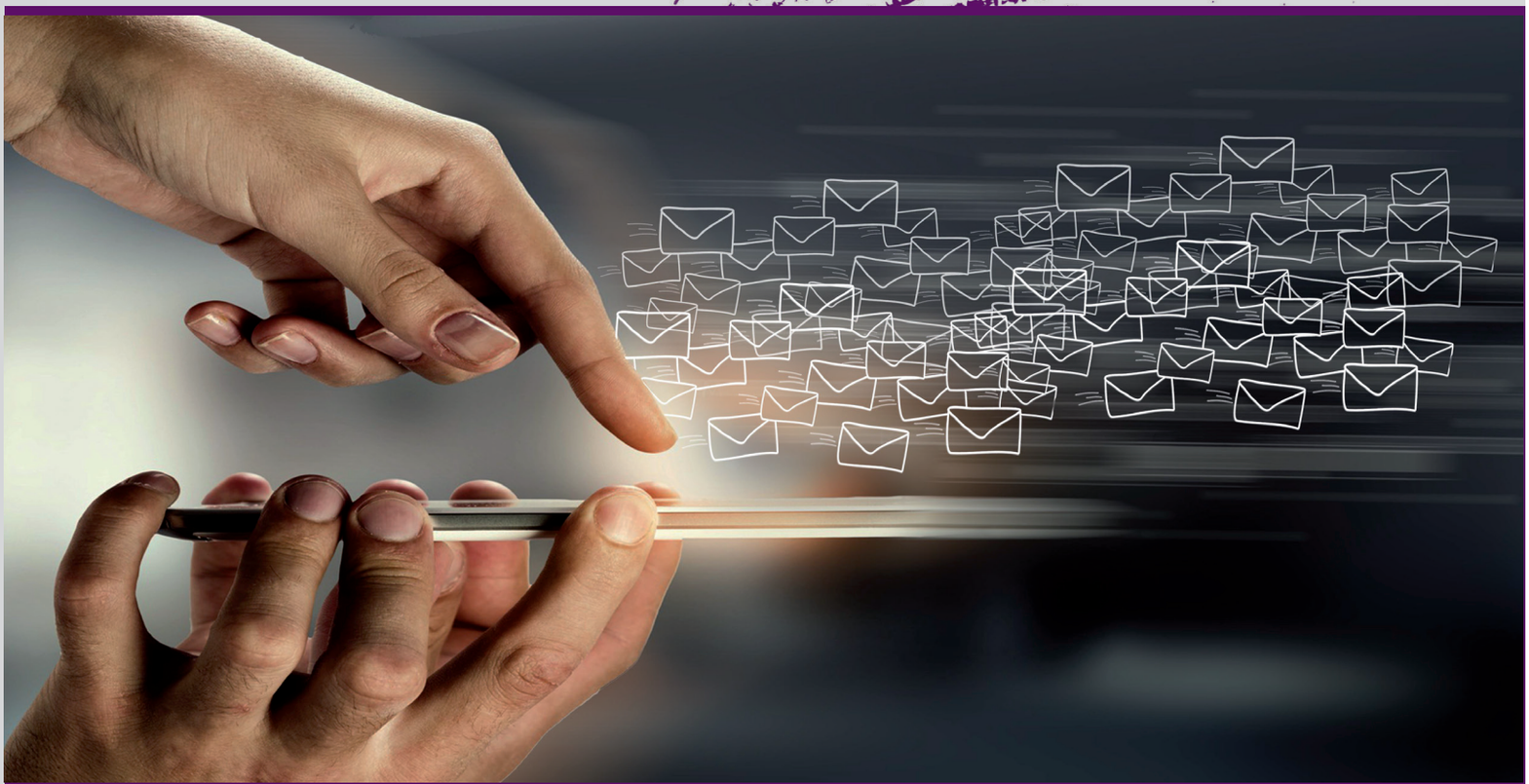
Mondragon Unibertsitatea

DOCTORAL THESIS

DIMENSIONALITY REDUCTION FOR THE IMPROVEMENT OF ANTI-SPAM FILTERS



IÑAKI VÉLEZ DE MENDIZABAL GONZÁLEZ | Arrasate-Mondragón, 2022

IÑAKI VÉLEZ DE MENDIZABAL GONZÁLEZ | Dimensionality reduction for the improvement of anti-spam filters

THESIS

Dimensionality reduction for the improvement of anti-spam filters

Author:
Iñaki VÉLEZ DE MENDIZABAL GONZÁLEZ

Supervisors:
PhD. Urko ZURUTUZA ORTEGA
PhD. Enaitz EZPELETA GALLASTEGI
PhD. Vitor BASTO FERNANDES

PhD Program in Applied Engineering
Electronics and Computing Department
Faculty of Engineering
Mondragon Unibertsitatea

Arrasate
July 2022

”Familia eta lagun guztiei.”
Mila esker

A Maribel, a Maialen y a Unai.
Iñaki

Acknowledgments

Susmoa neukan horrelako bidaiak ez zirela bakarrik egiten eta orain guztiz argi daukat. Lan hau nire inguruan egon zareten guztion fruitua da. Eskerrik asko tunelaren barruan argi apur bat egin duzuenoi, milesker motxila honen pisua eramaten lagundu didazuenoi eta nola ez, eskerrak bihotzez etengabe aurrerantz bultza egin didazuenoi.

Hoien artean bereziki denbora guztian nire ondoan egon zareten **Urko Zurutuza** eta **Enaitz Ezpeleta**. Momentu txarretan baikor, beldur momentutan ausart, zalantza momentutan sendo eta denbora guztian lagun. Zoragarria izan da bide hau zuekin batera egitea; edo hobeto esanda, zuen atzetik egitea, beti ibili zarete eta bi pauso nire aurretik. Ezin duzue imaginatu zer nolako lasaitasuna ematen duen horrelako zuzendariak edukitzea. Obrigadisimo **Vitor Manuel Basto Fernandes** por sua ajuda. Thank you for receiving me in Lisbon and for helping me in ISCTE-IUL.

Gracias también a **J. R. Méndez, Moncho**. O meu anxo da garda. Compañero desde hace tantos años en este camino que parecía no terminar nunca. A todo el **grupo SING de la Universidade de Vigo** que estais en Ourense, graciñas.

Eskerrik asko **Mondragon Goi Eskola Politeknikoko lankide** guztiei bidai hau egiteko aukera eta medioak eskeintzeagatik. **Datuen Analisia eta Zibersegurtasun taldeko guztioi** bereziki besarkada handi bat. Orain faborea bueltatzeko nire txanda da. Ikerketa honetako experimentuak egitea posible egin duzun **Antton Rodriguez** eta azken metroak egiten lagundu didazuen **Ekhi Zugasti**, **Xabier Vidriales** eta **Iban Barrutia**, zuei ere eskerrak bihotz bihotzez. **Amigos do ISTAR ISCTE-IUL**, levar-vos-ei sempre comigo.

Eta gertukoenei, aita eta ama, zaudeteneko tokian, pena bat izan da ezin lehenago amaitzea zuekin ospatzeko. **Maribel**, **Maialen** eta **Unai**, eskerrik asko zuen pazientziagatik, laguntzagatik eta oraindik zer den ulertzen ez duzuen honetan laguntzeagatik.

Zuek guztiok gabe hau ezinezkoa izango zen.

Originality Statement

I declare that I am the sole author of this work. This is a true copy of the original document, including any revision which may have been ordered by my examiners. I understand that my work may be available to the public, either in the school library or in electronic format.

Iñaki Vélez de Mendizabal González
Arrasate, July 2022

Abstract

Nowadays, spam represents more than 45% of the world's email traffic. Filtering techniques to combat the problem of spam distribution have been the subject of many research studies in recent years. Several combinations of legal, administrative and technical perspectives were tested. The combination of technical approaches, namely, the widely exploited content-based and token-based filtering techniques, revealed low significance improvements on spam classification performance. Due to the limited performance of token-based strategies, new knowledge representation schemes (such as those based on word-embeddings, topics, or synsets) have been developed. The use of synsets to represent the meaning of the words guides the community towards the identification of the intentionality of a message, allowing the classification of messages that want to sell products, obtain information about us, etc. The advantage of this kind of synsets representations lies on the capability to taxonomically group concepts, handling the polysemy and synonymy. These properties have been successfully exploited in this research work to design a novel Machine Learning (ML) based lossless feature reduction schemes by grouping concepts strategies. This type of reduction schemes has achieved a reduction in the classification problem dimensionality (number of features), improving the classification performance. In a second step we introduce and demonstrate the effectiveness of a new feature reduction scheme that combines the strengths of lossless and lossy strategies. Finally, in order to use the Leetspeak encrypted words, a decoder has been designed and tested. The proposed system reduces the number of unprocessed words considerably, improving the classification rates of spam messages.

key words:
Spam, Synset-based representation, Semantic information, Multi-objective evolutionary algorithms, Leetspeak, deobfuscation

Laburpena

Gaur egun spam mezuek mundu osoko email trafiko globalaren %45-a suposatzen dute. Azken urteetan spam-aren arazoa konpontzeko tekniketan ikerketa ugari egin dira. Soluzio desberdinak probatu dira alderdi legalak, administratiboak eta teknikoak nahastuz. Ikuspuntu tekniko batetik edukietan eta token-etan oinarrituriko teknikek hobekuntza eskasak lortu dituzte. Azken hauek lortutako emaitzak hobetzeko, mezuen barruko informazioa errepresentatzeko era berriak garatu dira (adierazpen bektoriala, gaiak edo *synset*-ak). Hitzen esanahiak erabiltzeak mezua zein asmorekin idatzia izan den asmatzera bideratzen gaitu, produktuak saldu nahi dituen mezu bat bezala klasifikatuz, informazioa lortu nahi duen mezu bat bezala, etabar. Informazioa errepresentatzeko metodo berri hauek kontzeptuak elkartzeko gaitasuna daukate, esanahi desberdineko hitzak eta esanahi bereko hitzak taxonomikoki azteretuz. Propietate hauetan oinarrituz, ikerketa lan honetan, informazio galera gabeko ezaugarri kopuru murrizketa lortzen duen sistema bat garatu da, zein Ikaste Automatikoan oinarritzen den kontzeptuak elkartzeko. Honi esker arazoaren dimentsioa (tamaina) gutxitu da mezuen sailkapenaren errendimendua hobetuz. Bestalde, garaturiko lan honen abantailetan oinarritzen den bigarren sistema bat ere garatu da, non informazio galera gabeko sistemaren sendotasuna, informazio galera txiki batekin konbinatzen den. Amaitzeko *Leetspeak*-ean kodifikaturiko hitzen informazioa berreskuratzeko dekodifikatzaile bat garatu da. Garaturiko dekodifikatzaileak berreskuratzen dituen hitzen informazioari esker, klasifikazioaren emaitzak hobetu egiten dira.

Resumen

Actualmente el spam representa cerca del 45% del trafico mundial de emails. En los últimos años las técnicas de filtrado para combatir el spam han sido objeto de innumerables estudios. Se han probado distintas soluciones combinando aspectos legales, administrativos y técnicos. Desde el punto de vista técnico, la combinación de técnicas de filtrado basadas en tokens y técnicas de filtrado basadas en contenidos han traído mejoras poco significativas en las tasas de clasificación del spam. Debido a las limitadas mejoras conseguidas con estas estrategias, se han desarrollado nuevos esquemas de representación del conocimiento (como las representaciones vectoriales, temas o *synsets*). El usar *synsets* para representar el significado de las palabras nos guía hacia la identificación de la intencionalidad de un mensaje, permitiendo clasificarlos como mensajes que quieren vender productos, obtener información sobre nosotros, etc. La ventaja de este tipo de representaciones está en su capacidad de agrupar taxonómicamente los conceptos, resolviendo la polisemia y la sinonímia. Estas propiedades han sido utilizadas con éxito en este trabajo de investigación, para diseñar un nuevo esquema de reducción de características sin pérdida de información mediante agrupaciones de conceptos basado en técnicas de Aprendizaje Automático. Gracias a este esquema de reducción, se ha conseguido reducir la dimensionalidad del problema de clasificación (número de características), mejorando el rendimiento. En un segundo paso, presentamos y demostramos la eficacia de un nuevo esquema de reducción de características que combina los puntos fuertes de la estrategia sin pérdida de información combinándola con una leve pérdida de información. Por último, para recuperar la información de las palabras cifradas mediante *Leetspeak*, se ha diseñado y probado un decodificador. El sistema presentado reduce considerablemente el número de palabras cifradas (ofuscadas) que se quedan sin procesar, mejorando los índices de clasificación de los mensajes de spam.

Contents

List of Figures

List of Tables

Acronyms

ANN Artificial Neural Network 22

AOI Attribute Oriented Induction 16

CNN Convolutional Neural Network 5

DIMr Dimensionality Reduction ratio 51

DL Deep Learning 58

FP False Positive 4

FPr False Positive ratio 33

FN False Negative 4

FNr False Negative ratio 33

GA Genetic Algorithm 20

IG Information Gain xv

IM Instant Messaging 7

IP Internet Protocol 8

IR Information Retrieval 19

ML Machine Learning vii

MOEA Multi-Objective Evolutionary Algorithms 30

NLP Natural Language Processing 2

NSGA-II Non-Dominated Sorting Genetic Algorithm 30

OCR Optical character recognition 21

OSN Online Social Network 7

PCA Principal Component Analysis 15

POS Part Of Speech xv

SMS Short Message Service 7

SVM Support Vector Machines 15

TCR Total Cost Ratio 40

TREC Text Retrieval Conference 7

UCI University of California, Irvine 35

URL Uniform Resource Locator 2

WSD Word Sense Disambiguation 19

XAI Explainable Artificial Intelligence 29

Chapter 1

Introduction

"*Spam is an irrelevant or unsolicited message sent typically to a large number of users, for the purposes of advertising, phishing, spreading malware, etc.*" - Oxford Dictionaries

This chapter describes the problem that this research addresses. The main objective of the research is described, as well as the technical objectives to that guide the achievement of the research statements. It also presents the starting hypothesis, the work developed to accomplish objectives and finally, the contributions of this work, enumerating the publications that have been produced.

To better understand the problem that needs to be solved, it is necessary to take into account that in just a few years, the Internet has changed the way people communicate, get information and do business, transforming economic and social interactions and relations. From 1,1 billions connected users in 2005 to 4,950 billions in 2022[1]. This number of connected users has increased especially as a result of the use of smartphones with an Internet connection. Gartner already reported in 2013 that the sale of smartphones surpassed[2] sales of feature phones. Many of these users use the Internet legitimately and take advantage of its benefits. However, there are other kinds of users which use the Internet for their own benefit, such as spammers, delivering their content through the Instant messaging [17, 92] services, email [19, 96] and social networks [19, 113].

Many technologies such as collaborative solutions [100], content-based schemes [8, 52, 68, 109] and even network standards, like Request for Comments 63761[3] or 72082[4], have been developed to combat spam, but it has not been completely

[1] https://www.statista.com/statistics/617136/digital-population-worldwide/

[2] Available at https://www.gartner.com/en/newsroom/press-releases/2014-02-13-gartner-says-annual-smartphone-sales-surpassed-sales-of-feature-phones-for-the-first-time-in-2013

[3] Available at https://tools.ietf.org/html/rfc6376

[4] Available at https://tools.ietf.org/html/rfc7208

eliminated. The European Parliament has addressed the problem from a legal point of view with the European Directive[5] on Privacy and Electronic Communications, but this has not solved the problem either.

In the work of Bhuiyan et. al. [14], a review of anti-spam systems, authors describe their evolution from the most primitive ones that consist in a simple filter to identify the sender's address to lock it, passing through the first content filters that discarded messages that had a specific word in the subject, up to the most recent ones, on which this research work is going to focus, based on ML and Natural Language Processing (NLP).

ML and NLP techniques have been widely used to classify spam messages and have demonstrated high classification performance[25]. In [74] a graphical summary of the different used techniques is presented, showing the results obtained in terms of accuracy, false positive and false negative ratios. In order to be processed by these ML algorithms, texts have to be represented in a specific way. This form of representation is key to reduce the computational requirements to run the classifier as well as to achieve high performance classification ratios. In [74] we can observe that the classification ratio is more than 70% in all cases, revealing that this techniques have high performance.

Despite all these efforts, spam is still present in about 45% of the messages on the Internet[6]. Spam not only has invaded email, it has also invaded social media, which is used by many users to keep in touch with their friends and family, companies to engage potential customers, and other many cases. With a big amount of users ready to access the contents shared or posted by their contacts, it is not surprising that social media is a usual target for spammers. Moreover, spam has changed from being simply inconvenient to become a cyber threat. Spam messages may include malicious Uniform Resource Locator (URL)s that can redirect the user to malware download pages or phishing sites.

1.1 Research statement

This research work aims to address the spam problem from a novel approach. Currently, most anti-spam filters are based on statistical classifiers that compute the probability of a message to belong to spam or not based on the words in the message. To make this estimation, the classifier has to be previously trained with a set of legitimate messages. This thesis has the goal of extracting the meaning of the messages received, and react

[5] Available at https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32002L0058
[6] Available at https://securelist.com/spam-and-phishing-in-q3-2021/104741/

upon it. Using word meanings and semantic dictionaries (free means different in "free people" and "free drugs") this work improves the identification of message types (ham or spam).

The use of the meaning of the words guides the community towards the identification of the intentionality of a message, or the intentionality of the author of the message, allowing the classification of messages that want to sell products, messages that want to obtain information about us, etc. In order to work in this direction, we first dissect the texts making use of semantic dictionaries, disambiguation and grouping processes, to classify messages with reference to a set of defined categories based on the use of semantic dictionaries, disambiguation and text generalization processes to classify messages with reference to a set of defined categories.

1.1.1 Research objectives and hypothesis

This thesis has one research hypothesis and three objectives, which are described next:

- **Hypothesis**: The semantic processing of messages towards intentionality detection, by the means of semantic dictionaries, allows the improvement of spam messages classification performance.

- **Objective 1**: *Perform a reduction in the number of tokens in a message by the means of semantic generalization and message meaning simplification.*

  As an example, it will make it possible to join the tokens "Viagra", "Cialis" and "Tadalafil" into a single token as "anti_impotence_drug". This will speed up the training process and reduce the memory requirements for classification, increasing the performance.

- **Objective 2**: *Identification and the reduction of irrelevant tokens to improve the classifier performance and computing time.*

  Therefore we want to focus on the core of the message, that would leverage the intention. There are stop words, which are words that do not provide information (such as "at", "in", "the") to separate legitimate messages from spam messages. This may also be the case with the meaning, in which case they could be removed.

- **Objective 3**: *Identification and decoding of obfuscated words, to correct message content and enhance syntactic and semantic analysis.*

  Obfuscated words such as "g00d" do not exist in semantic dictionaries and consequently must be deobfuscated before searching for their meaning.

1.2 Contributions

The main contributions of this thesis and the corresponding scientific publications are described as follows:

- A system that classifies messages using an approach based on the semantics of words has been developed. The meaning of the word is contextualized using the Babelfy [78] services and then semantic relations are extracted and analyzed using Babelnet [80, 83] semantic network.

- The time and computational resources for the classifier training phase have been reduced (the number of tokens has been reduced more than four times with very close classification results). This has been achieved by reducing the number of redundant words by grouping taxonomically related words. Moreover, from a theoretical point of view, when combining highly dependent features, intermediate features are reduced, which increases the performance of techniques such as Naïve Bayes.

- Similar meaning words clustering, leads to identification of the topic of a message. This provides the background for the intentionality identification in terms of the subject of the message. It has been demonstrated that texts about "anti-impotence drugs" could be connected in the same group, enabling the identification of the subject of various spam-related messages.

- The experiments performed allow to conclude that the use of a synset[7] representation reduces the number of False Positive (FP) errors while lead to a slight increase of False Negative (FN) errors (see the results in Chapter 4). This shows that ham contents, usually without obfuscation and spelling errors, allow a higher number of words to be successfully translated into synsets and a better ranking in this type of instance. In contrast, most of the words included in spam contents (with many misspelled words, obfuscated tokens or URLs) cannot be successfully represented in synset-based representations, resulting in lower information collection for this type of texts, as well as lower classification performance.

- Three different feature vector element reduction (dimensionality reduction) strategies have been formulated and tested to be used when texts are represented using synsets. The first one involves an information lossless strategy, the second one involves a low loss of information and the third one involves loss of information.

[7] Is a group of data elements that are considered semantically equivalent for the purposes of information retrieval.

The first two strategies have been experimentally tested and the third one has only been developed and analyzed on a theoretical level. The results obtained allow us to conclude that lossless feature reduction schemes can be successfully complemented with low loss approaches for the identification and removal of irrelevant or noisy features, in order to reduce the computational costs of classification.

- A system based on Convolutional Neural Network (CNN)s has been developed to identify and decode Leetspeak obfuscated characters, allowing the recovery of tokens that were previously unusable for classification purposes. At the same time and to carry out this research work, several datasets have been developed (an image database to train the CNN and four datasets for evaluating Leetspeak decoding processes) and are now publicly available.

1.3 Publications

The following is a list of national and international conferences and journals in which sections of this thesis have been published.

JCR journals

- de Mendizabal, I. V., Vidriales, X., Basto-Fernandes, V., Ezpeleta, E., Mendez, J. R., Zurutuza, U. (2022). Deobfuscating *Leetspeak* with Deep Learning to Improve Spam Filtering. International Journal of Interactive Multimedia and Artificial Intelligence. ***[In Review]***

- de Mendizabal, I. V., Basto-Fernandes, V., Ezpeleta, E., Mendez, J. R., Gómez-Meire, S., Zurutuza, U. (2022).Multiobjective Evolutionary Optimization for Dimensionality Reduction of Texts Represented by Synsets. Knowledge and Information Systems. ***[In Review]***

- de Mendizabal, I. V., Basto-Fernandes, V., Ezpeleta, E., Mendez, J. R., Zurutuza, U. (2020). SDRS: A new lossless dimensionality reduction for text corpora. Information Processing & Management, 57(4), 102249. ***[In Press]***

Conference papers

- de Mendizabal, I. V., Ezpeleta, E., Zurutuza, U. (2021). Reducción de dimensionalidad sin pérdida en representaciones semánticas de texto. JNIC 2021. VI Jornadas Nacionales de Investigación en Ciberseguridad. Ciudad Real, Spain. 9-10 June. ***[In Press]***

- de Mendizabal, I. V., Ezpeleta, E., Ortega, U. Z., Ordás, D. R. (2018). La intención hace el agravio: técnicas de clustering conceptual para la generalización y especialización de intencionalidades en el spear phishing. In Actas de las Cuartas Jornadas Nacionales de Investigación en Ciberseguridad (pp. 41-42). Mondragon Unibertsitatea. ***[In Press]***

1.4 Document Structure

The rest of this thesis document is organised as follows:

First, in Chapter 2, the technical background of thesis-related topics are described to introduce the reader the concepts and terminology used throughout the document. Next, in Chapter 3 the state of the art in the spam filtering is presented.

