This is a peer reviewed version of an article published in Applied Soft Computing (2019), (76), p. 89-104. The published version is available online at: https://doi.org/10.1016/j.asoc.2018.12.008

A New Semantic-Based Feature Selection Method for Spam Filtering

José R. Méndez[1,2] , Tomás R. Cotos-Yañez[2,3], David Ruano-Ordás[1,2*]

[1]Department of Computer Science, University of Vigo, ESEI, Campus As Lagoas, 32004 Ourense, Spain

[2]Centro de Investigaciones Biomédicas (Centro Singular de Investigación de Galicia), Campus Universitario Lagoas-Marcosende, 36310 Vigo, Spain

[3]Department of Statistics and Operations Research, ESEI, Campus As Lagoas, 32004 Ourense, Spain

Email addresses:

JRM: moncho.mendez@uvigo.es

TRCY: cotos@uvigo.es

DRO: drordas@uvigo.es

Corresponding author:

David Ruano Ordás [Tlf.: +34 988 387015 – Fax: +34 988 387001]

ESEI: Escuela Superior de Ingeniería Informática. Edificio Politécnico. Campus

Universitario As Lagoas s/n, 32004 – Ourense – Spain.

Abstract

The Internet emerged as a powerful infrastructure for the worldwide communication and interaction of people. Some unethical uses of this technology (for instance spam or viruses) generated challenges in the development of mechanisms to guarantee an affordable and secure experience concerning its usage. This study deals with the massive delivery of unwanted content or advertising campaigns without the accordance of target users (also known as spam). Currently, words (tokens) are selected by using feature selection schemes; they are then used to create feature vectors for training different Machine Learning (ML) approaches. This study introduces a new feature selection method able to take advantage of a semantic ontology to group words into topics and use them to build feature vectors.

To this end, we have compared the performance of nine well-known ML approaches in conjunction with (i) Information Gain, the most popular feature selection method in the spam-filtering domain and (ii) Latent Dirichlet Allocation, a generative statistical model that allows sets of observations to be explained by unobserved groups that describe why some parts of the data are similar, and (iii) our semantic-based feature selection proposal. Results have shown the suitability and additional benefits of topic-driven methods to develop and deploy high-performance spam filters.

Keywords

Feature selection methods, text mining, spam filtering, e-mail, classification, machine learning.

1. Introduction and motivation.

Despite having emerged in a military context (ARPANET[1]), the outstanding growth and evolution of the Internet converted it into a reliable means of worldwide communication and the exchange of information between people. Nowadays, the availability of new generation smartphones, together with the emergence of 3G/4G network technologies, guarantee a permanent (24 hours a day, 7 days a week, 365 days a year) broadband Internet connection for everybody in any developed country. In this scenario, Internet users can select the most suitable means or provider to effectively communicate and exchange information within a wide variety of e-services including (*i*) Social Networks, (*ii*) Instant Messaging, (*iii*) forums, (*iv*) e-mail, (*v*) weblogs, (*vi*) Peer to Peer (P2P) Networks or (*vii*) web sites. However, the advantages of these types of services can be used for unethical purposes such as the delivery of disturbing content or advertising campaigns unsolicited by target users. Currently, the use of Internet services for these purposes (known as spamming) is very common and hampers the achievement of an efficient and affordable experience. Some examples of this abuse are Instaspam [1] (an example of Social Media Spam [2]), SPIM (Spam Instant Messaging) [3,4], WebSpam [5,6], P2P spam [7], e-mail spam [8] and/or Forum spam [9]. Although all services can be successfully used to distribute spam, e-mail spamming became very popular due to its extended use for multiple purposes (including a notification method for other services such as Social Networks).

As statistics and reports show [10], the percentage of spam e-mails exceeded 50% of global e-mail deliveries in the first 6 months of 2016. This study also reveals how a popular social network was used to distribute Trojan viruses, a high number of advanced persistent threat phishing attacks, and other risks for Internet users. This scenario could dramatically diminish the popularity of Internet services and threaten the useful advantage of this service.

