

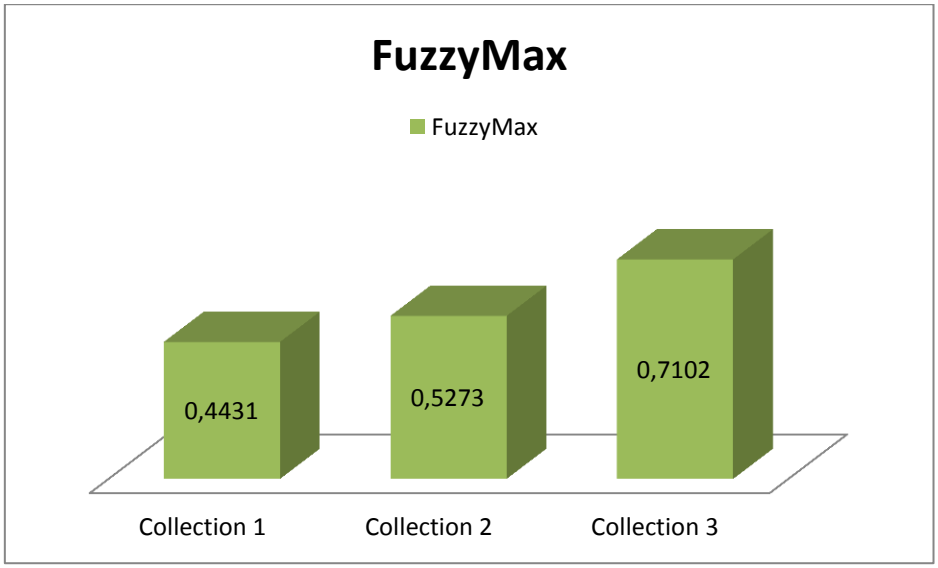

Figure 43 Fuzzy Max - comparision of results

Figere 44 depicts all single classifiers and their combination.

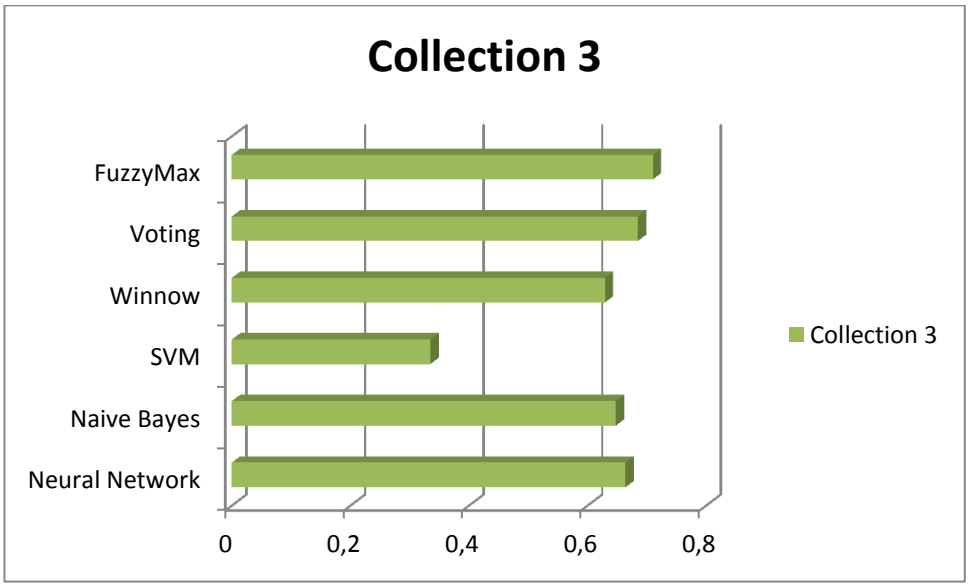

Figure 44 All results of the biggest collection

CHAPTER 8 – CONCLUSIONS AND FUTURE WORK

Algorithms of machine learning have been discussed and implemented in the thesis. The results obtained from testing classifiers were in most cases very good. Three collections of data were used in this work. All of them came from the Enron Corpus. Collections varied in terms of number of emails included in the categories. The structure of each set was similar. Each set included six persons with 2 to 5 categories which resulted in total 25 categories used for learning. The first corps included a 417 e-mails, the one included 4104 e-mails, and the third one included 8241 e-mails.