In Chapter 4, the first contribution is presented. The new proposal for reducing the number of features by clustering words/tokens is described. Next a new proposal for feature reduction with low-loss of information as a result of the elimination of low significance words/tokens is presented in Chapter 5. In Chapter 6 the problem of text obfuscation in spam messages is discussed, as well as its effect in syntactical and semantic text processing and classification. A system for solving this problem using neural network-based computer vision is presented.

Finally, Chapter 7 synthesises the contributions of this thesis and describes possible future work.

Chapter 2

Technical Background

The aim of this chapter is to present the technical background in the spam filtering domain and to introduce the reader to the core concepts and terminology used throughout the document. The chapter consists of four sections. The first section introduces the types of spam and the threats they represent. The second section introduces spam filtering techniques, feature extraction and representation. The third section addresses the feature dimensionality problem and dimensionality reduction strategies. As a consequence of this reduction, the most relevant words, that can help to determine the intentionality of the message, are identified and used for classification purposes. Finally, the problems resulting from obfuscated words embedded in text messages is described.

2.1 Spam

One of the earliest definitions of e-mail spam is provided by the Text Retrieval Conference (TREC)[1]. Spam was defined as: "*Unsolicited and unwanted e-mail that was sent indiscriminately, directly or indirectly, by a sender who has no current relationship with the recipient*".

This definition is still valid today, but needs to be adapted to new media, new platforms and Online Social Network (OSN). Nowadays spam can come not only by email, it can also come by Social Media, Short Message Service (SMS), Instant Messaging (IM) and other new platforms. The terms "unsolicited" and "unwanted" in the TREC definition are applicable for all of the spam messages.

Unfortunately spam distribution is currently a serious problem that spreads through a wide variety of channels and services. Some services commonly used to distribute junk content are web 2.0 applications (which gave rise to the concept of spam 2.0)

[1] Available at https://trec.nist.gov

[19], search engines [20, 40] (Webspam), email [86], short message service SMS [49, 104] and IM applications [92].

There are some manual spam-fighting techniques that work really fine and require a little effort from a system administrator. Examples of these manual technique are whitelisting and blacklisting [67]. It is very easy for an administrator to establish the directive that "everything that comes from a specific Internet Protocol (IP) address" is spam (blacklist) and everything that comes from other ones (whitelist) is ham (legitimate). However, in this chapter we will not explore this manual spam filtering systems, because they have maintenance costs and are not robust against spam techniques that use dynamic origins/sources of spam.

A variety of different automatic techniques to combat the problem of spam are also available. Some of them are used to filter spam in more than one environment, such as SMS message spam detection, Twitter messages, e-mail or IM text message. Other spam filtering systems are more specific, such as Webspam detectors. However, most of them are based on the same operating principles and use the basics of ML.

In order to distinguish legitimate messages from spam, the automatic classifiers need to "learn" the features of spam messages and the features of legitimate messages. This phase is called training and it is at this stage at which the classifier configures itself to distinguish the legitimate messages from the spam messages. To understand how this works, consider that some of the messages that a user receives contain the word "viagra". It's easy to understand that the existence of the word "viagra" in the texts, increases significantly the probability to be a spam message. If it is also combined with the words "offer" or "cheap", it is probably a spam message. However, the appearance of the word "at" does not provide relevant information, since it appears indistinctly in any type of message. The selection of the word ("viagra") as a relevant word and the exclusion of the word ("at") is a very important part of the learning process and is known as "feature engineering". The following section describes this process in detail.

2.2 Feature Selection

Based on the language properties in which the messages to be classified are written, it is necessary to discriminate between the words that can be used to improve the classification and between the words that will not provide any contribution to the messages classification. The identification of the words that must be considered message features is a work that has to be done before the classifier training process. A review of the literature has shown that there are statistical indicators to calculate the

measure and the quality of the words to distinguish text messages between spam and ham. The techniques that make possible the processing of text to extract information from messages are known as NLP and have been widely used in the field of antispam [45]. The classification ability is strongly related to the characteristics or words used to identify each class. A set of text preparation and text processing steps need to be done before the feature extraction takes place. These steps are presented next.

1. **Pre-Procesing**.

   Pre-processing comprises the following tasks:

   - **Removing punctuation marks**

     In the real life the texts include punctuation marks, which are useful to make sentences easier to understand for the reader, but they have low or no value for machine processing purposes. We can find punctuation marks like colon, semicolon, special symbols, emojis, etc. but they are not helpful in the computer automatic classification. When working in NLP, it has been demonstrated [98] that the cleaning and elimination of these characters is a mandatory step.

   - **Removing Stop Words**

     Texts message often contain words that are not used in the classification process because they can exist equally in spam and legitimate messages. To reduce the classifier complexity and processing time, it is convenient to remove these kind of words [47]. The classifier can also do this itself, usually using a dictionary/list or detecting the use of words that do not add value to the classification process, relegating them as not relevant to the classification process, but it may take quite some time. Stop Words are usually removed from texts at this step. The words like "in", "the" and "a" and others of this kind, can be removed before the tokenization process, resulting in a reduction in the number of tokens. Taking as an example the sentence "This is a very long text", we can see that after removing the stop words the sentence gets reduced while keeping its meaning "long text".

   - **Stemming**

     In the stemming process, the different forms of the word are converted into a single recognized form, avoiding concept duplicities and the problem of handling a concept as two different words. As an example a stemmer would leave the three words "cleans", "cleaning" and "cleaned" as the word "clean".

Once the first phase of pre-processing has been completed and the text is free of punctuation marks and Stop Words, the text is converted into input elements for the tokenisation phase.

2. **Tokenization**
   Tokenization splits a piece of text into individual words based on a certain delimiter. In this process [50] a piece of text is divided into individual words based on a delimiter (such as a blank space) for further text processing. These elements are represented in a feature vector, like the one shown in Table 2.1. When tokens are created, they can be formed by a single word, by two words, by three words or even more words. Continuing with the previous example we could have two unigrams ["long", "text"] or one bigram ["long text"] or even the combination of unigrams and bigrams such as ["long text", "message"].

3. **Representation**
   The processed texts must be represented in a specific format that can be read by the classification algorithm. As a general rule, this information can be stored in any way if there is a data transformer to make it as required by the classifier. However, in order to visualize how the algorithm works, the data must be represented in a way that has sense to humans. Next we will see the two representation methods that are most commonly used in text classification.

   - **Simple count**
     Is the simplest way to represent text. In this case the feature vector (one row per message) represents the tokens contained in a message and how many times they appear in each message. The collection of feature vectors forms a matrix, in which rows they appear the messages and columns the tokens and how many times they appear it in each message. The intersection between rows and columns shows the occurrences of the token in each message. The representation in Table 2.1 shows the occurrences of each word in the text. If a token appears twice in a message, this is indicated by the number 2 in the corresponding row. If the token is not contained in the message, it is shown with a 0 in the corresponding row.

     Message 1: The text has few words and is very easy to understand.
     Message 2: The text is written in English.
     Message 3: It is a fragment of a book.

the	text	has	few	words	and	is	easy	to	understand	writen	in	english	it	a	fragment	of	book
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	1	1	1

Table 2.1: Simple count feature vector representation.

- **Term Frequency - Inverse Document Frequency (TF-IDF)**

  TF-IDF (Term Frequency - Inverse Document Frequency) indicates the frequency of occurrence of a term/word in all messages belonging to the corpora /dataset of the messages. It is a numerical indicator that expresses how relevant is a word to identify a message type (ham or spam) in a collection of messages. The value increases when a term appears more often in a message because it permits to match the word with a message, and in this way, with the class ham or spam. This indicator helps to identify the words that appear more often in spam or legitimate messages. It may occur that a word appears many times in a message, but as a very common word it may also appear in many other messages in which case, its value to identify the class (ham or spam) of a message may be reduced. The TF-IDF set allows to handle the problem of the existence of words that are very common to all messages. The TF method provides better metrics related with the quality of a term than the number of occurrences of the term, making TF-IDF one of the most widely used metrics to show the quality of terms. Term Frequency is mathematically represented in equation 2.1 and Inverse Document Frequency is represented in equation 2.2.

$$TF(t, D) = \frac{f_{t,D}}{\sum_{t' \in D} f_{t'D}} \tag{2.1}$$

  In this equation $f_{t,D}$ is the Frequency of the term t in the document D and $\sum_{t' \in D} f_{t'D}$ is the number of terms in document D.

$$IDF(t, D) = \log \frac{N}{|d \in D : t \in d|} \tag{2.2}$$

  In the case of Inverse Document Frequency, for the term t in document D, N is the total of documents and $|d \in D : t \in d|$ is the number of documents with term t.

4. **Feature selection**

   Feature selection is the most important step before classification. Once the stop words have been removed, tokenization is completed and the terms are represented

in a feature vector, it is necessary to select the terms to be used by the classifier. The size of the feature set or terms will impact on the speed at which the classifier learns. As more information needs to be processed, more time will be spent in the learning or training phase. On the other hand, if there are few terms, the training phase will be very short, but the classification may not be good. In this section we describe some metrics that can help to select the best terms and discard others that do not contribute to the classification.

- **Information Gain (IG)**

  Before describing the Information Gain calculation, it is necessary to introduce the concept of entropy, which is mathematically represented by the equation 2.3.

$$entropy(S) = \sum_{i=1}^{n} -p_i \log_2 p_i \tag{2.3}$$

  Where S is a collection of sets, p_i is the probability of a certain event and i is an event, where $i = [1, n]$ and event is a word/term appearing in a message . As an example, we can see in the equation 2.4 what is the calculation of the entropy for obtaining a specific number on a six sided dice.

$$(S) = -6\left(\frac{1}{6}\log_2\left(\frac{1}{6}\right)\right) \approx 2,58 \tag{2.4}$$

  Entropy is a metric of uncertainty or disorder, and is used to help to decide which feature should be selected in the next step. In general, an attribute that helps to discriminate more objects has a tendency to reduce entropy, so it should be used in the next division of the dataset.

  Once the entropy concept has been described, the next step is to describe the Information Gain, which is mathematically represented by the equation 2.5.

$$InfoGain(S, F) = Entropy(S) - \sum_{v \in V(F)} \frac{|S_v|}{|S|} entropy(S_v) \tag{2.5}$$

  Where S is an objects set, F are the objects features and $V(F)$ are the values that the objects can take. This expression shows that a higher value of the Information Gain of a feature, makes higher its discriminative power. Thus, in order to identify the features that will be better to divide the messages into the two classes (ham and spam), the elements with the highest Information Gain are selected.

- **Gini index**

  The Gini index helps to select features based on the degree of purity of each feature in relation to the class. Purity measures the level of discrimination of one feature to differentiate between different classes. This feature selection method indicates the degree of purity of the chosen feature. For a selected feature, the Gini index is calculated as shown in equation 2.6.

$$GI(t_i) = \sum_{j=1}^{m} p(t_i|C_j)^2 p(C_j|t_i)^2 \tag{2.6}$$

  Where m is the number of classes, $p(t_i|C_j)$ is the t_i term probability to be in the given class and $p(C_j|t_i)$ is the probability that the class C_j contains the term t_i.

- **Chi-Square**

  The Chi-Square indicator is used to determine the existence or not of independence between two variables. When two variables are independent, it means that they are not related, so one does not depend of the other and vice versa. In this case, we can measure the lack of independence between a word (w) and a class C. Thus, with the study of independence, a method also verifies whether the frequencies observed in each category are compatible with the independence between both variables. To assess the independence between the variables, the values that would indicate absolute independence known as "expected frequencies" are calculated and compared with the frequencies in the sample. The calculation of the lack of independence between feature or characteristic (c) and class i can be seen in equation 2.7.

$$X_i^2 = \frac{nF(c)^2(p_i(c) - Pi)^2}{F(c)(1 - F(c))Pi(1 - Pi))} \tag{2.7}$$

  In this equation n is the total number of messages in the dataset, $p_i(c)$ is the probability of class i for messages that contains the feature c, P_i is the global fraction of messages consisting in class i and $F(c)$ is the global fraction of messages that contains the feature c. This measure returns the normalized value of X_i^2 and permits to identify the relevant features for different classes.

  With the features filtered, cleaned, represented and with the values they provide to the classification process calculated, it is time for feature selection. The objective is to use features that improve the quality of the classification results of a machine learning process, comparing with the raw data supply. The selection of those features, which do not always improve the classification process due to

their high value, is known as feature engineering.

2.3 How to distinguish legitimate messages from spam messages

From the first email sent by Ray Tomlinson in 1971 [94, 95] to the arrival of social networks or Instant Messaging, email has been the most common communication tool in the Internet and has consequently become the most [2] affected by the problem of spam, with spam shares close to 45%. A variety of systems have been developed to try to filter unwanted messages, some of them based on manual techniques as can be seen in Chapter 2 of the book of Gordon V. Cormack. et.al. [24] and others based entirely on ML techniques [91].

Today's spam filters can be divided into two categories[67], (i) non-ML based systems [24] and (ii) ML based systems [91]. In this research work we are going to address only ML based techniques.

2.3.1 Machine Learning Approaches

ML is one of the better performing approaches to spam filtering. ML techniques have the ability to learn what identify the spam emails by parsing lots of spam messages from a large collection of previously collected emails. Classifiers developed with these techniques have the ability to adapt to variable conditions, as they generate their own rules based on what they have learned. ML techniques can be divided into two classes [63], one based on "supervised learning" [70] and other based on "unsupervised learning" [46].

In supervised learning models, algorithms work with previously classified and labelled datasets, looking for a function to classify the input data into the correct class. The algorithm must infer what is the classification function, so an input data called training set is used to create it, which will be able to predict, with greater or lesser accuracy, the appropriate output class for a new input data. There exists several types of ML based classifiers that have been used in spam filtering. Some of the most popular classifiers are:

- **Decision trees**

  A decision tree is a technique used to represent the relationship between the elements of a dataset based on a series of conditions. For the classification of spam

[2] Available at https://securelist.com/spam-and-phishing-in-2021/105713/

messages, Boosting algorithms are usually used, which work by sequentially adjusting simple models that predict only slightly better than expected by random chance and each new model uses the information from the previous model, improving iteration by iteration. There are studies that claim that Boosting methods outperform [99] Naïve Bayes classifiers on specific email corpora. However, in the case of spam filtering, the most widely used algorithm has been AdaBoost [42].

- **Naïve Bayes classifiers**
  One of the best known algorithms for text classification is also called a linear classifier. In the case of spam classifiers, this type of probabilistic classifiers based on Bayes' theorem has been one of the first proposals, being widely analyzed in different works [32] [9].

- **Support Vector Machines (SVM)**
  SVM are a set of supervised learning algorithms developed by Vladimir Vapnik [105] and his team. These algorithms can help in classification as well as regression tasks. An SVM is a model that splits the points to be classified into two spaces as wide as possible by means of a separation hyperplane defined as the vector between the two closest points between the two classes, which is called the support vector. SVMs are very useful for text categorization problems, and are used for spam email filtering [64].

In the case of unsupervised learning algorithms, these are used when there is no labelled data for training. Therefore only the structure of the data can be described in order to find some type of organization to simplify the analysis. Clustering tasks try to makes groups based on similarities, but there is no guarantee that these have any useful meaning or utility. Some of the most commonly available algorithms for unsupervised learning are the following:

- **Clustering algorithms**
  Clustering [114] is the grouping of data into groups of similar items. Representing the data in a series of clusters, involves the loss of detail, but achieves simplification of detail. From a practical point of view clustering represents a very important role in data mining applications, such as information retrieval and text mining and is a technique to be used in this thesis. In case of using a suitable representation, most clustering algorithms can split between spam or legitimate email datasets, as demonstrated by Whisshell et.al. [111].

- **Principal Component Analysis (PCA):**
  PCA [35] is a statistical procedure that uses an orthogonal transformation to convert

Id	Ok	Pressure	RPM	Temperature
1	0	0	0	125
2	1	0,50	0	126
3	1	0,72	0	127
4	1	3,25	0	133
5	1	3,75	0	137
6	1	4,50	100	147
7	1	5,00	300	137
8	1	5,00	390	147
9	1	3,25	380	141

Table 2.2: AOI manufacturing process variable generalization step 0.

a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

2.4 Conceptual clustering

The Attribute Oriented Induction (AOI) algorithm is a hierarchical clustering algorithm based on the generalization concept. It was first used by Jiawei Han, Yandong Cai, and Nick Cercone in 1992 as a method for knowledge discovery in databases [54]. The AOI algorithm follows an iterative process, in which each attribute or value of a system variable has its own hierarchical tree. When a variable moves from one level of generalisation to another, that generalisation is applied to all the data in the set. This step is called concept tree ascension [23].

The use of more general elements makes it possible to find clusters that otherwise could not be generated. To understand the concept of generalization graphically, an example with variable values of a manufacturing process is proposed. These variables are shown in Table 2.2, whose first column identifies the number of the part produced, the second variable is the indicator of defective (with 0) part or good (with 1) part, the third measure is the pressure applied to the part, the fourth shows the r.p.m. of the pressure pump and finally, the value of the temperature of the fluid. In this example it is clear that in table 2.2 all the parts have been produced with different values.

In the first iteration of the algorithm, the values of the temperature variable are generalised. Values below 130 will be generalised to the value "low". Values between 131 and 140 will be generalised to the "medium" value and values above 140 will be generalised to the "high" value. The result of these changes can be seen in table 2.3.

In the second iteration of the algorithm, the values of the pressure variable will be generalised. Values between 0 and 1,00 are generalised to the value "low". Values

Id	Ok	Pressure	RPM	Temperature
1	0	0	0	low
2	1	0,50	0	low
3	1	0,72	0	low
4	1	3,25	0	medium
5	1	3,75	0	medium
6	1	4,50	100	high
7	1	5,00	300	medium
8	1	5,00	390	high
9	1	3,25	380	high

Table 2.3: AOI manufacturing process variable generalization step 1.

Id	Ok	Pressure	RPM	Temperature
1	0	low	0	low
2	1	low	0	low
3	1	low	0	low
4	1	high	0	medium
5	1	high	0	medium
6	1	high	100	high
7	1	high	300	medium
8	1	high	390	high
9	1	high	380	high

Table 2.4: AOI manufacturing process variable generalization step 2.

between 1,01 and 2,99 are generalised to the value "medium". Finally, values above 3,00 are generalised to the value "high". The result of these changes can be seen in Table 2.4.

At this stage of generalization and after several iterations of the AOI algorithm it is possible to generate clusters. In this example, the data for parts 2 and 3 are the same and the data for parts 4 and 5 are the same. These equal data will be used to create a two clusters and we will continue with the application of the algorithm with the data of the Table 2.5.

For the next iteration of the AOI algorithm, additional data needs to be generalised, so we work with the column of revolutions per minute of the pump. In this case values below 100 rpm are generalised to "low" values, values between 101 and 299 RPM are generalised to "medium" values, values between 300 RPM and 349 are generalised to "high" values and values above 350 are generalised to "very high". In this way there are two other parts with the same manufacturing conditions, parts 8 and 9, which will

Id	Ok	Pressure	RPM	Temperature
1	0	low	0	low
6	1	high	100	high
7	1	high	300	medium
8	1	high	390	high
9	1	high	380	high

Table 2.5: Values for parts 2,3,4, and 5 are stored in a cluster.

Id	Ok	Pressure	RPM	Temperature
1	0	low	low	low
6	1	high	medium	high
7	1	high	high	medium
8	1	high	very high	high
9	1	high	very high	high

Table 2.6: New cluster is generated for parts 8 and 9.

Id	Ok	Pressure	RPM	Temperature
1	0	low	low	low
6	1	high	medium	high
7	1	high	high	medium

Table 2.7: Final state of the parts manufacturing table.

be grouped into a cluster. The results of this step can be seen in Table 2.6.

The iteration of the algorithm can be executed as many times as required and the generalization of the data can be as large as necessary. It would even be possible to group all available data into a single cluster under very general conditions. However, applying generalisations that group two or more tuples is enough for this example. The stopping condition of the algorithm is determined by the number of clusters to be generated. For this example, having reached the three clusters stage, it has been decided to stop the execution of the algorithm. This leaves out of the clusters some data, in particular the ones shown in Table 2.7.

The application of the AOI algorithm to the manufacturing parts in the example, makes it possible to identify the three clusters of data shown in the Table 2.8. These clusters make it possible to join manufacturing conditions that were initially completely different from each other.

cluster	Ok	Pressure	RPM	Temperature
1	1	low	0	low
2	1	high	0	medium
3	1	high	very high	hign

Table 2.8: Clusters generated final state of the parts manufacturing table.

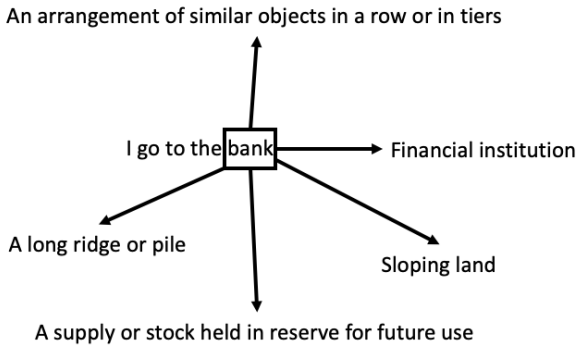


Figure 2.1: Disambiguation problem for the word "bank".

2.5 Disambiguation

The process of semantic disambiguation of words, also known as Word Sense Disambiguation (WSD), consists in the identification of the particular meaning of a polysemous word [79] within a sentence or within a context. WSD is necessary to improve the tasks such as machine translation [18], syntactic analysis [4] or Information Retrieval (IR) [118], and in our case it also helps to improve the classification of messages. The assignment of the correct meaning to each word within a context is complex and has been studied in conjunction with NLP, so there are many works on this topic, such as the article by Ide et. al. [58], chapters in generic NLP books such as Chapter 7 of the book Foundations of statistical Natural Language Processing [73], books focused in the state of the art such as Navigli et. al [79] and specific books and works [3, 103, 13].

In the sentence shown in Figure 2.1 it can be seen that the word "bank" has multiple meanings. It is necessary to disambiguate this word within a context because depending on the words that follows "bank", its meaning can be different. If it is followed by "blood" it will have one meaning, if it is followed by "to sit down" it will have another meaning and if it is followed by "to take out money", it will have another meaning. The ability to distinguish between the different meanings is key to improving filtering and classifying messages as spam or ham.

Currently in NLP tasks that require disambiguation, the trend is to use databases

that connect synsets (groups of meaning) with others through semantic relations represented in the form of graphs, such as WordNet[3] or BabelNet[4], as can be seen in the works [72], [82], [81]. The techniques used by these databases explore the existing relationships between the senses of a word in a specific context.