While these abuses continue taking place, ISPs (Internet Service Providers), authorities, enterprises, developers, and worldwide researchers have made invaluable efforts to fight against spam. As a result, various filtering software tools have been successfully introduced, such as SpamAssassin [11] or Wirebrush4SPAM [12]. These products are able to bring together a broad range of smart filtering techniques to accurately filter spam e-mails.

Spam filtering techniques are often classified into different groups [8,13] including: (*i*) content-based filtering, (*ii*) collaborative schemes, (*iii*) domain authorization methods and (*iv*) characteristics-based filters. The former embraces a set of methods able to perform a detailed analysis of message contents (text, image/s, attached documents) to determine a class for the message [14–16]. Collaborative schemes enable sharing detailed information about received spam messages (such as Nilsimsa hashes [17]) in an Internet community. Domain authorization methods are standard mechanisms to define trust servers (identified by their IP addresses) to send messages for a certain domain [18,19]. Finally, characteristics-based filters focus on the use of complementary features from the e-mail such as the number of recipients receiving the same e-mail [20], origin server blacklisting [21] or checking whenever IETF

[1] See details in http://www.darpa.mil/about-us/timeline/arpanet

(Internet Engineering Task Force) standards have been breached (such as MIMEEvalPlugin in latest versions of SpamAssassin).

Although all available groups of methods mentioned below are able to obtain meaningful information to allow the identification of spam messages, content-based filtering schemes are especially important because the decision-making is based exclusively on the content of the message and the interests of the target recipient user. Additionally, content-based methods are mainly based on the use of text patterns and ML supervised methods [14–16,22]. Despite the existence of automatic regular expression finding methods [23], text patterns should currently be manually discovered and included in a spam filter [11,12,22]. Moreover, the use of ML methods for content-based filtering involves additional benefits such as (*i*) the ease of compiling the knowledge required for filtering (can be easily and automatically extracted from a collection of messages belonging to recipient user) and (*ii*) the configuration required (which does not include complex parameters such as IP addresses, ports, and other technological advances). Although this group of methods should be greatly improved to safely operate in a real scenario, the previously mentioned facts encourage their development, use and research.