The test sets were organized in the same manner as the training sets. They contained 25 categories, but the number of e-mails was much smaller i.e. 168. In order to maintain coherence each classifier was tested with the same test set.

The training classifiers gave the best results if preprocessing was used, i.e.: removing stopwords, white space, digits, putting every letters to the lowercase, and stemming. Also, the testing phase involved implementation of these procedures for each test e-mail. All these activities resulted in significant reduction in dimensionality, which in turn resulted in reduction of the training time and increase of the quality of the classification.

The quality of the algorithms can be ranked as follows :

$$FuzzyMax > Voting > NN > NB > Winnow > SVN$$

As the number of e-mails in the corpuses of data increased the quality of these classifiers increased proportionally. Studies carried out in this study clearly demonstrate that a single classifier despite its precision is not able to achieve better results than a combination thereof.

The best results were obtained by the FuzzyMax classifier. The F1-measure for the smallest collection reached 0.4432, 0.5273 for the medium one and 0.7102 for the largest one. The improvement was at the level of 0.045 relating to the best individual classifier which was artificial neural network.

The second in terms of quality was the Voting classifier. The F1-measures were 0.6849, 0.4956 and 0.3703 for the largest, medium and the smallest collection respectively.

As far as individual classifiers are concern, the best results were obtained by Artificial Neural Networks. Chapter 5 discusses learning options available for this classifier . Better results were achieved when no hidden layers were applied and the best result in terms if F1-measure obtained for the largest collection and it was 0.6644.

The Naive Bayes classifier and the Winnow algorithm gave very similar results for each collection. For the largest one and at the same time the most accurate, the results of F1-measure were at the level of 0.63. For the smallest collection, they were at the level of 0.27, and 0.42 for the medium collection.

The worst algorithm in terms of F1-measure, with the best score at the level of 0.3349 achieved for the largest collection, turned out to be the SVM algorithm. Very poor performance in relation to the remaining classifiers can be due to the training process which was one class vs. all others. F1-measure obtained for the smallest collection did not exceed the level of 15%.

Algorithms of Machine Learning presented in the thesis manifested great potential for automatic categorization of e-mails. Despite the fact that the research ended up at this point, still there are many features which may be added in further development of the project:

- Implementation of the KNN algorithm
- Implementation of additional measures of the classifiers quality
- Acceptance of e-mails in different formats
- Handling different types of databases
- Implementation of other algorithms for features selection
- Ability of assigning a single e-mail to multiple categories
- Ability of reading attachments

BIBLOIGRAPHY

[1] William W. Cohen, MLD, CMU 2009, https://www.cs.cmu.edu/~enron/

[2] Maurice de Kunder, 2013, http://www.worldwidewebsize.com/

[3] D'Orazio, Dante (28 June 2012). "Gmail now has 425 million total users". The Verge. Vox Media. Retrieved on June 28th, 2012

[4] Vascellaro, Jessica (Oct. 31, 2012). "Gmail finally beats Hotmail, according to third-party data". Gigaom.com. Retrieved on October 3rd , 2011

[5] Witten, I. H., and Frank, E. Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufmann Publishers Inc., 2000.

[6] Dietterich, T. Machine-learning research: Four current directions. The AI Magazine 18, 4 (1998), 97–136.

[7] I.Kononenko, I.Bratko, and M.Kukar. Application of machine learning to medical diagnosis. In Machine Learning, Data Mining and Knowledge Discovery: Methods and Applications (1998), John Wiley & Sons.

[8] Page, G. F. Introduction to machine learning, by ethem alpaydin, mit press, 2004, xxx + 415 pp., with index. 160 ref. distributed chapter by chapter. Isbn 0-262-01211-1. (hardback £32.95). *Robotica 24*, 1 (2006), 143–143.

[9] Turban, E., Aronson, J. E., Liang, T.-P., and Sharda, R. *Decision*

Support and Business Intelligence Systems, eighth ed. Pearson, Upper Saddle River, New Jersey, 2007.