2.6 Optimization with Genetic Algorithms

The principles of the Genetic Algorithm (GA) were established by Holland [55] in 1975. GAs consist of adaptive methods that are used to solve search or optimization problems using the genetic selection process of living organisms. GAs start working with a population of candidate solutions (called individuals), each one representing a possible solution to the problem. GAs are able to scan the solution space, evaluate how valid is the proposed solution, measuring how close or not they are to their goal. With GA it is mandatory to have a genetic representation of the solution domain. For each individual, a value is assigned that indicates how good it is at solving the problem, known as a fitness function, that evaluates the solution domain and identifies the best individuals. These top individuals will be the ones that are crossed with other individuals, generating offspring that will inherit some of the characteristics of their parents. In this way, a new population of possible solutions is produced, which replaces the previous one as long as it has better characteristics to solve the problem.

In GAs, the possible solutions to the problem are represented as a set of parameters, known as the genes, which are grouped together in a string to form the chromosome. These chromosomes will be selected in groups of two to generate two offspring, which will be other possible solutions. These offspring will have a combined part of their parents' chromosome. A mutation operation can act on the offspring, which can randomly generate "super-individuals" that can be closer to the optimal solution. Figure 2.2 shows the process of combining the genes of two parents and mutating of one of the children.

Genetic algorithms are widely and successfully used in complex optimization processes, including spam detection works [62, 93, 12]. It is also common that the optimization process is performed to improve more than one simultaneous objective, known as multi-objective optimization [90, 56]. It is in these cases where genetic algorithms can explore different combinations of the solution space in a targeted way using multiple fitness functions.

[3] Available at https://wordnet.princeton.edu/
[4] Available at https://babelnet.org

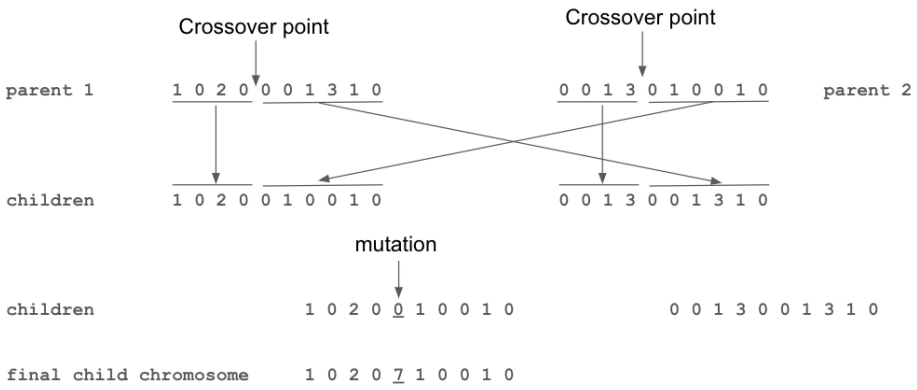


Figure 2.2: Genetic algorithm iteration.

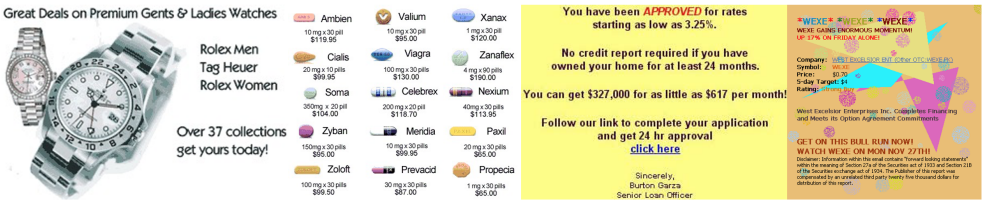


Figure 2.3: Examples of images attached to spam messages. These images are part of publicly available "Image Spam Dataset".

2.7 Text obfuscation

In the fight against spam, spammers are constantly looking for new ways to avoid anti-spam filters. One way to prevent content filters from analyzing the text of messages consisted in embedding text inside images [15]. This trick became very popular in 2006 and 2007 [10, 15] and is based on the fact that anti-spam filters cannot analyze the content of images but they are easily identifiable and understandable by humans. Figure 2.3 shows several images containing embedded text, which cannot be processed by content-based classifiers [6], but are easily identifiable as spam by anyone.

In order to avoid this type of spam, some researchers have used Optical character recognition (OCR) techniques [22] but they are quite computationally demanding and vulnerable to image artefacts [44] even making a post-corrections [60]. Moreover, Fumera et al. published in their work [43] "Spam filtering based on the analysis of text information embedded into images" an empirical experiment demonstrating that text can easily be hidden from the OCR system making spam filters unable to identify this type of spam messages. In addition, spammers also add noise [36], as in the image on the right hand side of figure 2.3. Recent techniques such as CAPTCHA [16], perform image distortions in order to make them impossible to be interpreted by an

v̲iagra	vi̲agra	via̲gra	viag̲ra	viagr̲a	viagra
\/iagra	v1agra	vi4gra	via6ra	viag12a	viagr/\
|/iagra	v¡agra	vi/\gra	via(_-a	viag/2a	viagr4

Table 2.9: Word "viagra" writen in Leetspeak.

automatic text recognition system. However, we may be close to Artificial Neural Network (ANN)-based systems being able to recognize the texts [110, 34] embedded in these captchas.

Another important area in terms of image spam is the Leetspeak slang, also known as leet, leet text or 1337. This type of syntax has been used since 1980 [34, 41] and consists of replacing some characters with symbols that are visually similar to those characters, allowing to read the text without any lexical, syntactical or semantic loss. This type of encoding has two effects (i) it prevents the classifier from identifying, tokenising and processing the word and (ii) it produces a Bayesian poisoning attack [116] by inserting random, and apparently harmless, words into the texts of spam messages, causing the spam message to be incorrectly classified. Table 2.9 shows 12 Leetspeak representations of the word "viagra". Each column of Table 2.9 shows a possible substitution of a single character in the word.

As can be seen in Table 2.9, Leetspeak exploits the similarity of certain characters with a punctuation mark (or combinations of them) to substitute the characters and preventing the spam filter to correctly process the word and classify the message. This character substitution causes misrecognition and misrepresentation of the word that contains the character during the classification phase, which allows spammers to bypass filters. In Leetspeak any character can be replaced in some cases by a single symbol, as in the case of the "i" which can be replaced by a "¡", or in other cases by a combination of several symbols, as in the case of the "A" character, which requires two "/\" and even three "/-\" symbols. Since Leetspeak does not use a limited and predefined number of characters, it it is not possible to enumerate all possible words transformations into a dictionary. Table 2.10 shows an example of possible substitutions used when writing texts in Leetspeak.

a	4, /\, /-\, /-\	h	#, /-/, [-],]-[,)-(, (-), :-:, /∼/, /-/,]∼[, }{	o	0, (), [], ø	v	\/, /\/
b	8,/3, ß,]3,]8, /8, !3	i	1, !, /,], :	p	/º, /?	w	\/\/, \^/, _/_/, _:_/, /\/, ‘//
c	(, <, [, ©, ç, {	j	¿, , _/, _), 7	q	"(_,)", "(),"	x	)(, %, ><
d	[]), /), /], [>, ,])	k	/{, /<, /(, }<	r	/2, /2, 12	y	’/
e	3, [- ,£, €	l	/_, []_, [_, 1_	s	5, $, §	z	7_, 2, >_
f	=, /=, /#	m	/\/, /\/\, (\/), [\/], /\/\/\, /^^\,	t	+, †		
g	6, (_+, (_-	n	/\/, /\/, [\], (\), /\/\/	u	/_/, _/, (_), /_/,]_[		

Table 2.10: Examples of posible Leetspeak substitutions.

Chapter 3

State of the art

The aim of this chapter is to describe the state of the art in the areas to which this research has contributed. In order to establish the current state of the art, the literature related to each of the topics covered in this thesis has been reviewed, such as topic and concept identification, a subject that has been developed in several previous works, and the deobfuscation of characters, an area in which very few research works have been found.

3.1 Topic and concept identification

ML has experienced incredible expansion as it has been used to solve a multitude of problems. Thanks to its ability to use past experiences and related information as input data, it is being widely used in problems where there is a large volume of data. For the particular problem of spam filtering, a multitude of binary classifiers can be used in popular tools such as Weka [53]. The good values in term of evaluation metrics (Precision and Recall) obtained with these classifiers for spam filtering in any kind of environment has been demonstrated in several works and is therefore beyond doubt [109, 66, 5]. However, these classifiers are highly dependent on the input data, which means that if the input data is not relevant or not clean, the result in terms of Precision and Recall can be get worse. A key aspect to apply ML techniques on a dataset is the previous preparation [117] because the performance achieved by the classifiers depends directly on this aspect.

Token-based information mining has provided the basis for what is known as text mining [97]. This has emerged as the way to exploit token information to solve problems such as information retrieval [65] and text classification [59] for instance. From the first known ML proposal for spam filtering by Paul Graham[1], this technique has evolved to others [77] [87] that have been introduced afterwards and that take

[1] Available at http://paulgraham.com/spam.html

advantage of token-based information. This has made possible the emergence of topic classification models [57]. These models were used to identify the terms that are usually related to the texts, allowing the identification of the subject related to the message. This association of terms and topics was done without using semantic information about the connections between the terms. This type of models could find the statistical relationships between topics and terms, allowing to classify the different documents according to these topics.

Thanks to advances in online dictionaries and, above all, the development of ontological dictionaries, a new method of content representation of a text has been developed in recent years: The concept [76] representation. Concepts have been modeled and represented by synsets (synonym sets), which are groupings of synonyms that identify meanings. This type of information mapping allows the identification of different textual representations [7] as referring to the same concept. In particular, the following text fragments "flat for sale", "I sell my flat", "flat offer for sale" represent the same information but are formed by different tokens.

A common problem for text representations which have to be processed by an ML algorithm is dimensionality. The dimensionality problem occurs in token-based representations, and it also occurs in concept-based representations [76]. When dimensionality is high, which is common in text classification problems, the classifiers complexity and computing resources to operate with the data increase, eventually to computationally unfeasible levels. Texts may contain redundant, irrelevant, dependent or incongruent words or tokens that must be identified and ignored in the text conceptual representation. In order to solve this problem, as we can see in this review of the literature that has been performed, different types of feature selection have been proposed, with the objective of reducing the number of features. Previous works [21, 31, 71] suggest that the classical schemes used for feature reduction up to now can be classified into three groups such as (i) fliltering, (ii) wrapping and (iii) embedding,

Feature selection methods (i) can reduce the dimensionality problem by evaluating the relevance of features present in the input data. These relevance values are quick to compute and can be easily used to select the most relevant features. A threshold value can be set and only those features that exceed it will be selected. A maximum number of features can also be set and selected only the maximum number of features exceeding the threshold value. The major advantage of filters is their simplicity but they have the problem that they are not able to select independent variables and cannot avoid duplication of information. Wrapping methods (ii) combine a strategy of clustering input features. In this case, for the selection of each group of features, an

ML algorithm is used to solve the problem with a subset of the input data, selecting the ones that come closest to solve the problem. The main issue is that the selected features may be the ones that most contribute to the solution of the problem, eventually leading to overfitting phenomena. The third and final method, embedded feature selection schemes (iii), use feature reduction methods that are based on specific ML algorithms for this task.

The advances in the use of semantic information provided by ontological dictionaries, permitted the development of new methods for the feature selection. This feature selection does not always have to involve a reduction, as in some cases it may be necessary to increase the number of features. In the work developed by Almeida et. al. with SMS [7], new features are artificially added to the messages. Due to the small size of these messages, after identifying the words that form the messages, synonyms of these words are added to improve their classification.

In other cases, as in the study by Bahgat et. al [11], the authors reduce the number of terms by analyzing the synonymy relations, making synonyms groups in a single term. Although the feature reduction was low, this work was the first to introduce the "concept" feature to address the problem of email classification. Table 3.1 shows the reduction achieved when using this proposal for the text: “The lowest prices in the world on: Alprazolam, Xanax and tranquillizer. And many other great offers, only using your computer network.” when using this proposal.

Another work related with feature reduction is a study done by Abiramasundari et. al. [2] which makes use of semantic information to reduce the number of words of the messages, resulting in a reduction in the number of features. The proposed algorithm checks the meaning of each word in a semantic dictionary and in case the word has no meaning, it is immediately removed.

Using others information sources like Wikipedia [85] and also the property of synonyms we find the study of Venkatraman et. al. [107]. In this work the authors train the model that classifies spam messages and legitimate messages and make it work in conjunction with another subsystem that reinforces the classification. In this study, each of the terms is first checked in the Naïve Bayes classifier, performing a classification of messages that are spam and legitimate. In the second subsystem, terms that cannot be used to classify the message because they have not been used in the training phase and are unknown to the Naïve Bayes classifier are reviewed. Semantic Similarity (SS) is calculated based on the words that are used to create links between the different Wikipedia pages. In case the Bayesian classifier identifies the word "sales" as spam, all words that are used as links to other sites on the Wikipedia web page for the term "sales" are analyzed and marked as possible spam. Thus, if in

Original term	lowes	prices	world	alprazolam	Xanax	tranquilizer	great	offers	computer	network
Occurrences	1	1	1	1	1	1	1	1	1	1
Union term				alprazolam						
Final term / occurrence ratio	lowest (1), prices (1), world (1), alprazolam (2), tranquilizers (1), great (1), offer (1), computer (1), network (1)									

Table 3.1: Bahgat et al. feature reduction proposal, based on the same synonyms.

Original term	lowes	prices	world	alprazolam	Xanax	tranquilizer	great	offers	computer	network
Four top level synset match		value	whole	agent	agent	agent	person	event	whole	system
Four top level synsetidentifyed				value, whole, agent, person, event, system						

Table 3.2: Feature reduction matching top level synsets of Wordnet by Mendez et al. .

the text that forms the message appears the word "advertisement" and it is an unknown word for the Bayes classifier, this word is checked in the second subsystem. In case of matching any related word in Wikipedia with a word that has been identified as spam, the subsystem marks the word as possible spam. In this case, no semantic dictionaries are used and it is the authors of the study who implemented the subsystem and also computed the relationship between the Wikipedia terms.

In the study of Mendez et. al. [76], more focused on semantic feature selection, used the hypernym and hyponym relations to achieve a reduction in the number of features from any number of features to a maximum of 181. The use of the hypernym relations allows for generalizations by ascending the tree of semantic relations and moving up to more general concepts. In the study by Mendez et. al. the generalizations were made up to one of the synsets in the first 4 levels of the Wordnet ontological dictionary. In this way, words such as "xanax", "viagra", "cialis" or "tadalafil" can be grouped into a higher concept such as "drug" or "chemical". The use of semantic information allows a generalization process with no information loss, as words are not eliminated, they are grouped into a higher level semantic concept. Table 3.2 shows the reduction achieved when using this proposal for the sentence *"The lowest prices in the world on: Alprazolam, Xanax and tranquilizer. And many other great offers, only using your computer network."*.

As shown in the Table 3.2, although the proposal achieves a good reduction level, some terms are deleted or transformed into quite different concepts (great - person). Additionally, this work does not include the definition of word disambiguation schemes [79] which is mandatory for this kind of processes.

In conclusion, it can be observed that the identification of the "concept" or the thematic identification of the message consists of grouping all the content of the message into a few key words or terms. For this task it is essential to analyze the semantic relationships between words, to group synonyms in a direct way and also those words that are not synonyms but descend from the same higher synset. This identification of the "concept" or subject matter allows for matching the message with

its intention. So we may have commercial messages that try to persuade us to buy a product or messages with offers of money and great financial opportunities, whose intention is to scam us.

In the literature review, other works have been found, such as the work of Saidani et. al. [89] in which they propose a method based on a two-level semantic analysis. At the first level, they categorise emails into specific domains, such as Adults, Computers, Health, Education, Finance, and Others. After performing feature selection using the IG of the tokens, several classification algorithms are tested to categorize the email documents by domain using different classification algorithms, such as Naive Bayes, K-nearest neighbour (KNN), Decision tree, AdaBoost, SVM and Random forest. Once this first categorization has been performed, specific rules are created inside each category to identify spam messages. The objective is to create a domain-specific semantic filtration rules for an efficient classification of spam messages. In this proposal automatically created rules do not use semantic information, this is only used for manually entered rules. The automatically generated rule *"r: buy* $\wedge$ *drug* $\rightarrow$ *Spam"* can be replicated by using synonyms existing in Wordnet, and a manually generated rule such as *"r': buying* $\wedge$ *medicine* $\rightarrow$ *Spam"* or *"r": buy* $\wedge$ *Viagra* $\rightarrow$ *Spam"* can be added. The problem with this approach is that the rules are generated manually and require experts to check the words present in each category and then find their synonyms in Wordnet.

We can conclude that the latest studies use synsets to represent the meaning of words. However, the possibilities provided by synset relations have not been sufficiently explored. From a simple search that groups synonyms into a single synset [11], to a reduction from any number of synsets to 181 [76] there are many intermediate possibilities to be explored. This study will focus on these intermediate states.

3.2 Obfuscation

There are few previous works that have addressed this problem without the use of a dictionary and the direct conversion of obfuscated characters to their equivalent in the form of a text character [15]. In a work by Tundis et al. [101] they show how they have designed a CNN to directly classify texts in which Leetspeak has been used to obfuscate characters. Considering that the response obtained by using direct classification strategies is not always explainable, it indicates that such methods have their limitations. Instead of classifying texts directly, it is considered that it would be more suitable to identify the obfuscated characters and then proceed to classify the

text without any encoding in Leetspeak, which is much closer to the human approach. This type of solution is included in Explainable Artificial Intelligence (XAI) [33], in a work [102] that the same authors proposed shortly after, with a new algorithm that complies with XAI. To do this, authors developed a rule-based algorithm that takes advantage of a low-precision CNN (rule-2) that was generated using the Chars74K [26] image dataset and a collection of images representing non-English characters. During the course of the work in this thesis, many CNN models have been trained for the identification of obfuscated characters with Leetspeak and when we trained the CNN with the Chars74K image dataset, we achieved classification scores in the range of 42-52%. The combination of rules-based strategies has allowed the authors to improve the quality of the results obtained.

Chapter 4

Token reduction using generalization

Content-based spam filters developed in recent years have been implemented using ML techniques, where text has been represented using tokens for subsequent classification. In this context, there are several works that propose the use of semantic representation in order to take advantage of the use of synonyms. This approach achieves an improvement in classification and in particular a reduction in the number of tokens in the feature vector, reducing the dimensionality (size) of the problem. These first proposals have limitations and apply simplifications that should be implemented more gradually. This chapter addresses the problem of synset based dimensionality reduction without information loss by using the semantic information.

4.1 Introducing the Semantic Dimensionality Reduction System

The use of synsets (synonym sets) and ontological dictionaries, like BabelNet, allows the navigation through the taxonomic connections of the words and synsets, allowing the clustering of words such as "viagra" and "cialis" under the more general "anti-impotence drug", "drug" or "chemical_substance" depending on the used hypernym relations level (1, 2 or 3). The ability to group the words of the messages under a more general concept, opens the door to identify the category of the message and its intentionality, that is the aim of this and the following research projects in this area. In our case and to identify the best generalization level for each word a Multi-Objective Evolutionary Algorithms (MOEA) has been used, in particular the Non-Dominated Sorting Genetic Algorithm (NSGA-II).

In order to perform the proposed feature clustering process it is necessary to repre-

sent the text messages (tweets, IM, SMS or email) in the form of synset. Following the recommendation of a previous work[7], the BabelNet ontological dictionary will be used in this research project. All of the tokens extracted from the text are previously disambiguated using Babelfy[1] to identify the BabelNet synset that best matches the meaning of the token. In the process of mapping words to synsets, a corpus has been transformed into a dataset (D) with synsets represented in columns and messages represented in rows. Each row contains the target class information (ham or spam), number of occurrences of each feature ($S = \{s_1, s_2, s_3, .., s_n\}$) in the message and a message identifier. Table 4.1 shows an example of BabelNet synset tokenized dataset (D) feature vector, for messages in dataset D.

The new proposal takes advantage of the semantic information of words using ontologies, but also follows the wrappers feature selection. For the generalization of concepts we follow the proposal of a previous work [76] that proposes the use of hypernym relations of synset. However, in this new proposal, not all the synsets are generalized. To identify the best combination of generalized synset and the number of generalization steps for each one, a multi-objective optimization problem has been formulated. To solve it, it has been decided to use MOEAs from which the NSGA-II [30] has been selected because it has been successfully used to solve many different types of multi-objective optimization problems.

To determine which synset should be generalized, it is necessary to model the problem using a chromosome, which in our case is represented as $C = \{c_1, c_2, c_3, ..., c_n\}$. Each of the elements of the chromosome (gene) represents how many levels are needed to generalize a feature $s_i \in S$ (a synset in an ontological dictionary). The integer values of this chromosome $c_i(i = 1..n)$ indicate, using an integer number in the interval $[0..\gamma]$, the number of generalization steps of the synset where γ represents the maximum generalization level.

Lets assume $T(D, C)$ as the transformation of a dataset D on which the chromosome C is applied. In order to produce a dimensionality reduction (grouping more than one synset into a hypernym), it is necessary to take into account that if a synset s_i is generalised m levels and converted into a new synset s'_i, all the features or synsets S that are its hyponyms s'_i ($\{s_j \in S | s_j \in hyponyms(s'_i)\}$) may be eliminated to be grouped and represented into the new synset s'_i.

Applying the chromosome $C = \{1, 0, 2, 1, 0\}$ to the example in Table 4.1, the synset 'bn:00007329n (bus)' is generalized using one level, becoming 'bn:00065101n (public transport)'. The synset 'bn:00008553n (Barcelona)' is maintainded in the same state as the level of generalization indicated by the chromosome is 0. The synset

[1] Available at https://www.babelfy.org

'bn:00071570n (viagra)' is generalized using two levels becoming first 'bn:00004605n (anti-impotence drug)' and then 'bn:00028872n (drug)'. The synset 'bn:00019048n (cialis)' is generalized using one level becoming 'bn:00004605n (anti-impotence drug)'. The last synset 'bn:00042905n (happy)' is also not generalized (chromosome indicates 0 levels of generalization). The transformation and reduction of the dataset can be observed in 4.2. It should be observed that the generalized synset 'bn:00019048n (cialis)' has become 'bn:00004605n (anti-impotence drug)' but it is a hyponym of the synset 'bn:00028872n (drug)' which is now the highest reference and which includes all its hyponyms. Due to that, the synset 'bn:00004605n (impotence medicine)' is grouped and added to the synset 'bn:00028872n (medicine)'. When s_1 and s_2 have the same direct hypernym, they are merged in the highest hipernym of s_1 and s_2.

$S = \{s_1, s_2, s_3, .., s_n\}$						
id	**bn:00007329n (bus)**	**bn:00008553n (Barcelona)**	**bn:00071570n (viagra)**	**bn:00019048n (cialis)**	**bn:00042905n (happy)**	**target**
1	0	0	0	0	1	ham
2	0	1	0	0	1	ham
3	1	1	0	0	0	ham
4	0	0	0	1	0	spam
5	0	0	1	0	0	spam

Table 4.1: BabelNet synset based tokenized dataset (D) feature vector.