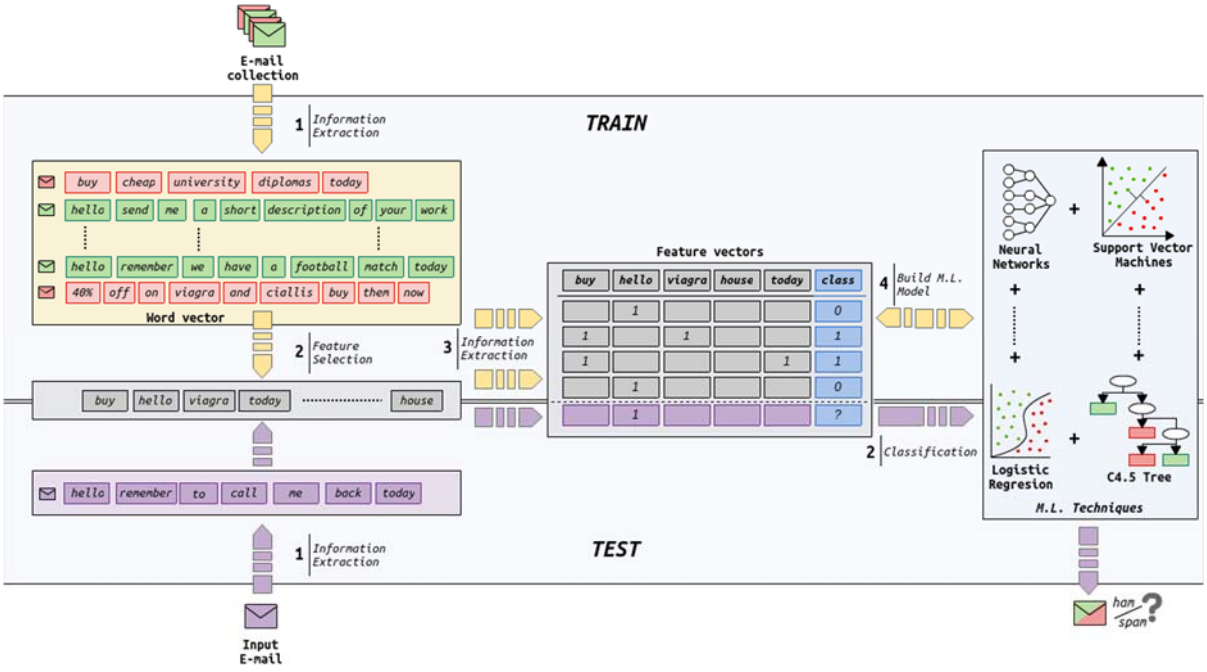

Figure 1. *Flow chart of M.L approaches*

As shown in Figure 1, the operation of ML approaches is commonly structured into four steps: (*i*) extract information (usually done by applying tokenizing schemes), (*ii*) discard confusing, noisy, inconsistent, irrelevant or redundant data (using a feature selection scheme), (*iii*) represent each message as a vector of features according the results of the previous steps; and (*iv*) use a Machine Learning (ML) approach to automatically classify messages. Although ML methods could be improved, feature selection methods should also evolve with them to ensure an increase in the overall performance. In this sense, the use of semantic information to improve feature selection methods seems to be a reliable way to enhance the

overall performance of filters [24–28]. Although several semantic-based classifying methods have been introduced before [29,30], we are convinced that opportunities for improvement are still available to enhance feature selection methods using semantic information.

This study introduces a feature selection technique for a spam-filtering domain that takes advantage of semantic information to guide the selection of features. The main idea behind this proposal is to group word-based features into semantic topics that can be successfully used to generate feature vectors. This method targets spam as a concept generally understood to refer to unsolicited and undesirable emails received by a user. Thus, representing messages according to topics covered by them seems a reliable way to represent the problem and assist ML classifiers in performing better.

The remainder of this paper is structured as follows: Section 2 shows the state of art in feature selection methods. Section 3 includes a thorough description of our proposal while Section 4 analyses its performance in a real scenario. Moreover, Section 5 provides detailed commentary on the results achieved by our methodology and compares its performance with the most used feature selection technique in the spam-filtering domain. Finally, Section 6 shows the main conclusions extracted from the present work and outlines future research lines.

2. State of the art in feature selection methods.

The latest advances in communication technologies (i.e. Internet or 3G/4G networks), together with the higher storage capabilities of newer systems, facilitate the collection and generation of multidimensional data with hundreds of variables. However, jointly managing all this information is not feasible due to: (*i*) the immense computational cost and resources needed; and (*ii*) the reduction of M.L. performance and accuracy motivated by both the high amount of input variables and the inclusion of irrelevant, redundant, and inconsistent information. In order to overcome these issues, feature selection methods emerged as a suitable alternative to find the subset of input variables that better describes the underlying structure of the compiled information [31]. As described in [32], feature selection is one of the most frequent and important techniques in data pre-processing and has become an essential component of the ML process due to its ability to detect relevant features and remove irrelevant, redundant, or noisy data.

As suggested in [33–37] feature selection (FS) methods can be divided into three main categories: (*i*) filter methods (also known as feature raking algorithms) used to compute the relevance of each variable according to an evaluation function that relies solely on the properties of the data (usually the correlation between the feature and the target variable; (*ii*) wrapped methods (also known as feature subset selection approaches), which use the performance of a classification algorithm as a quality criterion; and finally (*iii*) embedded methods, which inject the selection process into the learning of the classifier. Additionally, some forms of discovering and identifying topics (based on analysing the presence of words together with their frequencies) have been explored for the representation of documents in classification problems [38–42]. However, recent studies in the area of text classification and

spam filtering have discovered the utility of taking advantage of semantic information to reduce the dimensionality of input data and avoid discarding relevant information from training/testing datasets [27,28].