[10]

http://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/wipo-alpha-readme.html [11] Kevin K Dobbin and Richard M

Simon, - Optimally splitting cases for training and testing high dimensional classifiers

[12] http://www.cs.cmu.edu/~wcohen/

[13] Ron Bekkerman ronb@cs.umass.edu, Andrew McCallum mccallum@cs.umass.edu, Gary Huang ghuang@cs.umass.edu – "Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora"

[14] T. Zhang, 'Large Margin Winnow Methods for Text Categorization', *KDD-2000 Workshop on Text Mining*, 2000.

[15] Basu A., Watters C., Shepherd M., Support Vector Machines for Text Categorization, Proceedings of the 36th Hawaii International Conference on System Sciences, 2003.

[16] Thorsten J., Text Categorization with Support Vector Machines: Learning with Many Relevant Features, In: Proceedings of ECML-98, 10th European Conference on Machine Learning, edited by Claire Nédellec and Céline Rouveirol, pp 137-142. Springer-Verlag, 1998.

[17] Gärtner T., A survey of kernels for structured data, ACM SIGKDD Explorations Newsletter archive, Volume 5, Issue 1 (July 2003).

[18] Hammer B., Villmann T., Tutorial: Classication using non-standard metrics, in: M. Verleysen (ed.), ESANN'2005, to appear.

[19] L. Xu, A. Krzyzak and C.Y. Suen, 'Methods of combining multiple classifiers and their applications to handwriting recognition', *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 22. No. 3, May/June 1992.

[20] P.N. Bennett, S. T. Dumais and E. Horvitz, 'Probabilistic Combination of Text Classifiers Using Reliability Indicators: Models and Result', *In*

proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'02), Tampere, Finland, 2002.

[21] Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classier. In Fifth Annual Workshop on Computational Learning Theory, pages 144{152, 1992.

[22] TomM.Mitchel, McGrawHil, lectures lides for textbok Machine Learning, 1997

[23] Abraham Othman, Tuomas Sandholm, Decision Rules and Decision Markets, 1998

[24] Yiming Yang and Xin A re-examination of text categorization method, 1998

[25] Fabrizio Sebastiani, Machine Learning in Automated Text Categorization, March 2002

[26] M. Kubat, M. Cooperson, Jr.: Voting nearest neighbor subclassifiers, In: Proc. of the 17th Intl. Conference on Machine Learning, 503–510, 2000

[27] L.K. Hansen, P. Salomon: Neural network ensembles, IEEE Trans. on Pattern Analysis and Machine Intelligence, 12:993–1001, 1990.

[28] O. Matan: On voting ensembles of classifiers, In: Proc. of the 13th Natl. Conference on Artificial Intelligence, 84–88, 1996.

[29] Sample Compression, Margins and Generalization: Extensions to the Set Covering Machine Mohak Shah

[30] N. Littlestone, 'Learning Quickly when Irrelevant Attributes Abound: A New Threshold Algorithm', *Machine Learning*, 2, pp. 285-318.

[31] Ron Bekkerman, Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRICorpora, 2004

[32] Irena Koprinska, Josiah Poon, James Clark, Jason Chan Learning to classify e-mail, 2006

[33] Bryan Klimt and Yiming Yang Email Folder Classification using Threads, Language Technologies Institute Carnegie Mellon University Pittsburgh, PA 15213

[34] Yiming Yang and Xin Liu A re-examination methods of text categorization methods

[35] Henrik Mathiassenand Daniel Ortiz-Arroyo, Automatic Categorization of Patent Applications Using Classifier Combinations

[36] V. Vapnic The Nature of Statistical Learning Theory, Springer, New York, 1995

[37] C. Cortes and V. Vapnik. Support Vector Machine. Machine Learning, 1995

[38] Osuna, R. Freud, and F. Giros, Support vector machines, Training and Applications. In A.I Memo. MIT A.I. Lab, 1996

[39] Manu Aery and Sharma Chakravarthy, IT Laboratory and CSE Department eMailSift: Email Classification Based on Structure and Content

[40] W. W. Cohen. Learning rules that classify e-mail. Proceedings of AAAI-1996 Spring Symposium on Machine Learning Information Access, pages 124–143, 1996

[Q] Sebastiani F.: *A Tutorial on Automated Text Categorisation*, Proceedings of ASAI-99,1st Argentinian Symposium on Artificial Intelligence, Buenos Aires, Argentyna, 1999, s. 7–35

[W] Aas K., Eikvil L.: *Text categorization: A survey*, Raport techniczny, Norwegian Computing Center, 1999