$S = \{s_1, s_2, s_3, .., s_n\}$					
id	**bn:00065101n (public transport)**	**bn:00008553n (Barcelona)**	**bn:00028872n (drug)**	**bn:00042905n (happy)**	**target**
1	0	0	0	1	ham
2	0	1	0	1	ham
3	1	1	0	0	ham
4	0	0	1	0	spam
5	0	0	1	0	spam

Table 4.2: Transformation and reduction of dataset D applying the chromosome $C = \{1, 0, 2, 1, 0\}$.

When using a MOEA, in addition to representing the problem in a chromosome form, it is also necessary the definition of the optimization objective functions. Following the wrapper approach, we will use the values obtained from the classifier using a 10-fold cross validation scheme [69]. Analyzing metrics from previous work in spam filter optimization [12, 88, 115] the number of FP (legitimate texts classified as spam) and FN (spam messages classified as spam) will be established as objectives to optimize. The main objective will continue to be the dimensionality reduction

consisting in the grouping of synsets into a smaller number of more general concepts, so this function will also be included as another objective function. The objective functions of this problem are shown in equation 4.1.

$$\begin{aligned} f_1 &= 10xval\_eval.FP_r(c, T(D, C)) \\ f_2 &= 10xval\_eval.FN_r(c, T(D, C)) \\ f3 &= \frac{col\_num(T(D, C))}{col\_num(D)} \end{aligned} \tag{4.1}$$

In equation 4.1 in the f_1 and f_2 functions, $10xval\_eval.FP_r(c, T(D, C))$ and $10xval\_eval.FN_r(c, T(D, C))$ represent the FP and FN result ratios after the execution of the classifier c, that is selected for evaluation, over the D Dataset using a 10-fold cross validation. $col\_num(D)$ is the number of columns in the dataset (D).

The optimization experiment was performed with these objective functions, testing the different combinations of chromosomes generated by the NSGA-II genetic algorithm with the aim of grouping the largest number of features (dimensionality reduction and topic identification) maintaining the FP and FN values to a minimum. The following section shows the results obtained in the experiments carried out to evaluate this new proposal.

4.2 Experiments design and evaluation

In order to test the hypothesis of the proposal an experiment has been designed. The experiment operating scheme can be seen in figure 4.1. One of the keys to check if the proposal is valid or not consists in a double performance verification, using different data (not previously known by the classifier), which allows to detect and avoids the overfitting problem.

The evaluation of the new proposal has been performed using a text corpus as input data. As shown in figure 4.1, this corpus has been divided into two stratified parts containing 75% and 25% of the input texts. The larger subset is used to run the topics clustering and dimensionality reduction procedure, allowing MOEAs to solve the optimization problem of finding the best dimensionality reduction and classification performance tradeoffs for the dataset. The best chromosomes are obtained by computing the set of non-dominated solutions, which represent the optimal set of solutions, i.e. the Pareto front. After the best set of solutions have been identified, we need to select one of the solutions, which might represent the preferred tradeoff among the objectives. In our case, it was decided that the most balanced solution with respect to False Positive ratio (FPr) and False Negative ratio (FNr), is preferred.

The preferred solution is computed as the one presenting the lowest distance to the origin vector (0,0). From these minimum values, the four best ones are selected and a training/testing iteration is performed using the (75%,25%) data set in order to obtain results that can be compared with another set of synsets clustering solutions.

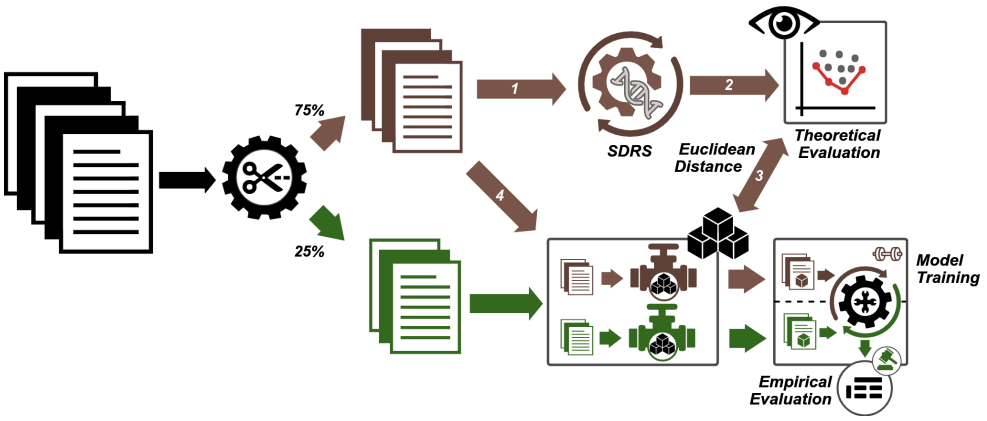


Figure 4.1: Experimental protocol.

4.3 Dataset selection

A large number of corpora are available on the Internet to evaluate different anti-spam filters and techniques. Table 4.4 lists a number of corpora to carry out experiments and to compare the results that have been extensively used in anti-spam filtering approaches. The table shows the percentage of spam contained in each corpus, the address from which it can be downloaded and a description of its content.

Due to the high number of evaluations and fine-tuning that have to be performed by the MOEAs, a small corpus has been selected, in order to carry out the experiments and to obtain significant results. The proposed term clustering and dimensionality reduction proposal involves that each chromosome selected for optimization needs a 10-fold cross validation for the 75% of the input dataset, as described in the section "Experiments design and evaluation". Table 4.3 shows the execution times of each experiment on different machines with different configurations.

Computer specifications	Execution time in days
4 x Intel Xeon E7-8890 (Q2/2016) v3 2.5 GHz (18 cores/36 threads) 1 TB RAM	10
2 x Intel Xeon E5-2640 (Q1/2012) v3 2.6 GHz (8 cores/12 threads) 128 GB RAM	13
2 x Intel Xeon X5675 (Q1/2011) 3.07 GHz with (6 cores/12 threads). 128 GB RAM	13

Table 4.3: Time required in days to complete the experiment.

Considering that the size of the dataset was a critical factor and also that the

selected dataset had to be sufficient in size to validate the results, the YouTube spam Collection dataset has been selected. This dataset is available in the ML repository of the University of California, Irvine (UCI). This dataset is considered adequate considering that MOEAs (due to their stochastic nature) requires thousands of evaluations of the objective functions and several tens of runs of the algorithms.

Dataset	spam percentage	URL	Content description
SMS spam Collection v.1	13%	https://archive.ics.uci.edu/ml/dataset/SMS+spam+Collection	5,574 SMS
British English SMS corpora	48%	https://mtaufiqnzz.files.wordpress.com/2010/06/british-english-sms-corpora.doc	875 SMS
Hspam14.s2	unknown	https://doi.org/10.1145/2766462.2767701	14M Twitter messages (tweets)
YouTube Comments Dataset	7%	http://mlg.ucd.ie/yt/	6M YouTube cooments
YouTube spam Collection Dataset	49%	https://archive.ics.uci.edu/ml/datasets/YouTube+spam+Collection	1,956 YouTube comments
spam Corpus	34%	https://github.com/hexgnu/spam_filter/tree/master/data	4,027 e-mails
TREC 2007 Public Corpus	66%	http://plg.uwaterloo.ca/~gvcormac/treccorpus07/	75,419 e-mails
spamAssassin	31%	http://spamassassin.apache.org/old/publiccorpus/	6,047 e-mails
Enron email	0%	http://www.cs.cmu.edu/~enron/	619,446 e-mails
Bruce Guenter spam collection	100%	http://untroubled.org/spam/	>3,000,000 e-mails
Ling spam	16%	http://csmining.org/index.php/ling-spam-datasets.html	2,893 e-mails
Webspam-UK 2007	unknown	http://chato.cl/webspam/datasets/index.php	105,896,555 websites (HTML)
Webspam-UK 2011	53%	https://sites.google.com/site/heiderawahsheh/home/web-spam-2011-datasets/uk-2011-web-spam-dataset	3,766 Web websites (HTML)
DC 2010 / EU 2010	unknown	https://dms.sztaki.hu/en/letoltes/ecmlpkdd-2010-discovery-challenge-data-set	23M websites (HTML)
Webb spam 2011	unknown	http://www.cc.gatech.edu/projects/doi/WebbspamCorpus.html	330,000 websites (HTML)
Clueweb 09	unknown	http://www.lemurproject.org/clueweb09.php/	1,040M websites (HTML)
Clueweb 12	unknown	http://www.lemurproject.org/clueweb12.php/	870M websites (HTML)
Common Crawl Data	100%	http://commoncrawl.org/	9 Billion in 2014 and increasing websites (HTML)

Table 4.4: Public corpora with ham/spam texts.

4.4 Text pre-processing

Since the messages contained in the YouTube dataset are in raw text, it has been necessary to pre-process them before starting to work with the dataset. For this work we have used the applications developed in the Big Data Pipelining for Java (BDP4J)[2] and Natural Language pre-processing Architecture (NLPA)[3] projects. Once the pre-processing phase was completed, we proceeded to disambiguate the text with the help of Babelfy, using in conjunction with the BabelNet ontological dictionary to identify the synset of each token.

The pre-processing of the texts starts with the extraction of the comments texts using the YouTube API. The next step is the removal of HTML tags, CSS-related tags, URLs and JavaScript code. Only entities are maintained, which are translated to their plain text equivalent. Emoticons and emojis[4] previously known and identified are removed from the messages, but stored for future processing. In order to correctly remove stop words, onomatopoeias and interjections, the language is identified using the Java library[5]. By knowing the language in which a message is written, it is possible

[2] Available at https://github.com/sing-group/bdp4j
[3] Available at https://github.com/sing-group/nlpa
[4] Available at https://en.wikipedia.org/wiki/List_of_emoticons
[5] Available at https://github.com/optimaize/language-detector

to replace abbreviations, contractions and also slang with text that represents their meaning.

As mentioned above, in order to match words with the synset that best represent their meaning in a given context, a disambiguation of the words is performed. There are words that have only one meaning independently of the context in which they are found, but this is not the case for other words. For those cases in which a word can be matched with more than one synset, Babelfy is used to identify the most appropriate one within the context of use of the word. For example, the word "bank" can have different meanings and can refer to a "blood bank", a "financial institution" or even a "sloping land". Each meaning refers to a different meaning or context and if the objective is to group synsets it is very important to identify the meaning, in order to match the synset in the best possible way.

In this data pre-processing step, all the text is sent to Babelfy, the disambiguator, which not only disambiguates the text, it also eliminates the stop words and returns only the relevant synsets. The obtained synsets also need to be processed because, although disambiguation has been carried out, some words can be grouped in different forms. As an example we can illustrate what happens when we try to disambiguate the word "neural network". In this case Babelfy returns three different answers, one corresponding to the meaning of the word "network", one corresponding to the meaning of the word "neural" and one corresponding to the meaning of the word "neural network". As it is possible to access the number of words that are part of the synset identifier, the one that groups the most words is always used, which in the case of the example corresponds to "neural network" and its synset.

4.5 Parameters configuration

In order to perform the process of synsets clustering, it is necessary to establish the classifier to be used and also the parameter γ, which indicates the maximum number of steps for the generalization. To carry out the classification process we will use the well-known Naïve Bayes classifier that has provided such good results [37, 38] in the spam classification and has reduced computational requirements. This classifier is implemented as a "Multinomial Naïve Bayes" classifier in the WEKA environment, which was the software used to run the experiments. This type of classifier assumes that the predictor variables are independent of each other, which allows it to select a higher number of independent features.

To identify the best value for γ an empirical evaluation has been carried out, in which different configuration values from 1 to 5 have been tested. For this purpose,

the complete YouTube dataset described in table 4.4 has been used. In the empirical evaluation, the dimensionality reduction and clustering procedure has been run several times and the solutions that are better than the original data with no preprocessing or clustering. To select the Pareto optimal solutions with most balanced tradeoffs among all objectives, the values of the distance to the origin of coordinates from each point has been evaluated. The coordinates of these solution are formed by the values of FPr and FNr values. The minimum distances obtained from each cartesian point (FPr, FNr) to the origin (0,0) can be seen in Table 4.5.

As a result of these empirical experiments, it has been identified that the dimensionality reduction system obtains its best results when a maximum of three generalization steps are performed, i.e. when γ=3. Taking into account these results, this value has been selected for the whole experimentation process.

Regarding the configuration of the MOEA algorithm and specifically the NSGA-II, this has been executed using its implementation in the jMetal framework. It has been configured to carry out 25,000 evaluations of the objective functions and each execution has to be carried for 25 times. We started from the default configuration of NSGA-II within JMETAL, with a population of 100 individuals, $1/NumberOfVariables$ mutation probability and a integer SBXCrossover and PolynomialMutation operators with 1.0 crossover probability.

γ	**Minimun distance**
0 (without reduction or topic merge)	0,394,946,621
1	0,322,903,916
2	0,319,093,112
3	0,312,350,806
4	0,319,875,171

Table 4.5: Geometric distance in relation with the value of γ.

4.6 Evaluation of classical feature selection methods

In this section we evaluate the performance of the new dimensionality reduction proposal, comparing the results obtained in two different types of scenarios. The first one is a situation where no feature selection method is used, that is, a situation where the words of the texts are transformed into tokens and there is no clustering or dimensionality reduction. The second is a situation in which a feature selection is performed using IG, in which a ranking can be performed and tokens can be selected based on a highest IG value contributing to the best classification. In order to be able

to compare both scenarios, Table 4.6 shows the accuracy values (%correctly classified) and the percentages of false positives FP and false negatives FN as indicators.

Token based classification				**Synset based classification**			
%Well classified	%FP	%FN	dimension	%Well classified	%FP	%FN	dimension
0.884	0.048	0.068	2279	0.828	0.02	0.152	1684

Table 4.6: Tokens and synsets classification results without feature selection.

The obtained results show that when synsets are used, the number of elements (the dimension) is smaller than when tokens are used. This is because (i) there are words that are synonyms and are only represented by a single synset that groups all of the words with the same meaning, (ii) in some texts there are misspelled or obfuscated words, which are used by spammers to evade filters making these words unable to be processed by classifiers (this problem is discussed in Chapter 6), (iii) some of the words are not in an ontological dictionary such as BabelNet, therefore it is not possible to find a synset to represent them.

Looking at the values in table 4.6, it can be seen that there exist differences between the values of FP and FN when using synsets and when using tokens. This is caused by the working procedure for the substitution of words by the synsets to represent their meaning, and in particular, by the information loss produced by the existence of some words that cannot be converted into synsets, as described above. The values in Table 4.6 shows that the use of synsets gives better results in the identification of legitimate messages, resulting in a lower FP error rate. In legitimate messages, misspelled or obfuscated words are almost non-existent. This indicates that removing terms or words that do not exist in a dictionary, improves the identification of legitimate messages (ham).

In order to validate the new dimensionality reduction and clustering proposal, an additional comparison has been performed between the novel proposal with the equal number of tokens identified using the traditional and popular IG based method for feature selection and filtering. Based on the use of IG the dimensionality of the dataset represented using tokens has been reduced. In order to make a comparison between the results using the IG and tokens or only synsets, with only the clustering dimensionality reduction due the synset nature, a classification process has been performed using a Naïve Bayes Multinomial. Table 4.7 shows the results obtained by this classification process. It was initially executed with the 2279 tokens, which corresponds to the maximum token value. It has also been carried out with the value of 1684, which corresponds to the maximum number of synsets. A lower value than the tokens due the phenomenon of loss of information originated by obfuscations or misspelled words

and by the clustering in a single synset of words with the same meaning. In order to be able to see the degradation suffered by each of those methods as the size of the information block becomes smaller and smaller, the values for 1500, 1000 and 500 tokens and synsets have also been calculated. The results shown in the Table 4.7 allow to note how the use of synsets reduces the number of FP errors and increases the number of FN errors when compared with the classical use of tokens and the use of IG for feature selection.

Dimension	**Token representation**			**Synset representation**		
	%Well classified	%FP	%FN	%Well classified	%FP	%FN
2279 (all of the tokens)	0.884	0.048	0.068			
2000	0.9	0.048	0.052			
1684 (all of the synsets)	0.888	0.056	0.056	0.828	0.02	0.152
1500	0.884	0.064	0.052	0.832	0.028	0.14
1000	0.884	0.072	0.044	0.888	0.012	0.1
500	0.896	0.056	0.048	0.868	0.02	0.112

Table 4.7: Token based classification results applying Information Gain for feature selection.

4.7 Performance

The reduction achieved in the number of features, the dimension, varies between 25% and 85% of the original size. There are solutions with remarkable values, such as the last of the solutions in figure 4.2, which obtains the lowest value in terms of the FNr indicator but maintains the other two at acceptable values. It is a solution with a good compromise between the three target values.

The results in figure 4.2 show that a reduction in the FNr value of 32%, i.e. lowering the value from 0.62 to 0.30, only generates a degradation of 8.2% in the FNr indicator, increasing it from 0.2 to 0.8. From the evolution shown by the indicators in figure 4.2, it can be seen that the value of FPr remains relatively stable despite the reduction process. It is the value of FNr that degrades as the dimensionality is reduced and the more generalizations are generated.

Once the classifier has been trained with 75% of the original dataset, the best results of the novel dimensionality reduction process are identified. When these results have been identified, their configurations are taken and applied to a new dataset, that is not previously seen by the classifier, that is, the 25% of the original dataset selected for the evaluation process. We performed a comparison between the 4 best results obtained in our reduction process with other 4 IG-based feature selection processes

using tokens. For this task a Multinomial Naïve Bayes model has been built using the same data from the training phase (75% of the original dataset). Once the token classifier was trained, we proceeded to classify the second 25%, in the same way as in our dimensionality reduction proposal. As we had used the same input data in both processes and also the same data for the test phase too, we generate two comparable sets of results. The selected configurations from the novel dimensionality reduction approach, are taken from the Cartesian points (FNr, FPr) evaluations that were closest to the origin of coordinates (0,0). Selected points were:

- FPr=0.074/FNr=0.303
- FPr=0.066/ FNr=0.306
- FPr=0.080/FNr=0.309
- FPr=0.057/FNr=0.3015

When the dataset is transformed and converted from tokens to synsets, the number of message wrongly classified as negative (FN) increases. However, the number of positive wrongly classified messages (FP) decreases. This can be seen in figure 4.6 which shows the values of the batting average. The batting average shows the proportion of spam messages correctly detected and consequently directly deleted or removed from the user's inbox (hit rate), and also shows the proportion of FP errors with respect to the total number of messages received (strike rate), that is, how many legitimate messages are marked as spam and consequently may not reach the destination. Evidently this allows us to understand that the best classifiers are the ones that obtain a high value in the hit rate and achieve low values in the strike rate.

The results obtained show that when the tokens are used, the hit rate is higher because more spam messages are detected, but the number of FP errors, strike rate, indicates that there are some problems and that their use should be discarded for some applications. To identify applications that should not rely on a token-based spam detection system, it will be compared with a cost-based performance indicator named Total Cost Ratio (TCR). Metrics such as TCR consider errors such as FP and FN in an asymmetric way. This is because they take into account a Lambda parameter that indicates how much worse it is to have an FP-type error than an FN-type error. In our comparison we have used the values of LAMBDA (1, 9 and 999) which are usually the values that model three different scenarios such as (i) a scenario in which jokes are exchanged (ii) a scenario in which the exchange of messages is used to receive e-commerce information and finally (iii) a scenario in which the information

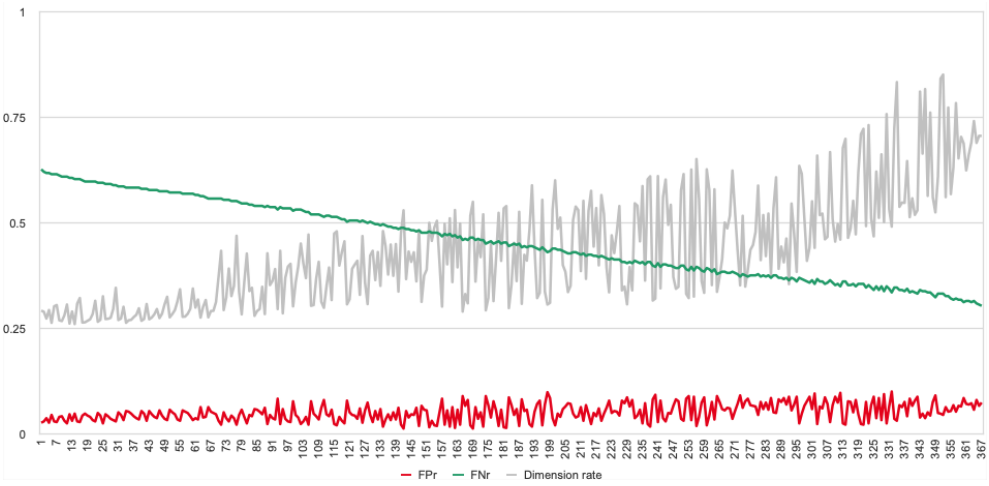


Figure 4.2: Multiple line chart representing solutions.

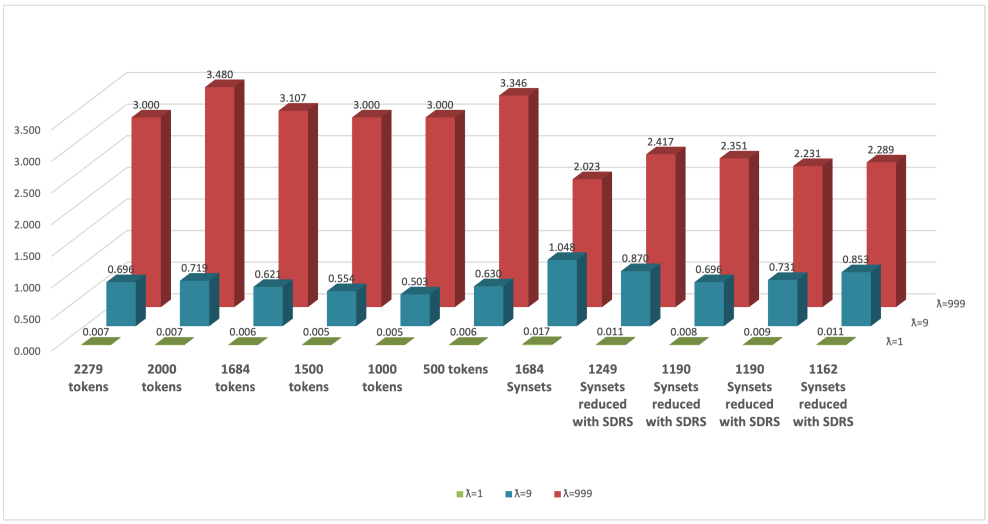


Figure 4.3: TCR benchmarking results.

exchanged is of a commercial or customer-related nature. The results of applying this TCR indicator to the different scenarios described above can be seen in figure 4.3.