2.1 Filter methods.

Filter methods are pre-processor methods that operate in two stages: (*i*) the significance of each feature is assessed; and (*ii*) the highly ranked features are selected (according to the defined threshold or by establishing the maximum number of features) and applied to the desired ML algorithm. The need for defining a cut-off value (also called threshold) or establishing the number of variables to select, allows generating a large number of features (even all features) [35]. In filter methods, the significance of the features is computed using several statistical measures (e.g. probability distributions, statistical correlations, information theory, etc...) [43]. The use of these measures to compute the score of each feature guarantees a fast execution speed while maintaining a moderate use of computational resources. Accordingly, filter methods are a suitable mechanism for extracting features from large datasets with a high number of features. Despite the manner by which they compute the relevance of each feature and their independence regarding the ML algorithm, filter methods achieve fast and reliable generalization approaches. However, discarding features based solely on their significance value can lead to a reduction in their classification performance. Examples of filter methods [31] are χ^2-test, principal/independent component analysis, mutual information techniques, correlation criteria, and Fisher's discriminant scores.

As suggested in [44] filter methods can be classified into four different categories: (*i*) correlation measurement approaches, which assume that a good feature subset contains high feature-class relevance and low inter-feature relevance (i.e. Pearson Coefficient); (*ii*) distance measurement schemes, which are able to compute a feature subset achieving low distance values between same-class samples and high distance values between different-class samples (e.g., Euclidean distance, Mahalanobis distance, Minkowski distance); (*iii*) information measurement strategies, which use the entropy concept [45] to select the best feature subset (e.g. mutual information or information gain); and finally (*iv*) consistency measurement procedures, which follow the assumption that good feature subsets should comprise the lowest possible number of features while simultaneously maintaining high consistency (i.e. avoid samples belonging to different classes and containing the same values of a certain feature set). One example is Rough Sets [46,47].

Finally, in the context of filter methods, recent studies have introduced the combination of different filter methods to increase the performance of ML classifiers [48], newer selection schemes using the same evaluation measures [49,50] or complementing the evaluation measures with additional information to avoid redundant features [51].

2.2 Wrapped methods

When using wrapped methods, feature selection is integrated into the training process [52]. Hence, wrappers use the performance of a (possibly nonlinear) classification algorithm

(generally used as a black box) as an objective function to assess the amount of relevant information conveyed by a subset of features. Using this evaluation mechanism, different candidate subsets of features are scored according to their classification performance, and the best among them (the one achieving the lowest classification error) is then selected. The use of wrapper methods requires: (*i*) the early selection of a classification algorithm; (*ii*) the establishment of a relevance criterion to assess the prediction capacity of a given subset of features; and finally (*iii*) the design of a search procedure able to explore the space of all the possible subsets of features. As introduced in [34,52,53] search methods can be divided into two categories depending on the heuristic type: (*i*) randomized search schemes such as genetic algorithms or simulated annealing [53]; and (*ii*) deterministic search approaches (also called greedy strategies), which carry out a local search in the feature space (e.g. forward and backward selection methods) [34,52].

Taking into account the behaviour of wrapped methods into account, is easy to realize their ability to (*i*) outperform filter strategies in terms of classification error and (*ii*) to consider feature dependencies. However, the use of these methods may cause overfitting, especially when working with reduced datasets. Moreover, they are computationally intensive, especially when building a classifier implies a high computational cost.

Finally it is important to highlight that a combination of filters and wrappers, (called frappers [54], is also possible. Frappers take advantage of filters to build an initial feature ranking. During the second stage, frappers will add new features in their ranking and reject those that do not improve results of a given ML classifier.

2.3 Embedded methods

Embedded methods emerged to combine the qualities of filter and wrapped methods (but not the methods themselves). To this end, embedded methods act as a trade-off between the two models by embedding the feature selection into the model generation stage. This strategy allows improving the results achieved by filter methods by including the learning model while reducing the computational cost of wrappers methods, since the act of performing multiple executions of the learning model to evaluate the features is avoided.