Taking into account the results obtained in the experimental phase, it is possible to conclude that in a situation where the cost of FP and FN errors is the same, the use of synsets does not bring a great improvement. However, the proliferation of the use of electronic means to search for a job, the use of e-commerce facilities to purchase goods or services, and the use of electronic banking, makes clear that the cost of errors cannot be symmetrical. It is in these cases when the use of synset-based representations and the selection of features and grouping of terms is considered more appropriate. The analysis of the results obtained in these experiments suggests that

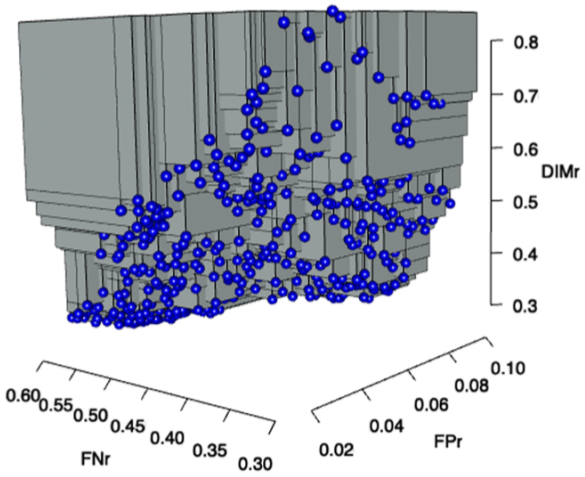


Figure 4.4: 3D Pareto front.

the use of synsets and dimensionality reduction method improve classification ratios.

From a content classification and topic identification point of view, the taxonomic clustering of words makes it possible the reduction of the number of topics, limiting the range of possibilities and making it easier for the user to select the messages related to a given topic. For example, all messages related to "anti-impotence drugs" could be in the same group, which, depending on the user's preferences, could be considered a topic that should be filtered out as spam. In terms of intentionality, the study makes it possible to create categories and relate them to bags of words, which, if they coincide with the words in the messages, could show the relationship between the message and the category created earlier.

From a practical point of view related to the performance of the classifiers, a reduction in the number of redundant words, which are grouped into a single feature, results in a reduction in time requirements and especially reduces the computational resources needed to carry out the training phase. On a theoretical level, the combination of highly dependent features results in a reduction of intermediate features, which allows some classifiers, such as the Naïve Bayes classifier used in the experiments, to increase their performance.

Experimental results show that the detection of legitimate messages is better when using synsets than when using tokens. This can be seen in figure 4.5, where it can be observed how the number of FP errors decreases when using synsets, causing a slight increase in FN errors. Although the difference does not seem to be significant, when considering the asymmetric cost of each type of error, as can be seen in the TCR image, the result is clear. This is caused because legitimate messages usually do not contain misspellings or obfuscations. This allows us to match practically all the

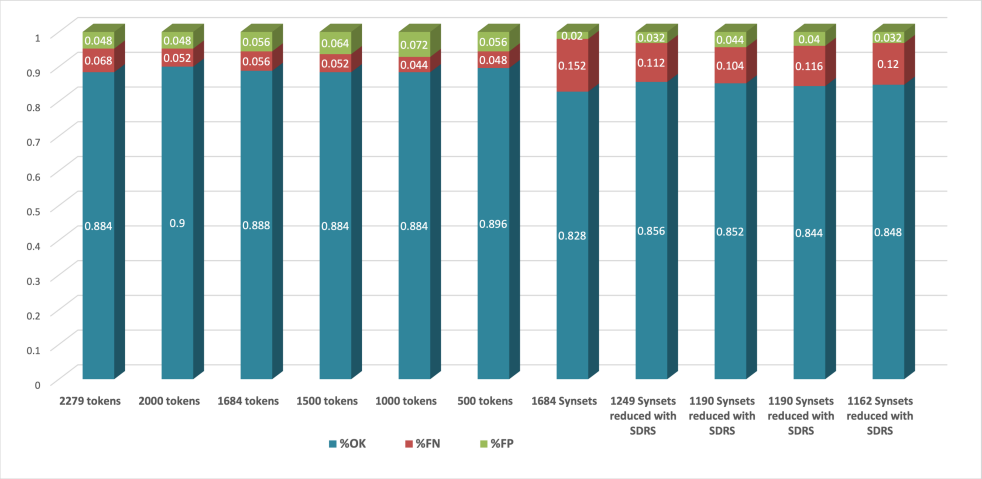


Figure 4.5: Performance comparative of different feature-reduction schemes.

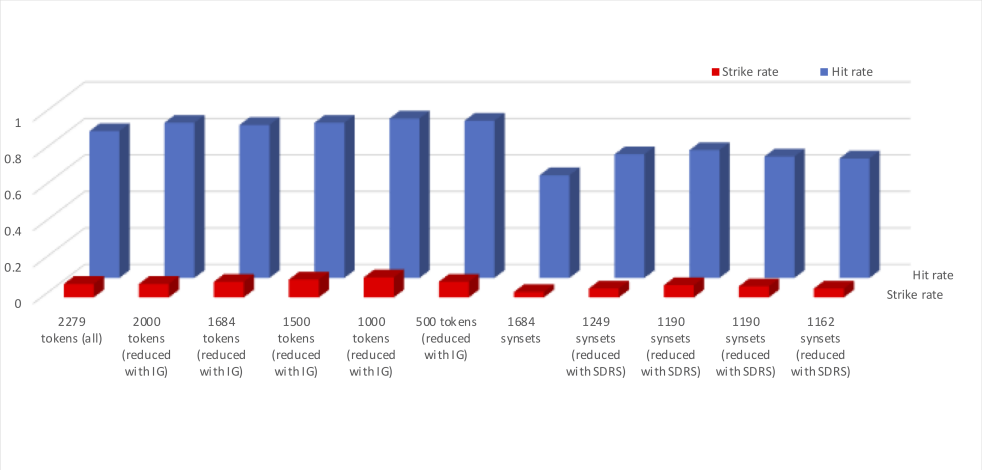


Figure 4.6: Batting average using tokens and synsets.

words that compose the messages with their equivalent synset, and allows for a greater number of synsets for this type of messages, allowing a better classification. At the other side there are spam messages which are often misspelled, contains obfuscated characters and often contain URLs. As a result of the existence of these types of elements, some of them cannot be translated into a synset, which gives fewer synsets and therefore less information to classify these types of messages, resulting in lower performance in the classification of messages. The possibility to identify a larger number of words in spam messages in order to improve their classification is discussed in Chapter 6, which addresses the identification of obfuscated characters.

4.8 Conclusions and future work

Based on the work performed and the results obtained in this study, which was initially focused on the grouping of specific terms into more general terms using synsets and their semantic relations to reduce the number of features, it has been demonstrated that the performance of spam filters can be improved if messages are represented using synsets over tokens. In this study, an experimental comparison has been made in terms of the performance that can be obtained by classifiers implemented in spam filters using a synset-based representation and exploiting their semantic relations. A new system for dimensionality reduction using a reduction scheme based on MOEA algorithms has also been proposed and presented as a multi-objective optimisation problem. The results obtained allow us to conclude that the approaches based on semantics and using synsets, obtain higher performances in the environments in which the FP and FN errors have an asymmetric cost. This is because with this approach the rate of FP errors decreases, which results in fewer messages being lost and identified as spam and therefore filtered. This corresponds to most of the message usage environments, where a spam message that is not blocked by the filters creates a small damage, but a legitimate message that is filtered and not read can result in a large problem.

This feature selection and dimensionality reduction system achieves dimensionality reduction without loss of information, because it does not eliminate any synset. The information contained in each word or synset is not eliminated since it is combined in a more general feature, making it possible to combine different synsets in a single concept. In Chapter 5 we present a variant of this dimensionality reduction method with an information loss or a customizable elimination of the terms. In this way, it will be possible to carry out a controlled information loss, trying to eliminate the less relevant features.

The main problem with the proposed feature reduction system is the training time, which, as can be seen in Table 4.3 in the computers where the system has been executed is no shorter than 12 days. This is caused by the multitude of evaluations and executions performed in the optimization process by the evolutionary metaheuristics of the MOEA algorithms. However, once the model has been trained, it is no longer necessary to re-run the training phase. This allows dimensionality reduction of the classified problem and identification of relevant features without loss of information to be used without problems for spam filtering.

Chapter 5

Low-loss dimensionality reduction approach

In the Chapter 4 a novel proposal for dimensionality reduction and clustering based on synsets has been presented. This strategy uses a MOEA to identify the number of levels to generalise a synset in order to maintain and in some cases even improve the performance in the classification of spam messages. In order to identify the features to generalise, the NSGA-II algorithm has been used, this algorithm provides a chromosome in the form $C = \{1, 0, 3, 4, 0, 2, 3, ...c_n\}$ where n is the number of features and the numbered position of the chromosome means the number of levels to generalise the synset. To perform this process, a set of three fitness functions have been proposed to evaluate each configuration. However, this method has a limitation because there are some synsets that do not provide information for classification but are still maintained as features, in other words, they are not eliminated. These features can even generate "noise" and their elimination could improve the performance of the classifiers. This chapter presents a dimensionality reduction algorithm with the possibility of eliminating some terms from the feature vector, so that the terms that do not contribute to the classification can be deleted.

5.1 Introduction

This new proposal maintains the same synset representation used in the previous Chapter 4. Each text message is represented as a vector of integer values in the form of $T = \{a_1, a_2, a_3, ..., a_n\}$, in which the value of each element of the vector represents the number of occurrences of a specific synset-based attribute ($A = \{a_1, a_2, ..., a_n\}$). Also for each text message there is a decision variable d that represents if the message was in the ham or spam class. All of this information is stored in the M matrix, which is required to work with the data both in this new proposal and in the previous one.

This novel feature reduction scheme can perform three different actions: (i) the clustering of two or more related attributes, (ii) the elimination of some attributes, (iii) both options. We may recall that in the previous proposal only the first action was performed, which corresponds to a lossless reduction scheme. The second option corresponds to a information loss reduction scheme and the third one corresponds to a low information loss scheme. Based on previous work [76, 28], attribute clustering strategy (i) uses the taxonomic relationships of hypernyms and hyponyms that exist between synsets.

In this section we describe the dimensionality reduction problem from the point of view of multi-objective optimization, in the same way as the cases mentioned above. In our case the optimization problem is formulated in a form to be solved by evolutionary algorithms. The objective is to achieve a vector of integers $V = \{V_1, V_2, ..., V_n\}$ that indicates the action to be taken with each of the synset that are used as attributes with the aim of minimizing the number of features and still achieving the highest performance of the classifier. Since the objective is very ambitious, it will be simplified into two objectives problem, where objectives are (i) to reduce the number of FP errors and (ii) to reduce the number of FN errors. To achieve this, the evolutionary algorithms have been configured to give us a V value indicating the action to be taken with the synset. These values are integers that can be from -1 in which case the synset is eliminated, passing through the value 0 that indicates that the synset should be kept as it is, up to a value $m(0 < m < \gamma)$ where γ is the number of jumps to perform the generalization.

The objectives involved are in conflict [12] because reducing the number of features can generate an increase in FP and FN errors, while trying to minimize FP and FN errors implies an increase in the number of features, all of this despite the fact that the FP and FN errors themselves are conflicting objectives. In order to work with the three objectives, they have been defined as three fitness functions that have to be minimized simultaneously, which are presented in equation 5.1.

$$\begin{aligned} f1 &= \frac{col\_num(T(M,V))}{col\_num(D)} \\ f_2 &= 10xval\_eval.FP_r(c, T(M,V)) \\ f_3 &= 10xval\_eval.FN_r(c, T(M,V)) \end{aligned} \tag{5.1}$$

In equation 5.1 $T(M,V)$ is the transformation of the dataset presented by Matrix M applying the changes of the vector V, $10xval\_eval.FP_r(c, T(D,C))$ represents the False Positive ratio and $10xval\_eval.FN_r(c, T(D,C))$ represents the False Negative

ratio. FPr and FNr results are calculated using a 10-fold cross validation scheme of the classifier c, when is applied to the matrix M that represents the dataset.

Transforming the matrix of the dataset M following the transformation vector V involves the following steps: (i) remove columns marked for deletion with a value of (-1); (ii) generalise the attribute or synset for as many levels as indicated by the integer value for that column in the transformation vector; (iii) group columns and add the values of their attributes. The first step consists in the identification of the columns where the attribute a_i is marked with a value of $-1(V_i = -1)$, this implies that the column a_i is eliminated for all the instances of the dataset represented by the M matrix. In the second step, the remaining a_i attributes are identified and replaced by their upper hypernyms of the level v_i specified by the vector V, identified by the value $(vi > 0)$ of the generalization. This leads to attributes semantically close to the original form of a_i become direct or indirect hyponyms of its transformation. In the last step, a set of attributes $A' \subset A, A' \cap \{a_i\} = \varnothing$ are merged into the same attribute a_i if and only if $\forall a'_i \in A', a'_i \in$ hyponyms(a_i) is the set of direct and indirect hyponyms of a_i.

The formulation proposed in this research work allows to represent different dimensionality reduction schemes: the proposal with information loss based on feature elimination is performed only with v_i values of -1 and 0, i.e. $-1 \leqslant v_i \leqslant 0$. The lossless scheme is performed when there is no term elimination and the values of the vector are positive or equal to 0, i.e. $0 \leqslant V_i \leqslant \gamma$. Finally there is the low-loss strategy in which there is no limitation to generalise or remove terms so the values of the changes vector can be any value between $-1 \leqslant v_i \leqslant \gamma$.

Once the different methods for dimensionality reduction have been presented and described, we can question if pure lossless approaches are the most adequate to address dimensionality reduction when using synonym-based classification schemes. Before obtaining the final results of this work, lossless schemes [28] were considered the best way to address the problem. However, in this new study we have found evidence that some synsets do not provide relevant information, even when they remain unchanged or when they are merged with others and may degrade the classification performance by the noise they generate. This discovery has driven us to consider that a reduction scheme without loss of information may not be the best solution and the possibility of deleting some synsets should be studied, because the elimination of some synsets allows us to obtain better classification performance.

The idea of this new low information loss proposal is based on the idea that the dimensionality reduction process should allow in its optimization phase not only the clustering of terms or synset, it should also allow the possibility of removing synsets.

The following section shows a detailed comparison between the lossless method described in the previous Chapter 4 and the new low-loss proposal.

5.2 Experimental protocol

In this section we present the designed experimental protocol, as well as the selected configuration values, in order to compare the performance of the low loss dimensionality reduction and lossy dimensionality reduction schemes, ensuring that the phenomenon of ovetfitting does not occur. The dimensionality reduction scheme presented here, in contrast to the reduction scheme described in the previous Chapter 4 and considered lossless, not only performs the generalizations of the synset, it also has the ability to eliminate them.

A major drawback of both schemes is that both could result in over-fitting, making the classifier fit excessively to the input data and work perfectly with it but producing many classification errors when working with other data. In order to avoid this problem at this step of the machine learning process, the dataset has been divided into two subsets, one with 75% of the data and another with 25% of the data. The 75% of the data will be used in the training phase and the remaining 25% will be used in the model testing phase. Figure 5.1 provides a graphical diagram of the experimental protocol.

As can be seen in figure 5.1, at the beginning, the experimental protocol calculates the dimensionality reduction of the two schemes, both the lossless and the low-lossy schemes, and at the end compares the results obtained by each one. It can also be seen that the input data of both schemes are exactly the same, using the subsets of 75% and 25% as explained before. At the end of each of the experiments, the best five chromosomes, the values that obtained the lowest FPr and FNr error rates provided by the genetic algorithms for each of the schemes, were selected for use in the last phase of the experimental protocol. In this last step, a test step was performed for each of the schemes, five executions with the five best chromosomes for the lossless scheme and the same for the low-loss scheme, using in both cases the 25% of the input dataset reserved for the test phase. Keeping 25% of the input data for the test phase makes it possible to evaluate the performance of the classifier when using data that is new to it, and also allows for the detection of the overfitting mentioned above.

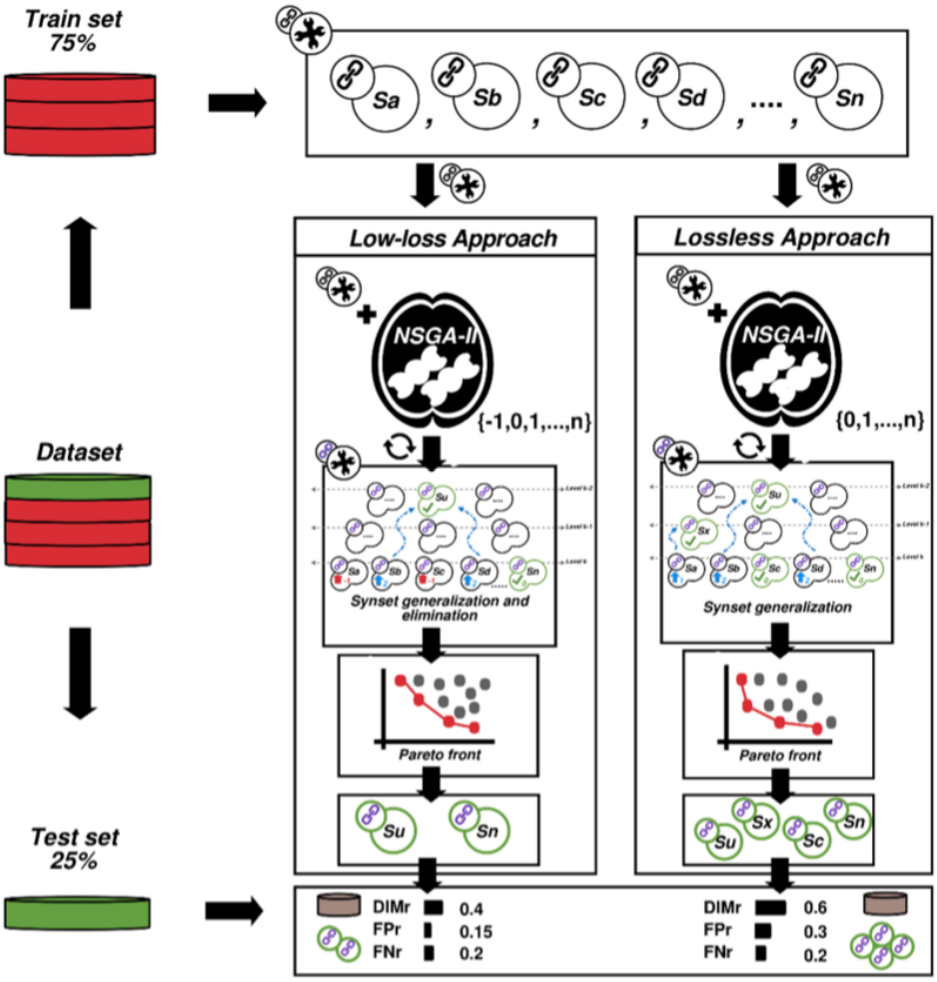


Figure 5.1: Low-loss approach experimental protocol.

5.3 Dataset selection

As mentioned in several recent works [84, 28, 106], there are many datasets that can be used to test new methods or techniques for spam message classification. However, given the new dimensionality reduction proposal is based on the same genetic algorithms and working principles used in the previous Chapter 4 proposal, it has been decided to use the YouTube spam Collection dataset for the low-loss dimensionality reduction approach similarly to the previous proposal. This is mainly motivated because, although the low-lossy scheme some features can be removed completely and this speeds up the computational process by generating a fast reduction in the size of the feature vector, the computation demand for this approach is still very high. The evaluation parameters of the genetic algorithms, which are stochastic

Values for V_i	Minimum Euclidean distance
0 (without optimization)	0.3949
[-1,0] (low-loss approach)	0.3157
[-1,1]	0.3241
[-1,2]	0.3120
[-1,3]	0.3148
[-1,4]	0.3685

Table 5.1: Minimum Euclidean distance depending on the gamma value.

methods, are still 25 executions with 25000 evaluations of the fitness functions for each execution. The use of the same dataset provides the possibility to compare the execution times (which as expected are about 10% shorter than in the lossless scheme), results and performance of the new proposal, with the lossless scheme.

5.4 Optimization process configuration

In the same way as with the dimensionality reduction scheme in Chapter 4, this low-loss reduction method has also been formulated as a multi-objective optimization problem, and the NSGA-II genetic algorithm available in jMetal framework has been used. During the experimentation process, 25 independent runs with a maximum of 25000 objective functions evaluations were executed. The default settings of jMetal were used, namely, the population of the NSGA-II algorithm is set to 100 individuals, $1/NumberOfVariables$ mutation probability and an integer SBXCrossover and PolynomialMutation operators with 1.0 crossover probability. Multinomial Naïve Bayes implementation from Weka [112] was used as the classifier to compute fitness functions (FPr, FNr).

In this low-loss dimensionality reduction scheme, in order to select the best values for the γ parameter, which indicates the maximum number of generalization steps for a synset, an empirical evaluation has been performed. These values were calculated using the YouTube Spam Collection Dataset. To identify the best quality solutions, the Euclidean distance to the origin of coordinates from each point composing a possible solution with the format (FPr, FNr) has been used as a metric. Table 5.1 shows the distance of the closest solution to the origin of coordinates for each of the γ values used.

Table 5.1 shows that the best result for the Euclidean distance to the origin of coordinates corresponds to the value of $\gamma = 2$ i.e. $(-1 \leqslant V_i \leqslant 2)$. For the lossless dimensionality reduction study developed in the Chapter 4 the value of gamma was 3, making the values of V_i in the range [0,3].

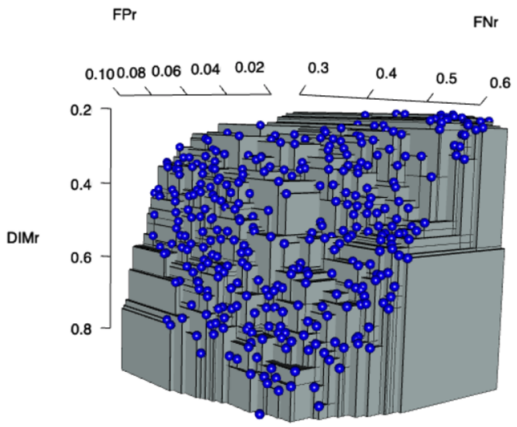


Figure 5.2: Pareto front of low-loss scheme.

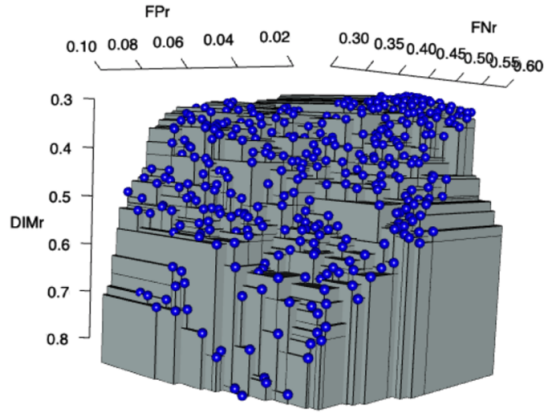


Figure 5.3: Pareto front of lossless scheme.

5.5 Results

The experimental protocol has been performed with the parameters shown in the previous section. The results provided by the NSGA-II genetic algorithm for the low-loss scheme were analized in detail and compared with the results of the proposal of Chapter 4. Figure 5.2 and 5.3 shows the Pareto Fronts generated by both schemes in order to see the differences.

Comparing the figures of the two dimensionality reduction schemes, we can see that the non-dominated solutions of the low-loss approach are more evenly distributed than in the lossless approach. This means that the 25 runs of the low-loss experiment, each one with 25,000 evaluations, have explored a larger solution space. The major difference between the two graphics is on the axis which represents the Dimensionality Reduction ratio (DIMr) where in the low-loss scheme the value has a lower value

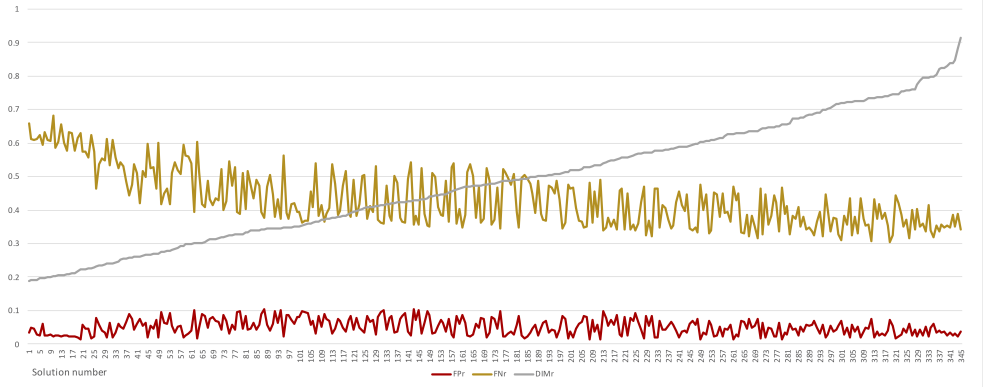


Figure 5.4: Low-loss approach solutions sorted by DIMr.

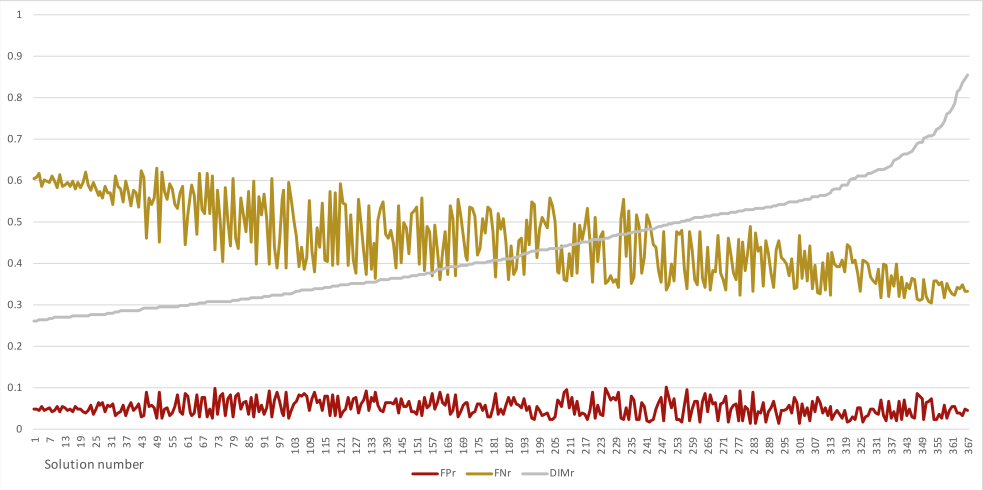


Figure 5.5: Lossless approach solutions sorted by DIMr.

of 0.2 and in the lossy scheme the lowest value is 0.3. In order to compare the two approaches with more details, all the non-dominated solutions have been plotted in figure 5.4 and figure 5.5, sorted by the DIMr indicator.

Figure 5.4 and figure 5.5, which correspond to the low-loss and lossless schemes, respectively, show that the FPr values are very close to 0 for all of the evaluated configurations, moreover, for most of the solutions, the FPr value is below 0.1. The values of the FNr are higher, they are in the range from 0.3 to 0.7. The most important difference can be seen in the DIMr value, which in the case of the low loss scheme (figure 5.4), shows better values, some of them even under 0.2 and with a low impact on the FNr rate. Observing the behaviour of the FPr values, it can be noticed that it has a more or less independent behaviour with respect to the DIMr value, revealing that optimality conditions can be preserved by DIMr and FNr trade-offs.

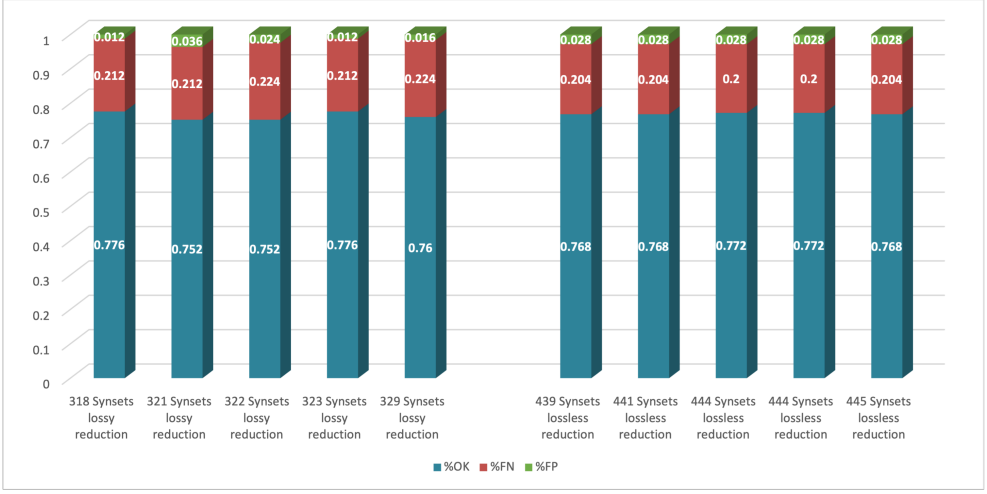


Figure 5.6: Top five configurations achieved by low-loss and lossless reduction schemes.

The five configurations of each scheme with the best results on the DIMr parameter were compared using the remaining 25% of the corpus. In all cases the Naïve Bayes Multinomial classifier model was generated with 75% of the data and tested with the remaining 25%. Figure 5.6 shows graphically the comparison of the ten results (5 low-loss and 5 lossless).

Looking at the values in terms of FPr and FNr we can see that the performance of both schemes is similar. However, in DIMr the low-loss approach obtains lower (better) values than the lossless approach.

For a better understanding of the new low-loss approach results, a study of the terms that are eliminated has been carried out, with the aim of verifying if the terms that are deleted are irrelevant or noisy. Table 5.2 shows the ratio of the solutions in which the algorithm has determined that a synset should be deleted. In order to check the relevance or not of the synset, a column with the IG value of the synset has been added. Table 5.2 shows only the 10 synsets that have been most often marked for elimination in the solutions included in the Pareto front of the low-loss scheme, which are the synsets with the lowest IG value (IG value of 0 for all synsets). Those synsets that have been eliminated correspond to names of people such as "Patrik" or adjectives such as "sad", "human" or "illegal", which might be present in spam and legitimate texts. In contrast, the synsets of table 5.3 show the terms maintained by the algorithm and not marked for elimination. Here it can be seen that these synsets are the ones with the highest IG value, which makes them relevant for the classification process.

The ratio of the synsets that have been marked for deletion in Table 5.3 is much lower than in Table 5.2. The IG value of these synsets, higher than the synsets marked

Synset	% of solutions where the synset is marked for removal	Information Gain	meaning
bn:00110036a	0.2361	0	sad
bn:14838845n	0.2278	0	Patrik
bn:00104384a	0.2115	0	human
bn:00082684v	0.2153	0	dress
bn:00085250v	0.2112	0	code
bn:00061695n	0.2029	0	perry
bn:00104562a	0.1988	0	illegal
bn:00086717v	0.1988	0	detest
bn:00083286v	0.1988	0	happen
bn:00076203n	0.1946	0	tatto

Table 5.2: Top 10 synset marked for removal, IG and meaning.

Synset	% of solutions where the synset is marked for removal	Information Gain	meaning
bn:00017681n	0.0455	0.0903	channel
bn:00094545v	0	0.087	subscribe
bn:00008378n	0.0289	0.0354	cheque
bn:00088421v	0.0165	0.0233	follow
bn:00032558n	0.0248	0.0226	eyeshot
bn:00066366n	0.0414	0.0184	subscriber
bn:00103299a	0.0372	0.017	free
bn:00094547v	0.0414	0.0155	take
bn:00042306n	0.0124	0.0129	Guy
bn:00055644n	0.0331	0.0123	money

Table 5.3: Top 10 synsets that are maintained and not removed, IG and meaning.

for deletion, shows that they are relevant in the classification process and should be preserved. Looking at the meaning of the synset, it is remarkable ttonote that the meaning of the first two synsets, corresponding to "channel" and "subscribe", which exemplify a message of a spam-like text, aimed to make people subscribe to a certain channel. In this case we can observe how for the second synset, the elimination ratio is 0, which shows that it is a relevant and is not marked for elimination by the MOEA.

Finally, an analysis of the probability of a synset being chosen for elimination as a POS has been performed. Table 5.4 shows the presence rates of each POS in the corpus, the probability of keeping a synset when it has a given POS and the distribution of the Information Gain for each POS.

Looking at Table 5.4 we can confirm that the MOEA is selecting the synset with

synset type	Composition dataset %	Probability of maintenance	Accumulated IG %
a (adjective)	14.01	0.1175	7.78
r (adverb)	2.25	0.0219	2.39
n (noun)	63.24	0.2474	62.27
v (verb)	20.48	0.1630	27.54

Table 5.4: POS analysis of results achieved by the low-loss approach.

the highest IG to be preserved and the synset with the lowest IG value for elimination. The nouns are the elements that have the highest IG value and are also the elements that have the highest non-elimination rate. The values in table 5.4 also allow us to conclude that the elements that provide the highest information to the spam classification process are nouns and verbs.

5.6 Conclusions

This research work introduced the formulation of three different strategies to reduce the dimensionality of the feature vector when using synsets. One is the existing strategy developed in Chapter 4, a new one with low-loss information and another one with information loss. In the same way as the first method, the new two schemes have been formulated as multi-objective optimization problems too. In order to identify the best dimensionality reduction method, additionally to the development of new approach, a comparison between the lossless and low-loss dimensionality reduction approaches has been performed. The new low-loss approach has been implemented using MOEA's and it has been confirmed that the synset marked by the algorithm for elimination are the synsets with the lowest IG value. The results allow us to claim that lossless feature reduction schemes can be successfully complemented with reduction schemes, to identify irrelevant or noisy synsets, reducing the training time of the classifiers as well as the computational requirements.

The results obtained in the experiments show that a low-loss scheme achieves a higher dimensionality reduction than a lossless scheme with almost identical classification quality in terms of FP and FN errors. Furthermore, allowing the MOEA to eliminate feature columns (low-loss approach) allows to explore a larger solution space as can be seen in the Pareto front in figures 5.2 (low-loss) and 5.3 (lossless). This is because the combination of noisy features with other relevant features (that have not been previously removed) would probably lead to a reduction of the classification performance. Overall, we can conclude that removing some features produces

a greater reduction in dimensionality but maintains very similar values of classifier accuracy.

For the lossless reduction scheme, the only possibility to reduce the number of features is the combination of two or more synsets into one, which requires looking through the hypernyms to find a common element that allows this reduction. However, in BabelNet it is only possible to generalize by the means of hypernyms in the case of verbs and nouns, so in the case of adjectives and adverbs they cannot be grouped and therefore this type of features cannot de reduced. Despite this, these words can sometimes be successfully removed without affecting the classification results.

Chapter 6

Decoding Leetspeak obfuscated words

The evolution of anti-spam filters is forcing spammers to develop new techniques to bypass the filters. The distribution of text content embedded in images or the use of Leetspeak are two techniques that hide text from content-based filters. Due to the difficulty of addressing these problems, the existing works on this issue are few and the results obtained are limited. This work proposes a new technique based on neural networks to decode Leetspeak. In addition, a series of specific datasets have been generated to address this problem, such as a database created to train Leetspeak decoding models and four datasets, with Leetspeak characters to test the performance of the decoding systems. Using these elements, the new proposal for decoding Leetspeak has been tested experimentally. The results obtained show that the proposed model is useful and can help in solving the problem of decoding texts containing Leetspeak characters.

In this study, we introduce new computer vision approach based exclusively on the use of a CNN model [108, 61] to decode Leetspeak. It is able to accurately identify sequences of Leetspeak encoded characters represented as images. Using this approach, we are able to recognize the obfuscated words and thus make the full text available to the spam filter. For the implementation, we used TensorFlow [1] and Keras [51]. Our contributions are: (i) an image database created for training CNNs for Leetspeak deobfuscation, (ii) an empirical demonstration that Leetspeak recognition can be accurately performed using only CNNs and (iii) datasets for the evaluation of Leetspeak decoding schemes.

6.1 Materials

Currently there is no publicly available dataset containing obfuscated text with Leetspeak. In order to perform this research work and to test the performance of the generated models, a dataset with these characteristics is required. The process followed for the generation of this dataset is described in section 6.1.2. A set of images is also needed to train the neural network, in other words, to create the Deep Learning (DL) model that identifies the obfuscated characters. To carry out this task, a set of images has been created, and its construction is described in section 6.1.1.

6.1.1 Image database for training

This work introduces a computer vision based approach using Deep Learning to identify text with obfuscated characters. In order to generate a model that can decode Leetspeak characters, it is necessary to train the model with a set of labelled images in which the system can identify the character represented on each case. Based on the work of Tundis et al. [101] we have used the Chars74K [27] dataset as a reference. However, this dataset is oriented to the identification and recognition of characters in real-life images, so it does not exactly fit our problem. To test its validity in the identification of obfuscated characters, several models have been trained using Chars74K and then an analysis of the identification of obfuscated characters has been carried out with Leetspeak characters, obtaining prediction rates with values between 42% - 52%. Given these results, it has been decided to create a set of images to train the model more effectively in order to obtain better results in the identification of obfuscated characters.

Our image database has been generated using 158 fonts and the styles regular, italic, bold and bold+italic, in which the characters from "A" to "Z" have been plotted. The images of these characters have been made in a size of 100x100 pixels. Figure 6.1 shows as an example some of the images that form the set of the image "A".

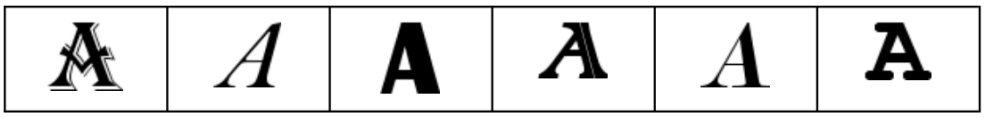


Figure 6.1: Images that are part of the set of the character A.

6.1.2 Generated datasets for deobfuscation model evaluation

This section describes the process followed to obtain a corpus with spam and ham text messages containing obfuscated characters. This corpus was used to evaluate the

performance of the proposed deobfuscation approach, and to compare the results obtained with other approaches. The generated corpus containing obfuscated characters, is based on other public and widely used corpora. In Table 4.4 it is shown a collection of public datasets made available by the scientific community working on the spam problem that we used as a starting point to carry out the obfuscations.

Table 4.4 shows a wide variety of datasets with different sizes and contents that are publicly available to test the performance of anti-spam solutions. In order to compare the results with a previous study by Ezpeleta et. al. [39] two datasets with YouTube messages have been selected, YouTube Comments Dataset and YouTube Spam Collection Dataset. As in the previous study, the complete YouTube Spam Collection dataset was used and a subset of 4000 messages (1000 spam and 3000 ham) was generated from the YouTube Comments Dataset. In this way we use the same YouTube Comments Datasets and it makes it possible to compare the results. Additionally, in order to extend the study to the email domain, the CSDMC2010 Spam Corpus datasets and a subset of 4327 messages (32% are spam) from the TREC 2007 Public Corpus have also been used. This has resulted in two email corpora with the same number of messages and also the same ham/spam ratio.

Once the datasets were selected for the experiment, we proceeded to obfuscate them. To perform this obfuscation task, an algorithm was designed using some of the Leetspeak replacements shown in Table 2.10. The designed obfuscation algorithm performs the following actions: (i) randomly select a word from a group of seven, (ii) randomly select a character within that word, (iii) randomly select a substitution from 2.10 and apply it on the character, (iv) repeat the process for the next seven words until the end of the text.

When a human is writing using Leetspeak, it is very probable that the substitution sequences are continuously repeated, for example, only the "4" is used for all substitutions of the character "A". However, when obfuscations are made by an automatic system in order to evade spam filters, the changes are made with random substitutions, in line with our algorithm behaviour. The obfuscation method presented in this section performs random substitutions of characters according to Table 2.10.

The four obfuscated datasets generated (YouTube Comments Dataset Leetspeak, YouTube Spam Collection Dataset Leetspeak, CSDMC 2010 Leetspeak, TREC 2007 Public Corpus Leetspeak) have been shared in a public repository on the website of Mondragon Unibersitatea (https://mondragon.edu) and [29].

6.2 Methods

Our new proposal is based on the use and application of Deep Learning to identify Leetspeak obfuscated characters. This section describes the identification process of obfuscated sequences with Leetspeak, and the discovery of the correct alphabet character. Section 6.3 describes the designed experimental protocol to evaluate the new proposal.

6.2.1 Leetspeak sequence identification

The deofuscation problem consists in matching the Leetspeak sequence with a text character, and the character must be the graphically most similar to the one that is being substituted. The detection of Leetspeak character sequences is performed by detecting non-alphabetic characters embedded between the words that form the text. Specifically, it is necessary to search for the first non-alphabetic character and the last non-alphabetic character, considering everything between these two in the Leetspeak sequence.

When a Leetspeak sequence is detected, it is isolated from the other text and an image is generated from it for posterior identification of the character it replaces. Table 6.1 shows some examples of obfuscated characters, the generated image and the text character matched by the DL model.

Obfuscated character	Generated image	Matching character
|_	|_	L
†	†	T
€	€	E
|	|	I
[\]	[\]	N

Table 6.1: Obfuscated characters examples.

Once the image is generated, it is processed by the Neural Network to identify the obfuscated character and pair it with the best matching text character. The identified character replaces the obfuscated character and the word is rewritten using only text characters. This process is repeated until all words in the dataset have been parsed. Once the text messages have been deobfuscated they are classified. The following section explains the Deep Learning scheme used to decode the Leetspeak sequences.

Layer	Convolution
1	Conv2D (filters 32, kernel_size(3,3), activation_function=relu, stride=(1,1) MaxPooling2D ppolsize=(2,2)
2	Conv2D (filters 64, kernel_size(3,3), activation_function=relu, stride=(1,1) MaxPooling2D poolsize=(2,2)
3	Conv2D (filters 128, kernel_size(3,3), activation_function=relu, stride=(1,1) MaxPooling2D poolsize=(2,2)
Dropout (0,7)	
Flatten ()	
Dense (neurons 512, activation_function=relu)	
Dense (neurons 26, activation_function=softmax)	

Table 6.2: CNN layer details for obfuscated character recognition.

6.2.2 Character identification model

For the identification of the character that corresponds to the obfuscated one, an image recognition system is implemented (not based on a static dictionary), providing the advantage of being able to recognize new variants of Leetspeak. Matching an obfuscated character with a corresponding text character involves looking at the signs, punctuation marks and numbers that compose it to find a visually compatible text character. The sequences used to encode a character can be of different lengths, for example, the character 'i' can easily be replaced by an inverted exclamation mark '¡'. However, the letter H may require three punctuation marks, such as '|-|'. Taking this into consideration our new proposal includes a CNN to identify the characters. Table 6.2 provides detailed information on the layers that compose the design of our proposed CNN.

As shown in table 6.2 the CNN has been defined as a stack of alternating layers of Convolution, ReLU and MaxPooling. The input data are 100x100 pixel images with a colour depth of 1 byte and the output layer consists in 26 neurons and a "softmax" activation function that calculates the probability of identifying a specific text character.

The CNN model has been trained with the image dataset described in section 6.1.1, keeping 20% of the total images for the validations that have been performed throughout the 15 training epochs. In addition, the possibility of adding an early stop as a callback in the training process has been considered to reduce overfitting. However, it was decided to train the model on a certain number of epochs, as the model will try to predict obfuscations formed by characters, but it will only be trained with different variations of real letters. Therefore, in this case it is not essential to apply an early stop to avoid overfitting the model. Figure 6.2a shows the CNN accuracy and Figure 6.2b shows the CNN loss measurements for training and validation phases.

Figure 6.2a shows the accuracy obtained by the CNN model in each epoch for the training dataset and validation subset. Figure 6.2b shows the acquired loss evolution for each epoch. As can be seen, after 10 epochs we obtain an accuracy near to 90%.

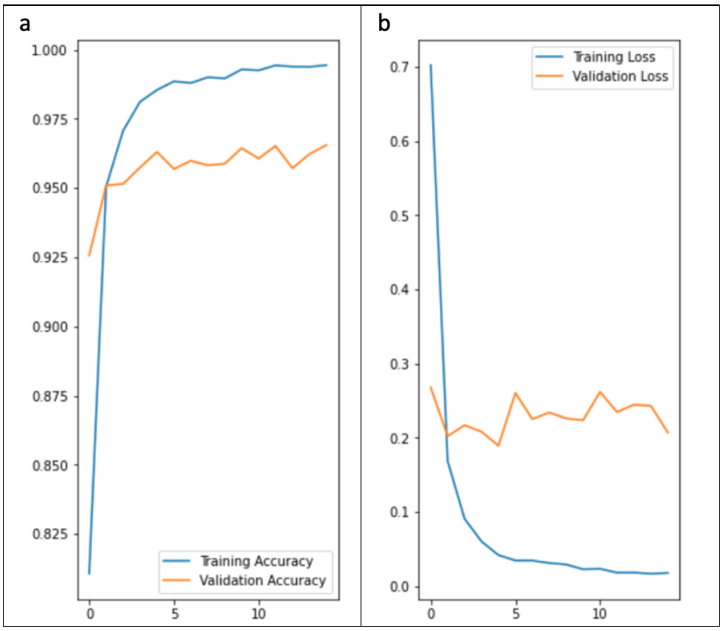


Figure 6.2: Training and validation accuracy and loss.

In the next epochs the increase in accuracy is slower (the neural network needs many extra epochs to achieve small improvements in accuracy).

6.3 Experimental protocol

In order to evaluate the performance of our CNN-based system in a real environment, we have generated different datasets with obfuscated characters (see section 6.1.2). The evaluation has been carried out using the experimental protocol illustrated in figure 6.3 designed for this purpose.