While wrapped models evaluate the performance of a certain ML technique with a different candidate subset of features, embedded models select features during the process of model construction to perform feature selection without further evaluation of the features. The main disadvantage of these methods lies in their dependency of the learning model due to its use within the feature selection process. Decision Trees, Weighted Naïve Bayes and Feature selection using the weight vector of Support Vector Machines (SVM) [43,46] are examples of the most common embedded methods used in feature extraction.

2.4 Topic-based models

Topic-based models [42] emerged within the context of Natural Language Processing to provide a probabilistic modelling of term occurrences in documents and their application to identify groups of documents matching a concrete topic without using semantic information.

Hence, topic models can be easily used to perform the unsupervised classification of documents, feature reduction schemes (represent documents as topics instead of terms) or estimate the similarity between two documents.
Hofmann [41] developed an early simple topic model called Latent Semantic Analysis (LSA). It is based on learning topics through the decomposition of a term-document matrix. Blei *et al*. [39] subsequently introduced the Latent Dirichlet Allocation, which implements a probabilistic model (Dirichlet-multinomial) to represent the relation between topics and words. Despite their limitations (neither takes advantage of semantic information), both models have been successfully used in the document classification domain [38,40].

2.5 Feature selection methods in text classification and spam filtering

In recent years, multiple feature selection methods have been applied to obtain an appropriate subset of features from datasets such as algorithms, based on the ability of the population to (*i*) select important features and (*ii*) remove irrelevant (and/or redundant) features. Moreover, the intrinsic characteristics of spam (such as the word spelling tricks used by spammers to avoid spam filters) forces the development of feature selection algorithms oriented to the spam-filtering and text classification domains.

Few research studies have exploited the use of embedded methods to work with e-mail messages. RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [55] is a rule-based induction algorithm able to incrementally generate classification rules directly from the training dataset. It has been applied to the non-binary classification of messages, and each learned rule covers the attributes of each class (the set of messages having a specific value for a class attribute). The RIPPER algorithm achieves a fast and effective feature selection when dealing with large and noisy datasets due to (*i*) the incremental rule-based learning procedure and (*ii*) the ability to continually prune generated rules achieving high error rates. In [56] the authors applied a SVM algorithm over a lowercase binary feature vector achieving error rate values up to 0.2132%. Additionally, authors in [57] proposed a Bayesian Network to extract highly relevant features from a set of the most representative words in the e-mail domain. Following this approximation in [58] an Artificial Neural Network (ANN) over a bag-of words representation is applied in order to automate spam filtering systems.

With regard to wrapped methods, the work presented in [59] analysed different combinations between ANNs and GA, and demonstrates the suitability of using GA to optimize ANN weights and input features. In [60] the authors use a feature selection method based on genetic algorithm together with an SVM based on Structural Risk Minimisation (SRM) for classifying e-mails.

Filter methods were very popular in the target domain where Information Gain and X^2 have provided great classification results [61–63]. With the aim of improving classification accuracy, the authors in [64] used the Rough Sets theory as a feature selection method in order to reduce the number of features used as input for the SVM classifier. Finally, in [65] the authors combined Random Forest algorithm and partial Decision Trees for spam

classification in combination with different filter selection methods (Entropy, Information Gain, Correlation based feature selection, Chi-square, Gain Ratio, Mutual Information, Symmetrical Uncertainty, One R and Relief). The obtained results demonstrate the high performance of the above-mentioned filter due to their ability to decrease computational complexity while increasing precision. One of the major drawbacks of these methods lies in the redundancy of selected features. To deal with this, some recent studies have studied the benefits of discovering the interaction of words to eliminate (or reduce) such redundancy [51]. Moreover, some newer studies have exploded the usage of semantic information with similar purposes. In detail, Almeida et al. [27] have redefined the pre-processing stage for the problem of spam filtering on SMS by including several additional semantic-based tasks such as SMS lingo/slang translating, word sense disambiguation using a semantic ontology. Moreover, Bahgat and Moawad [28] have exploited the use of synonymy relation from a semantic ontology (synsets on Wordnet) and a filter feature selection scheme to improve the performance of classifiers.

Despite the significant number of studies on selecting features to filter spam, the use of semantic information has not been deeply studied as a method to fully guide the selection of features. Filtering spam involves the ability to separate messages containing irrelevant topics for the end-user. We strongly believe that the identification of topics (and consequently, spam filtering) could not be addressed without using semantic information. Conversely, former proposals are based on using information about words without considering their meaning, and only a few (recent) proposals take semantic information into account [27,28].

In this study, we introduce an approach for semantic-guided feature selection based on the use of a semantic ontology to complete the information included in a message and identify topics from words. Section 3 provides a complete description of our proposal.

3. Feature selection using topic extraction methodology

As noted below, previous works on feature selection for spam filtering have used those techniques to select appropriate words to represent e-mail messages. Taking into account that this form of representing messages (without semantic knowledge) is not especially adequate, we designed a new methodology able to detect and select the features that best summarize the topic of each e-mail. Figure 2 shows the workflow of our methodology divided into four main stages: (*i*) loading the corpus; (*ii*) e-mail parsing process; (*iii*) e-mail topic extractor and guesser; and finally (*iv*) compute the topic-related significance of each feature.

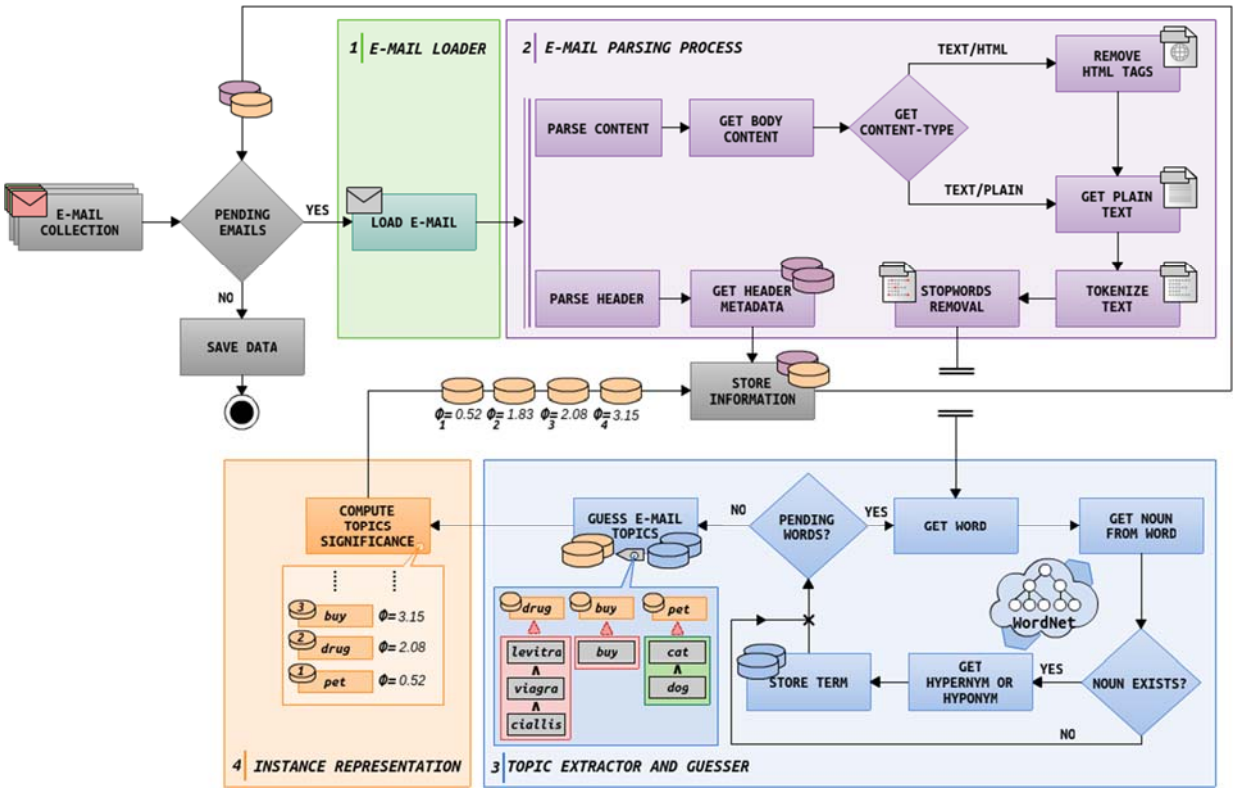

Figure 2. *Workflow of our semantic based-feature selection method.*