As can be seen in Figure 6.3 the experiment is comprised of 5 stages in which different aspects are evaluated: (i) the CNN, (ii) the classification of the original dataset (baseline), (iii) the classification of the dataset after performing the cleaning procedure, (iv) the classification of the dataset with obfuscations, (v) the classification of the dataset once it has been deobfuscated.

The first step consists in the CNN training process and its evaluation by deobfuscating Leetspeak characters. In the second step, the messages have been classified in their original form to obtain a baseline for future comparisons. Due to the large number of existing classifiers, we selected the classifiers that obtain the best classification values for the YouTube Spam Collection Dataset and the subset of YouTube Comments Dataset based on the work of Ezpeleta et. al. [39].

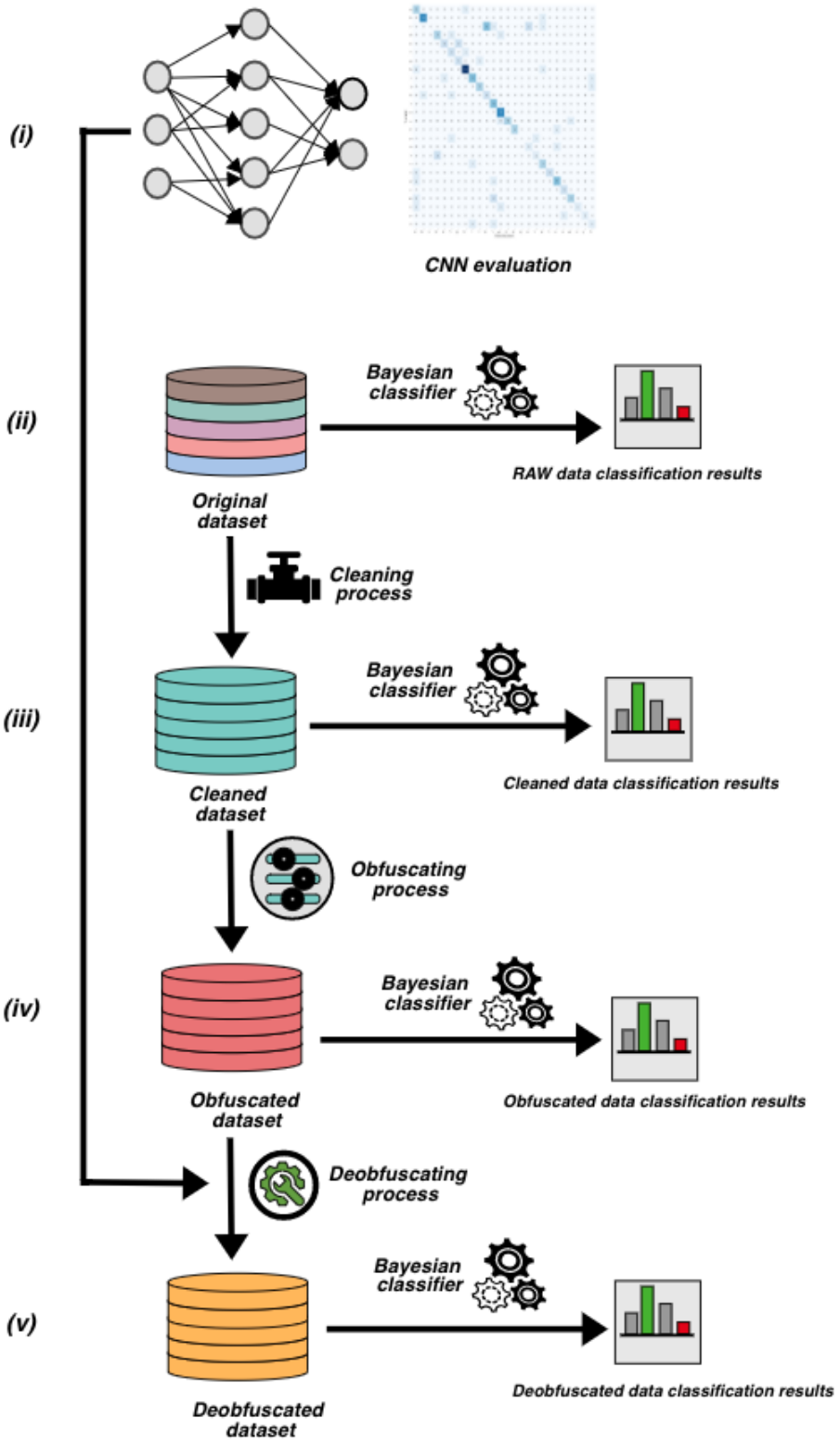


Figure 6.3: Experimental protocol.

The third step consists in the identification and elimination of the non-alphanumeric characters, a text cleaning process from the dataset represented in its original state and in the classification of the resulting texts. For each message, the phone numbers and web URLs embedded in the texts were preserved and the rest of the text was converted to lowercase.

Finally, the last step consists in executing the deobfuscation algorithm, the substitution of the obfuscated characters by the CNN and, when the dataset only contains plain text, the classification of the dataset.

The analysis of the results include a comparison of the performance achieved during the last four steps (baseline - step 2, cleaning - step 3, obfuscating - step 4 and deobfuscating - step 5) using standard measures [75] such as: accuracy, precision, recall and f-score. We have used a 10-fold cross-validation scheme to run experiments in the last four steps.

6.4 Results and discussion

This section describes the results obtained during the experiments. First of all, before starting the work with the datasets, the performance of the CNN has been tested by computing its confusion matrix. The CNN has been tested with a set of 115 obfuscated Leetspeak characters and the results obtained can be seen in Figure 6.4. Although there are some errors, the main diagonal of the confusion matrix shows a large number of hits in recognizing Leetspeak sequences.

Then a practical experimentation has been performed to evaluate the performance of the CNN based solution in a real text classification context. This experimentation is performed in the 5th step of figure 6.3, in which the detected Leetspeak sequences are replaced by the translations provided by the CNN. Figure 6.5 show the accuracy scores obtained for the 4 datasets tested. The initial baseline is based on our previous work [39] in which we identified which were the 10 best classifiers and their configurations for the two YouTube datasets in use. The figure has been divided in four separate parts grouping all configurations done by each dataset.

As shown in Figure 6.5, the best results are obtained with the original dataset configurations (Baseline and Cleaned). However, when having messages containing Leetspeak obfuscations, the use of the proposed deobfuscation scheme improves the classification results for all of the analyzed datasets. The use of the proposed deobfuscation scheme makes it possible, in some cases, to achieve the initial classification results (Baseline and Clean) as when no obfuscation had occurred. Therefore, the use of CNNs allows good deobfuscation results to be obtained without the use of other

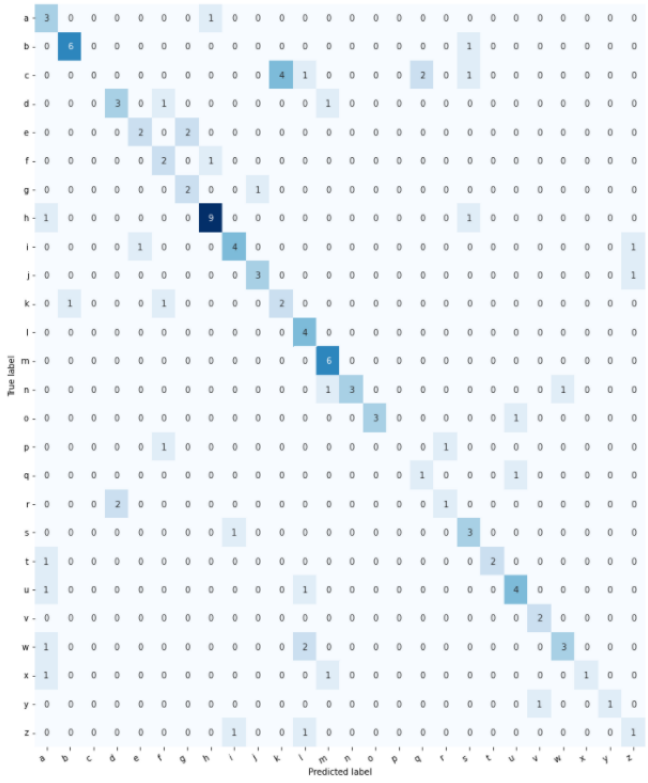
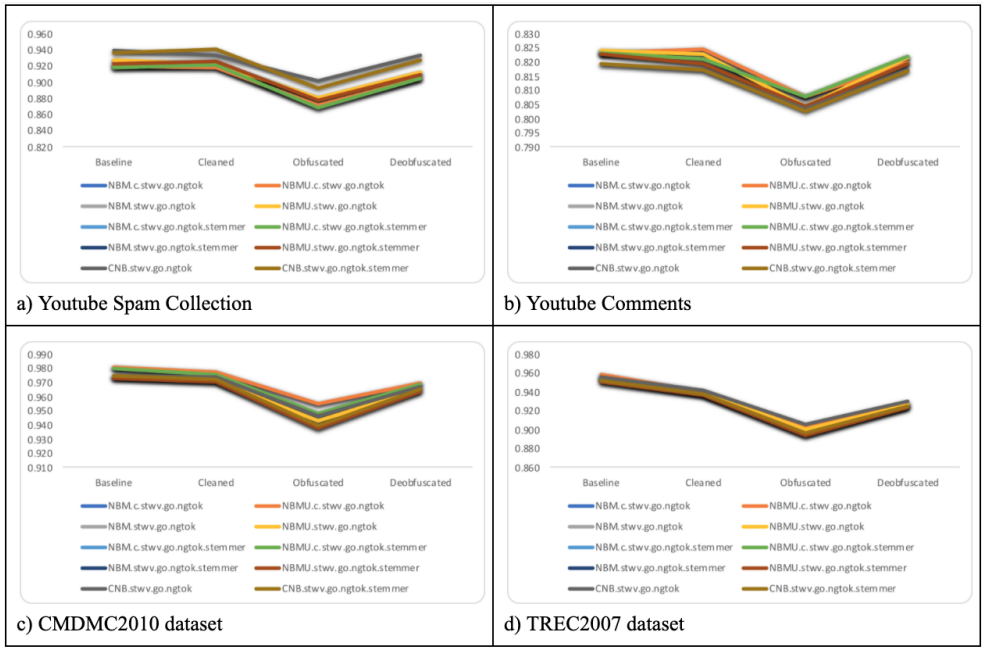


Figure 6.4: CNN confusion matrix.



a) Youtube Spam Collection

b) Youtube Comments

c) CMDMC2010 dataset

d) TREC2007 dataset

Figure 6.5: Experimental protocol achieved accuracy.

Classifier/preprocessing configuration	**Dataset status**	YouTube Comments Dataset		YouTube Spam Collection Dataset	
	Measure	precision	recall	precision	recall
NBM.c.stwv.go.ngtok	Baseline	0.801	0.387	0.884	0.972
	Cleaned	0.798	0.399	0.880	0.974
	Obfuscated	0.802	0.308	0.807	0.984
	Deobfuscated	0.808	0.366	0.857	0.979
NBMU.c.stwv.go.ngtok	Baseline	0.801	0.387	0.884	0.972
	Cleaned	0.798	0.399	0.880	0.974
	Obfuscated	0.802	0.308	0.807	0.984
	Deobfuscated	0.808	0.366	0.857	0.979
NBM.stwv.go.ngtok	Baseline	0.834	0.371	0.894	0.973
	Cleaned	0.820	0.373	0.892	0.966
	Obfuscated	0.809	0.284	0.822	0.982
	Deobfuscated	0.836	0.357	0.870	0.976
NBMU.stwv.go.ngtok	Baseline	0.834	0.371	0.894	0.973
	Cleaned	0.820	0.373	0.892	0.966
	Obfuscated	0.809	0.284	0.822	0.982
	Deobfuscated	0.836	0.357	0.870	0.976
NBM.c.stwv.go.ngtok.stemmer	Baseline	0.822	0.375	0.881	0.973
	Cleaned	0.805	0.376	0.883	0.978
	Obfuscated	0.828	0.293	0.805	0.982
	Deobfuscated	0.826	0.366	0.857	0.978
NBMU.c.stwv.go.ngtok.stemmer	Baseline	0.822	0.375	0.881	0.973
	Cleaned	0.805	0.376	0.883	0.978
	Obfuscated	0.828	0.293	0.805	0.982
	Deobfuscated	0.826	0.366	0.857	0.978
NBM.stwv.go.ngtok.stemmer	Baseline	0.847	0.355	0.890	0.969
	Cleaned	0.820	0.355	0.894	0.971
	Obfuscated	0.847	0.265	0.817	0.981
	Deobfuscated	0.856	0.333	0.863	0.978
NBMU.stwv.go.ngtok.stemmer	Baseline	0.847	0.355	0.890	0.969
	Cleaned	0.820	0.355	0.894	0.971
	Obfuscated	0.847	0.265	0.817	0.981
	Deobfuscated	0.856	0.333	0.863	0.978
CNB.stwv.go.ngtok	Baseline	0.750	0.415	0.917	0.972
	Cleaned	0.742	0.412	0.915	0.959
	Obfuscated	0.734	0.337	0.853	0.976
	Deobfuscated	0.755	0.398	0.907	0.969
CNB.stwv.go.ngtok.stemmer	Baseline	0.779	0.388	0.912	0.969
	Cleaned	0.757	0.396	0.924	0.965
	Obfuscated	0.757	0.311	0.841	0.975
	Deobfuscated	0.768	0.384	0.896	0.971

Table 6.3: Precision and recall values for YouTube Comments Dataset.

complex procedures.

In addition, we also performed an evaluation of the impact of our deobfuscation scheme using precision and recall measures. Table 6.3 shows precision and recall evaluations achieved for datasets containing comments (YouTube Comments Dataset and YouTube Spam Collection Dataset).

As shown in Table 6.3, the results show the same behaviour as for the accuracy evaluation. Table 6.3 shows the precision and recall achieved in email datasets (CSDMC 2010 Spam and TREC 2007 Public Corpora).

The results obtained in Table 6.3 and 6.4 confirm the robustness of the deobfuscation process. Additionally, a f-score evaluation is also performed with the previous four datasets to check if the deobfuscation process was valid according to other quality

Classifier/preprocessing configuration	**Dataset status**	CSDMC 2010		TREC 2007	
	Measure	precision	recall	precision	recall
NBM.c.stwv.go.ngtok	Baseline	0.991	0.948	0.987	0.847
	Cleaned	0.992	0.936	0.991	0.767
	Obfuscated	0.999	0.861	0.998	0.617
	Deobfuscated	0.991	0.913	0.993	0.718
NBMU.c.stwv.go.ngtok	Baseline	0.991	0.948	0.987	0.847
	Cleaned	0.992	0.936	0.991	0.767
	Obfuscated	0.999	0.861	0.998	0.617
	Deobfuscated	0.991	0.913	0.993	0.718
NBM.stwv.go.ngtok	Baseline	0.992	0.927	0.990	0.829
	Cleaned	0.994	0.916	0.991	0.763
	Obfuscated	0.997	0.824	0.997	0.603
	Deobfuscated	0.994	0.898	0.992	0.714
NBMU.stwv.go.ngtok	Baseline	0.992	0.927	0.990	0.829
	Cleaned	0.994	0.916	0.991	0.763
	Obfuscated	0.997	0.824	0.997	0.603
	Deobfuscated	0.994	0.898	0.992	0.714
NBM.c.stwv.go.ngtok.stemmer	Baseline	0.990	0.946	0.989	0.828
	Cleaned	0.993	0.932	0.993	0.757
	Obfuscated	0.999	0.839	0.998	0.584
	Deobfuscated	0.993	0.909	0.996	0.706
NBMU.c.stwv.go.ngtok.stemmer	Baseline	0.990	0.946	0.989	0.828
	Cleaned	0.993	0.932	0.993	0.757
	Obfuscated	0.999	0.839	0.998	0.584
	Deobfuscated	0.993	0.909	0.996	0.706
NBM.stwv.go.ngtok.stemmer	Baseline	0.992	0.924	0.991	0.810
	Cleaned	0.994	0.914	0.991	0.754
	Obfuscated	0.998	0.806	0.997	0.579
	Deobfuscated	0.994	0.891	0.994	0.700
NBMU.stwv.go.ngtok.stemmer	Baseline	0.992	0.924	0.991	0.810
	Cleaned	0.994	0.914	0.991	0.754
	Obfuscated	0.998	0.806	0.997	0.579
	Deobfuscated	0.994	0.891	0.994	0.700
CNB.stwv.go.ngtok	Baseline	0.991	0.930	0.990	0.833
	Cleaned	0.994	0.923	0.991	0.777
	Obfuscated	0.997	0.835	0.997	0.625
	Deobfuscated	0.993	0.903	0.992	0.728
CNB.stwv.go.ngtok.stemmer	Baseline	0.992	0.927	0.992	0.818
	Cleaned	0.995	0.919	0.991	0.758
	Obfuscated	0.998	0.812	0.997	0.587
	Deobfuscated	0.994	0.898	0.993	0.707

Table 6.4: Precision and recall values for email datasets.

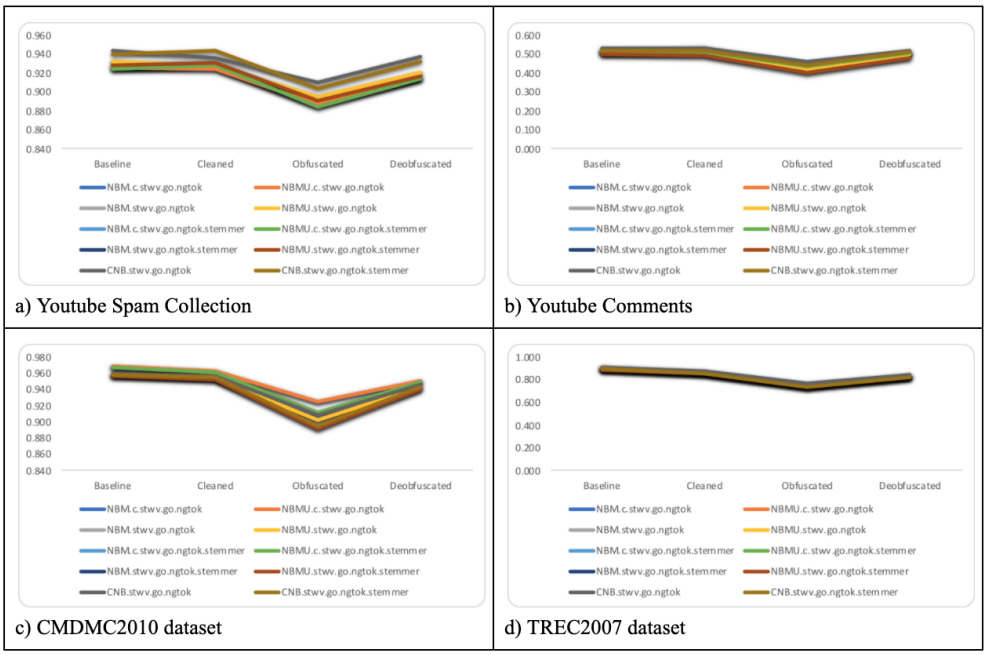


Figure 6.6: Experimental protocol achieved f-score.

criteria. The results are shown in figure 6.6.

6.5 Conclusions and future work

The present study intends to discover methods for automatically deobfuscate Leetspeak character sequences using a CNN based decoding model. This study provides (i) a well-founded CNN model for Leetspeak decoding processes and its corresponding text substitution, (ii) an image database that has been used to train the CNN model in this study, and (iii) four datasets for evaluating Leetspeak decoding processes. By experimental tests, it was found that the CNN design and building processes are efficient and reveal high performance.

When analyzing the difference between clean text and obfuscated text, it can be seen that the use of Leetspek has a huge impact on the performance of the algorithms as a whole. When obfuscating the characters, spammers can hide the whole word from the classifier, making the word unusable in the spam classification process. Once the messages have been deobfuscated with our system, the performance of the classifiers improve, sometimes reaching the initial values previous to the obfuscation process. This shows that the approach works well and can be used to identify obfuscated characters. However, there are some characters, as can be seen in figure 6.4, that are not correctly identified and further improvement of the system is needed. Therefore, future

work includes extending the image database and improving the CNN architecture to obtain better deobfuscation results.

The main limitation of our proposal is the detection of obfuscated characters containing a single punctuation mark, as this requires further analysis. For example, the character H could be obfuscated with a dash ("-") between two "i "s (i.e. "i-i"). This situation could lead to a large number of decoding errors (e.g. the translation of "semi-interlaced" to "semhnterlaced", which is incorrect). To solve this problem, we consider the use of dictionary-based schemes (to look up whether the word exists unchanged) before using a deobfuscation algorithm in future. In addition, we take advantage of the multiple outputs of the CNN (e.g., we consider the five outputs of the CNN that reach the highest value) and check the existence of the resulting word in a dictionary. In addition, the algorithm used to recognise Leetspeak sequences also needs to be improved. The one used in this study can only detect one Leetspeak sequence per word. Therefore, future work involves improvement in several directions (CNN performance, algorithms for detecting Leetspeak sequences and use of a dictionary) that will lead to significant improvements in the deofuscation process.

Chapter 7

Summary

This chapter presents the main outcomes achieved on this thesis. Section 7.1 outlines the work accomplished during the realisation of the research work. Section 7.2 describes the opened work lines for future researches.

7.1 Conclusions

This research work has focused on improving the identification of spam messages. To achieve this, we have used the semantic approach to extract the meaning of the words that compose the messages. The meanings of the words have been represented using synsets present in ontological dictionaries such as BabelNet. Using the relations available in ontological dictionaries, it is possible to navigate through meanings from the more specific (e.g. car) to the more generalistic (e.g. mode_of_transport), going through the intermediates (e.g. motor_vehicle). This navigation and browsing of hypernyms and hyponyms makes it possible to group words, tokens or, in our case, meanings.

The synsets-based representation makes it possible to group the specific meanings (elements such as "1 car", "1 bicycle" and "4 ship") into a more general one ("6 means_of_transport"). This has made it possible to achieve a reduction in the number of features (1 car, 1 bicycle, 4 ship $\rightarrow$ 6 means_of_transport) with a reduction in the dimensionality of the feature vector (cardinal of the synsets). If the synsets are clustered in an ascending order of generality, from a specific element to a more general one (being the clustering determined by a Genetic Algorithm), the dimensionality of the resulting group is reduced but without loss of information. This has enabled us to achieve the first objective, which was a reduction in the number of features. The tests carried out show that the number of features has been reduced to nearly 50% value, while maintaining the FP and FN indicators.

The use of synset-based representation combined with the ontological dictionaries,

has also made it possible to identify the categories upon which the elements that compose the messages fall (such as adjectives, adverbs, nouns and verbs), and to identify which of them provide more or less information or deteriorate the accuracy of the message classification ("noisy" elements). The designed algorithm is capable of identifying and deleting the most irrelevant or noisy terms in the classification process, allowing to further reduce the size of the feature vector, which was the second objective of this work. Nonetheless, in this case the additional dimensionality reduction is achieved at the cost of a small loss of information. In the carried experiments, it has been demonstrated that the genetic algorithm used can possitively identify the less significant synsets and propose their elimination. In this case, higher reduction rates are obtained achieving similar values of FP and FN.