As seen in Figure 2, the first stage is in charge of loading messages into memory. Once loaded, the header and body parts of each e-mail are extracted from the original message. Additionally, several pre-processing tasks are carried out to transform the obtained text into valuable information. Specifically, the *header module* extracts useful metadata from the header (such as e-mail date of sender address) while the *body module* extracts raw text from the body of the message (by removing HTML tags, if available). Afterwards, raw text is split into words using spaces ('\s+'), tabs ('\t+'), and newline ('\n') characters as word separators. Additionally, in order to reduce the computational overhead and discard useless information, we decided to (*i*) remove the possessive forms of tokens (termination 's), (*ii*) discard meaningless tokens (using stopwords removal module) and finally, (*iii*) delete tokens included in a list of English proper names[2] as well as those containing Uniform Resource Locators (URLs) or digits. During the third stage, all remaining words are used to guess the topic or topics that best match each target message. To accomplish this task, we used the WordNet Lexical Database[3] [66]. The hierarchical WordNet database groups words into *synsets* (words from the same lexical category that are roughly synonymous), provides short definitions and usage examples of each word, and finally, defines different kinds of semantic relations between synsets (nouns or verbs). Focusing on noun semantic relations, we can distinguish four basic types: (*i*) hyponym (X is a kind of Y), (*ii*) hypernym (X includes the notion of Y among others), (*iii*) meronym (X is a part of Y), (*iv*) holonym (X contains Y among others).

[2] Available at: http://www2.census.gov/topics/genealogy/1990surnames/dist.all.last
[3] Available at https://wordnet.princeton.edu