The conversion into synsets of the text that composes the message (for further processing on synset-based representation) proposed in this work, implies an initial loss of information, given that any word that has an obfuscated character (Hell0) and does not exist in an ontological dictionary is discarded. This means that these words are lost and cannot be used for classification. To address this problem, a system has been developed based on a Convolutional Neural Network to decode the obfuscated characters in Leetspeak and replace them with their equivalent correct text character. This allows the word to be analyzed in its context and meaning extracted from the ontological dictionary. The proposed system achieves a high degree of accuracy in Leetspeak character deobfuscation. This system has achieved the third and last of the defined objectives, increasing the number of useful tokens to be processed and represented as synsets.

7.2 Future work

The main disadvantage of the system proposed for feature reduction is the training time it demands. As seen in Table 4.3, the process takes 10 days on an Intel CPU based machine with 72 cores, due to the multitude of evaluations and executions performed in the optimization process by the MOEA. An appropriate future line would be to run the genetic algorithms in a faster language, such as C or C++, in order to increase the speed of the training phase. Previous work [48] has compared the performance of both programming languages when carrying out intensive computations such as 3D modelling. In these scenarios it has been demonstrated that the using the appropriate optimizations, Java is from 1.21 to 1.91 times slower than C++under Linux. It is true that after training the model, the classification process is fast, but by shortening the training phase duration it would be possible to achieve a much shorter global cycle

duration.

As it has been mentioned in the final section of Chapter 6, the main limitation of the proposed deobfuscation scheme is the detection of single punctuation mark obfuscations. The character H could be obfuscated with two "i" and one hyphen "-" such in "i-iello". This situation requires a much deeper treatment to avoid incorrect deobfuscations (e. g. "semi-interlaced" to "semHnterlaced"). This kind of errors, as well as the observed in the confusion matrix, can be detected and corrected by linking the deobfuscation system to a semantic dictionary such as BabelNet. Checking the deobfuscated word in a dictionary would reduce the mistakes in the deobfuscation process. In case the deobfuscated word does not exist, that clearly indicates an error. In that case, the next more probable character would be tried, and after that the third one etc., until the character forms a word that exists in the dictionary.

At this point, it is necessary to remember that the aim of this thesis was to advance on the path to the message intentionality detection. This objective is now closer due to the presented generalization algorithm that allows for the grouping of more than one token in a synset, and the deletion of words that may not provide relevance to the text. These provide the ability to reduce the number of features, making it possible to compose a summary of the messages in a few simpler sentences, which consist of a subject and a predicate, free of noisy terms, due to the low loss characteristic of the proposed low-loss dimensionality reduction approach.

Indeed it seems to be key to focus on the verb and the noun of the predicate first. The direct object completes the meaning of a transitive verb, being a noun (generalized synset), pronoun, or word group that tells who or what receives the action of the verb. This is, getting to know the action or the verb (e.g., to sell, to buy, to invoice) and the direct object, we might be able to very precisely know the intention behind the message (e.g., sell a drug, preview an attached document, invite to a conference, ship boxes). Then, the subject is what (or whom) the sentence is about. In that case it would be possible to detect subjects like "shopping list" or "drugs" with predicates like "have to be purchased". The first subject combined with the predicate forms a legitimate sentence inside a message ("shopping list have to be purchased). This does not occur in the combination of the second subject and the same predicate ("drugs have to be purchased"). This can help us distinguish a legitimate informational message from another message that clearly is spam. At this point it will be necessary to create bags of words of verbs (e.g., sell, stole, inform, have fun) and generalized nouns (e.g., drug, invoice, conference, business) to aid in classification. All this will be future research work in order to get closer to the intentionality detection.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[2] S Abiramasundari, V Ramaswamy, and J Sangeetha. Spam filtering using semantic and rule based model via supervised learning. *Annals of the Romanian Society for Cell Biology*, pages 3975–3992, 2021.

[3] Eneko Agirre and Philip Edmonds. *Word sense disambiguation: Algorithms and applications*, volume 33. Springer Science & Business Media, 2007.

[4] Eneko Agirre, Timothy Baldwin, and David Martinez. Improving parsing and pp attachment performance with sense information. In *proceedings of ACL-08: HLT*, pages 317–325, 2008.

[5] Naeem Ahmed, Rashid Amin, Hamza Aldabbas, Deepika Koundal, Bader Alouffi, and Tariq Shah. Machine learning techniques for spam detection in email and iot platforms: analysis and research challenges. *Security and Communication Networks*, 2022, 2022.

[6] Tiago A Almeida and Akebo Yamakami. Content-based spam filtering. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2010.

[7] Tiago A Almeida, Tiago P Silva, Igor Santos, and José M Gómez Hidalgo. Text normalization and semantic indexing to enhance instant messaging and sms spam filtering. *Knowledge-Based Systems*, 108:25–32, 2016.

[8] Berna Altınel and Murat Can Ganiz. Semantic text classification: A survey of past and recent advances. *Information Processing & Management*, 54(6): 1129–1153, 2018.

[9] Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Georgios Sakkis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *CoRR*, cs.CL/0009009, 2000.

[10] Abdolrahman Attar, Reza Moradi Rad, and Reza Ebrahimi Atani. A survey of image spamming and filtering techniques. *Artificial Intelligence Review*, 40(1): 71–105, 2013.

[11] Eman M Bahgat, Sherine Rady, Walaa Gad, and Ibrahim F Moawad. Efficient email classification approach based on semantic methods. *Ain Shams Engineering Journal*, 9(4):3259–3269, 2018.

[12] Vitor Basto-Fernandes, Iryna Yevseyeva, José R Méndez, Jiaqi Zhao, Florentino Fdez-Riverola, and Michael TM Emmerich. A spam filtering multi-objective optimization study covering parsimony maximization and three-way classification. *Applied Soft Computing*, 48:111–123, 2016.

[13] Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, Roberto Navigli, et al. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conference on Artificial Intelligence, Inc, 2021.

[14] Hanif Bhuiyan, Akm Ashiquzzaman, Tamanna Juthi, Suzit Biswas, and Jinat Ara. A survey of existing e-mail spam filtering methods considering machine learning techniques. *Global Journal of Computer Science and Technology*, 18 (2), 01 2018.

[15] Battista Biggio, Giorgio Fumera, Ignazio Pillai, and Fabio Roli. A survey and experimental evaluation of image spam filtering techniques. *Pattern recognition letters*, 32(10):1436–1446, 2011.

[16] Elie Bursztein, Matthieu Martin, and John Mitchell. Text-based captcha strengths and weaknesses. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 125–138, 2011.

[17] Ylermi Cabrera-León, Patricio García Báez, and Carmen Paz Suárez-Araujo. Non-email spam and machine learning-based anti-spam filters: trends and some

remarks. In *International Conference on Computer Aided Systems Theory*, pages 245–253. Springer, 2017.

[18] Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, 2007.

[19] Manajit Chakraborty, Sukomal Pal, Rahul Pramanik, and C Ravindranath Chowdary. Recent developments in social spam detection and combating techniques: A survey. *Information Processing & Management*, 52(6):1053–1073, 2016.

[20] Ashish Chandra and Mohammad Suaib. A survey on web spam and spam 2.0. *International Journal of Advanced Computer Research*, 4(2):634, 2014.

[21] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[22] Arindam Chaudhuri, Krupa Mandaviya, Pratixa Badelia, and Soumya K Ghosh. Optical character recognition systems. In *Optical Character Recognition Systems for Different Languages with Soft Computing*, pages 9–41. Springer, 2017.

[23] David W Cheung, HY Hwang, Ada W Fu, and Jiawei Han. Efficient rule-based attribute-oriented induction for data mining. *Journal of Intelligent Information Systems*, 15(2):175–200, 2000.

[24] Gordon V Cormack. Email spam filtering: A systematic review. *Found. Trends Inf. Retr.*, 1(4):335–455, 2006.

[25] Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, and Opeyemi Emmanuel Ajibuwa. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802, 2019. ISSN 2405-8440.

[26] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, February 2009.

[27] Teófilo Emídio De Campos, Bodla Rakesh Babu, Manik Varma, et al. Character recognition in natural images. *VISAPP (2)*, 7:2, 2009.

[28] Iñaki Velez de Mendizabal, Vitor Basto-Fernandes, Enaitz Ezpeleta, Jose R Mendez, and Urko Zurutuza. Sdrs: A new lossless dimensionality reduction for text corpora. *Information Processing & Management*, 57(4):102249, 2020.

[29] Iñaki Velez de Mendizabal, Xabier Vidriales, Vitor Basto Fernandes, Enaitz Ezpeleta, José Ramón Méndez, and Urko Zurutuza. Set of obfuscated spam dataset by using leetspeak transformations, March 2022. URL https://doi.org/10.5281/zenodo.6373653.

[30] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[31] Xuelian Deng, Yuqing Li, Jian Weng, and Jilian Zhang. Feature selection for text classification: A review. *Multimedia Tools and Applications*, 78(3): 3797–3816, 2019.

[32] Vikas P Deshpande, Robert F Erbacher, and Chris Harris. An evaluation of naïve bayesian anti-spam filtering techniques. In *2007 IEEE SMC Information Assurance and Security Workshop*, pages 333–340. IEEE, 2007.

[33] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

[34] Feng-Lin Du, Jia-Xing Li, Zhi Yang, Peng Chen, Bing Wang, and Jun Zhang. Captcha recognition based on faster r-cnn. In *International Conference on Intelligent Computing*, pages 597–605. Springer, 2017.

[35] George H Dunteman. *Principal components analysis*. Sage Publications, Inc., 1989.

[36] John Evershed and Kent Fitch. Correcting noisy ocr: Context beats confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, pages 45–51, 2014.

[37] Enaitz Ezpeleta, Iñaki Garitano, Ignacio Arenaza-Nuno, José María Gómez Hidalgo, and Urko Zurutuza. Novel comment spam filtering method on youtube: Sentiment analysis and personality recognition. In *International Conference on Web Engineering*, pages 228–240. Springer, 2017.

[38] Enaitz Ezpeleta, Inaki Garitano, Urko Zurutuza, and José María Gómez Hidalgo. Short messages spam filtering combining personality recognition and sentiment analysis. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 25(Suppl. 2):175–189, 2017.

[39] Enaitz Ezpeleta, Mikel Iturbe, Inaki Garitano, Inaki Velez de Mendizabal, and Urko Zurutuza. A mood analysis on youtube comments and a method for improved social spam detection. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 514–525. Springer, 2018.

[40] Jorge Fdez-Glez, David Ruano-Ordás, Rosalía Laza, José Ramon Méndez, Reyes Pavón, and Florentino Fdez-Riverola. Wsf2: a novel framework for filtering web spam. *Scientific Programming*, 2016, 2016.

[41] Eveline Flamand. Deciphering l33t5p34k internet slang on message boards. *Diss. Ghent University*, 2008.

[42] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[43] Giorgio Fumera, Ignazio Pillai, and Fabio Roli. Spam filtering based on the analysis of text information embedded into images. *Journal of Machine Learning Research*, 7(12), 2006.

[44] Yan Gao, Ming Yang, Xiaonan Zhao, Bryan Pardo, Ying Wu, Thrasyvoulos N Pappas, and Alok Choudhary. Image spam hunter. In *2008 IEEE international conference on acoustics, speech and signal processing*, pages 1765–1768. IEEE, 2008.

[45] Pranjul Garg and Nancy Girdhar. A systematic review on spam filtering techniques based on natural language processing framework. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 30–35. IEEE, 2021.

[46] Robert Gentleman and Vincent J Carey. Unsupervised machine learning. In *Bioconductor case studies*, pages 137–157. Springer, 2008.

[47] Martin Gerlach, Hanyu Shi, and Luís A Nunes Amaral. A universal information theoretic approach to the identification of stopwords. *Nature Machine Intelligence*, 1(12):606–612, 2019.

[48] Luca Gherardi, Davide Brugali, and Daniele Comotti. A java vs. c++ performance evaluation: a 3d modeling benchmark. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pages 161–172. Springer, 2012.

[49] José María Gómez Hidalgo, Guillermo Cajigas Bringas, Enrique Puertas Sánz, and Francisco Carrero García. Content based sms spam filtering. In *Proceedings of the 2006 ACM symposium on Document engineering*, pages 107–114, 2006.

[50] Gregory Grefenstette. Tokenization. In *Syntactic Wordclass Tagging*, pages 117–133. Springer, 1999.

[51] Antonio Gulli and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[52] Thiago S Guzella and Walmir M Caminhas. A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, 36(7):10206–10222, 2009.

[53] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[54] Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge discovery in databases: An attribute-oriented approach. In *VLDB*, volume 18, pages 574–559, 1992.

[55] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[56] Wen-Jing Hong, Peng Yang, and Ke Tang. Evolutionary computation for large-scale multi-objective optimization: A decade of progresses. *International Journal of Automation and Computing*, 18(2):155–169, 2021.

[57] Kurt Hornik and Bettina Grün. topicmodels: An r package for fitting topic models. *Journal of statistical software*, 40(13):1–30, 2011.

[58] Nancy Ide and Jean Véronis. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, 24(1):1–40, 1998.

[59] M Ikonomakis, Sotiris Kotsiantis, and V Tampakas. Text classification using machine learning techniques. *WSEAS transactions on computers*, 4(8):966–974, 2005.

[60] Niddal H Imam, Vassilios G Vassilakis, and Dimitris Kolovos. Ocr post-correction for detecting adversarial text images. *Journal of Information Security and Applications*, 66:103170, 2022.

[61] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys (CSUR)*, 50(2):1–38, 2017.

[62] W Stalin Jacob et al. Multi-objective genetic algorithm and cnn-based deep learning architectural scheme for effective spam detection. *International Journal of Intelligent Networks*, 3:9–15, 2022.

[63] Ruholla Jafari-Marandi. Supervised or unsupervised learning? investigating the role of pattern recognition assumptions in the success of binary predictive prescriptions. *Neurocomputing*, 434:165–193, 2021.

[64] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[65] Karen Sparck Jones, Peter Willett, et al. *Readings in information retrieval*. Morgan Kaufmann, 1997.

[66] Akash Junnarkar, Siddhant Adhikari, Jainam Fagania, Priya Chimurkar, and Deepak Karia. E-mail spam classification via machine learning and natural language processing. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pages 693–699. IEEE, 2021.

[67] Asif Karim, Sami Azam, Bharanidharan Shanmugam, Krishnan Kannoorpatti, and Mamoun Alazab. A comprehensive survey for intelligent spam email detection. *IEEE Access*, 7:168261–168295, 2019.

[68] Zenun Kastrati, Ali Shariq Imran, and Sule Yildirim Yayilgan. The impact of deep learning on document classification using semantically rich representations. *Information Processing & Management*, 56(5):1618–1632, 2019.

[69] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[70] Sotiris B Kotsiantis, I Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[71] Thomas Navin Lal, Olivier Chapelle, Jason Western, and André Elisseeff. Embedded methods. In *Studies in Fuzziness and Soft Computing*, volume 207, pages 137–165. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[72] Egoitz Laparra, German Rigau, and Montse Cuadros. Exploring the integration of wordnet and framenet. In *Proceedings of the 5th Global WordNet Conference (GWC 2010), Mumbai, India*, 2010.

[73] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[74] RAZA Mansoor, Nathali Dilshani Jayasinghe, and Muhana Magboul Ali Muslam. A comprehensive review on email spam classification using machine learning algorithms. In *2021 International Conference on Information Networking (ICOIN)*, pages 327–332. IEEE, 2021.

[75] Javier Martínez Torres, Carla Iglesias Comesaña, and Paulino J García-Nieto. Machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics*, 10(10):2823–2836, 2019.

[76] José R Méndez, Tomás R Cotos-Yañez, and David Ruano-Ordás. A new semantic-based feature selection method for spam filtering. *Applied Soft Computing*, 76:89–104, 2019.

[77] José Ramon Méndez, Daniel Glez-Peña, Florentino Fdez-Riverola, Fernando Díaz, and Juan M Corchado. Managing irrelevant knowledge in cbr models for unsolicited e-mail classification. *Expert Systems with Applications*, 36(2): 1601–1614, 2009.

[78] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

[79] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69, 2009.

[80] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial intelligence*, 193:217–250, 2012.

[81] Roberto Navigli and Paola Velardi. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 27(7):1075–1086, 2005.

[82] Roberto Navigli, Mirella Lapata, et al. Graph connectivity measures for unsupervised word sense disambiguation. In *IJCAI*, volume 7, pages 1683–1688, 2007.

[83] Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. Ten years of babelnet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4559–4567, 2021.

[84] Maria Novo-Loures, Reyes Pavon, Rosalia Laza, David Ruano-Ordas, and Jose R Mendez. Using natural language preprocessing architecture (nlpa) for big data text sources. *Scientific Programming*, 2020.

[85] Shreya Patankar, Madhura Phadke, and Satish Devane. Wiki sense bag creation using multilingual word sense disambiguation. *IAES International Journal of Artificial Intelligence*, 11(1):319, 2022.

[86] Noemi Perez-Diaz, David Ruano-Ordas, Florentino Fdez-Riverola, and Jose R Mendez. Sdai: An integral evaluation methodology for content-based spam filtering models. *Expert Systems with Applications*, 39(16):12487–12500, 2012.

[87] Noemí Pérez-Díaz, David Ruano-Ordás, Florentino Fdez-Riverola, and José Ramon Méndez. Boosting accuracy of classical machine learning antispam classifiers in real scenarios by applying rough set theory. *Scientific Programming*, 2016.

[88] David Ruano-Ordás, Vitor Basto-Fernandes, Iryna Yevseyeva, and José Ramón Méndez. Evolutionary multi-objective scheduling for anti-spam filtering throughput optimization. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 137–148. Springer, 2017.

[89] Nadjate Saidani, Kamel Adi, and Mohand Said Allili. A semantic-based classification approach for an enhanced spam detection. *Computers & Security*, 94:101716, 2020.

[90] Naveen Saini and Sriparna Saha. Multi-objective optimization techniques: A survey of the state-of-the-art and applications. *The European Physical Journal Special Topics*, 230(10):2319–2335, 2021.

[91] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[92] Renato M Silva, Tulio C Alberto, Tiago A Almeida, and Akebo Yamakami. Towards filtering undesired short text messages using an online learning approach with semantic indexing. *Expert Systems with Applications*, 83:314–325, 2017.

[93] Abhinav Sinhmar, Vinamra Malhotra, RK Yadav, and Manoj Kumar. Spam detection using genetic algorithm optimized lstm model. In *Computer Networks and Inventive Communication Technologies*, pages 59–72. Springer, 2022.

[94] Dag Spicer. Raymond tomlinson: Email pioneer, part 1. *IEEE Annals of the History of Computing*, 38(2):72–79, 2016.

[95] Dag Spicer. Raymond tomlinson: Email pioneer, part 2. *IEEE Annals of the History of Computing*, 38(3):78–83, 2016.

[96] Shubhangi Suryawanshi, Anurag Goswami, and Pramod Patil. Email spam detection: An empirical comparative study of different ml and ensemble classifiers. In *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, pages 69–74. IEEE, 2019.

[97] Sayali Sunil Tandel, Abhishek Jamadar, and Siddharth Dudugu. A survey on text mining techniques. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pages 1022–1026. IEEE, 2019.

[98] Jie Tang, Hang Li, Yunbo Cao, and Zhaohui Tang. Email data cleaning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 489–498, 2005.

[99] Shrawan Kumar Trivedi. A study of machine learning classifiers for spam detection. In *2016 4th international symposium on computational and business intelligence (ISCBI)*, pages 176–180. IEEE, 2016.

[100] Chi-Yao Tseng, Pin-Chieh Sung, and Ming-Syan Chen. Cosdes: A collaborative spam detection system with a novel e-mail abstraction scheme. *IEEE transactions on knowledge and data engineering*, 23(5):669–682, 2010.

[101] Andrea Tundis, Gaurav Mukherjee, and Max Mühlhäuser. Mixed-code text analysis for the detection of online hidden propaganda. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pages 1–7, 2020.

[102] Andrea Tundis, Gaurav Mukherjee, and Max Mühlhäuser. An algorithm for the detection of hidden propaganda in mixed-code text over the internet. *Applied Sciences*, 11(5):2196, 2021.

[103] Nemika Tyagi, Sudeshna Chakraborty, Aditya Kumar, Nzanzu Katasohire Romeo, et al. Word sense disambiguation models emerging trends: A comparative analysis. In *Journal of Physics: Conference Series*, volume 2161, page 012035. IOP Publishing, 2022.

[104] Afroza Sharmin Urmi, Md Ahmed, Maqsudur Rahman, AZM Islam, et al. A proposal of systematic sms spam detection model using supervised machine learning classifiers. In *Computer Vision and Robotics*, pages 459–471. Springer, 2022.

[105] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[106] Ismael Vázquez, María Novo-Lourés, Reyes Pavón, Rosalía Laza, José Ramón Méndez, and David Ruano-Ordás. Improvements for research data repositories: The case of text spam. *Journal of Information Science*, page 0165551521998636, 2021.

[107] S Venkatraman, B Surendiran, and P Arun Raj Kumar. Spam e-mail classification for the internet of things environment using semantic similarity approach. *The Journal of Supercomputing*, 76(2):756–776, 2020.

[108] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.

[109] Tarjani Vyas, Payal Prajapati, and Somil Gadhwal. A survey and evaluation of supervised machine learning techniques for spam e-mail filtering. In *2015 IEEE international conference on electrical, computer and communication technologies (ICECCT)*, pages 1–7. IEEE, 2015.

[110] Jing Wang, Jiaohua Qin, Xuyu Xiang, Yun Tan, and Nan Pan. Captcha recognition based on deep convolutional neural network. *Math. Biosci. Eng*, 16(5): 5851–5861, 2019.

[111] John S Whissell and Charles LA Clarke. Clustering for semi-supervised spam filtering. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, pages 125–134, 2011.

[112] Ian H Witten, Eibe Frank, Mark A Hall, Christopher J Pal, and MINING DATA. Practical machine learning tools and techniques. In *DATA MINING*, volume 2, page 4, 2005.

[113] Hailu Xu, Weiqing Sun, and Ahmad Javaid. Efficient spam detection across online social networks. In *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, pages 1–6. IEEE, 2016.

[114] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[115] Iryna Yevseyeva, Vitor Basto-Fernandes, David Ruano-Ordás, and José R Méndez. Optimising anti-spam filters with evolutionary algorithms. *Expert systems with applications*, 40(10):4010–4021, 2013.

[116] Jonathan A Zdziarski. *Ending spam: Bayesian content filtering and the art of statistical language classification*. No starch press, 2005.

[117] Shichao Zhang, Chengqi Zhang, and Qiang Yang. Data preparation for data mining. *Applied artificial intelligence*, 17(5-6):375–381, 2003.

[118] Zhi Zhong and Hwee Tou Ng. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–282, 2012.