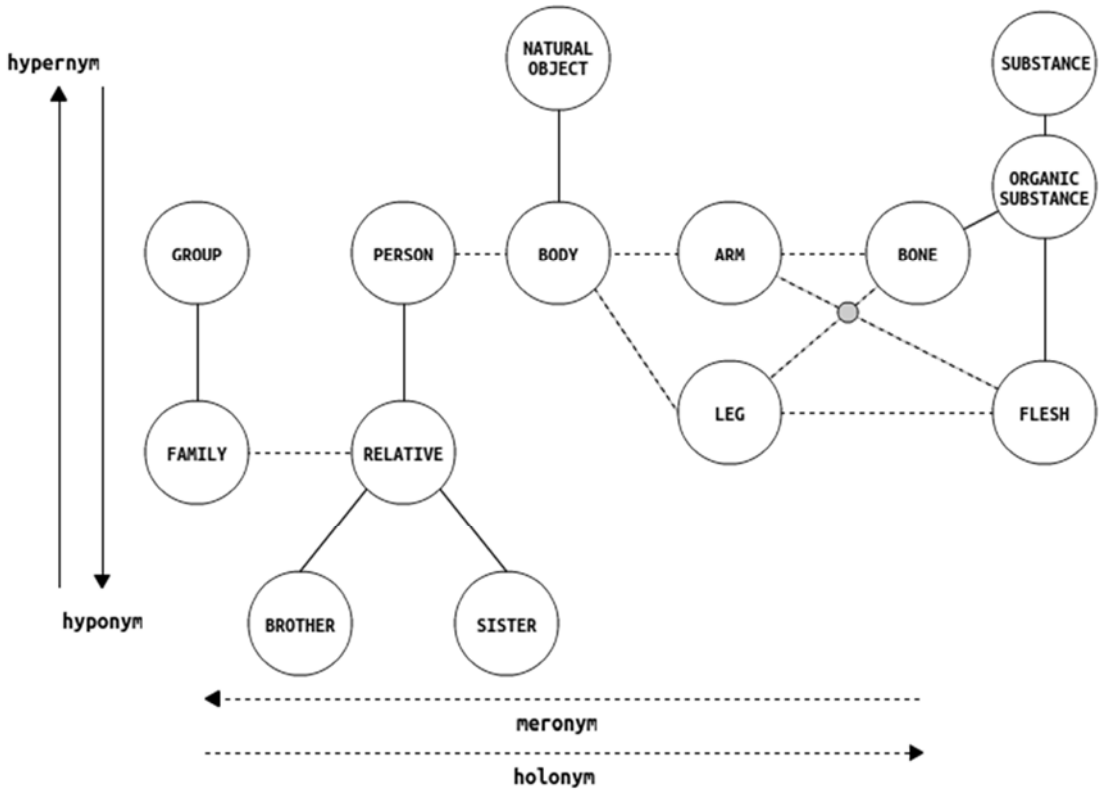

Figure 3. *Network representation of several semantic relations among an illustrative variety of lexical concepts*

As seen in Figure 3, the first two types (hyponym and hypernym) are designed to model the specialization/generalization of concepts (e.g.: 'person' is a hyponym of 'relative' -generalization- and 'body' is a hypernym of 'natural object' -specialization-). Moreover, the last types are complementary and define three types of complex semantic relations (i.e., component parts, substantive parts, and member parts) between concepts (e.g., 'bone' is a meronym of 'arm' and 'person' is holonym of 'body'). Furthermore, we should consider that the semantic relations represented in Figure 3 are transitive. Hence, 'flesh' is a holonym of 'person', and a hyponym of 'natural object'.

Taking into account the semantic information included in WordNet, we consider it appropriate to take advantage of hypernym and hyponym relations to design our topic guessing methodology. Using both hyponym and hypernym relations, WordNet can be seen as an ontology [67] hierarchically structured in levels, where the synset at the root level (called 'entity') encompasses all available synsets.

To find e-mail topics, a hierarchical level (h) should be selected in order to semantically group terms (synsets) into more generic topics (synsets close to the ontology root). Thereby, topic guessing (see '*guess e-mail topics'* process in stage three of Figure 2) entails grouping message terms into k topics $(T_1, T_2, ..., T_k)$ where k is the number of synsets belonging to any level $l \leq h$ in the WordNet hierarchy. Each topic T_i is represented by a synset s_i that characterizes its meaning. A topic T_i is present in a message if it contains a term t belonging to a synset s_j and s_j is a hyponym of s_i (one of the representative synsets for the topic T_i) or

s_j=s_i. Therefore, by using Equation 1, it is possible to guess the set of topics (*TM*) of a message *M* containing the synsets $\{s_j \mid j \in J\}$ where *J* is an index set:.

$$TM = \bigcup_{i=1,..,k} \{T_i \mid \exists \mathrm{j} \in J, hyponym(s_i, s_j) \vee (s_i = s_j)\} \quad (1)$$

where $hyponym(s_i, s_j)$ is true when s_j is hyponym of s_i.

At first glance, our proposal to guess the topics presents two important advantages: (*i*) the possibility of selecting the generalization level to adjust between the use of computational resources and specificity (by choosing an appropriate level *h* in the WordNet hierarchy); and (*ii*) the need to consider that one single message can handle multiple topics.

Finally, during the last stage of the whole process (see '*instance representation'* in Figure 2) the knowledge is represented for its use with the selected ML technique. In order to use the wide variety of ML techniques, different representations are possible: (*i*) binary (1 when topic is found, 0 otherwise), (*ii*) frequency (number of appearances of the same topics) or (*iii*) continuous (using different ponderation schemes such as that (Φ) included in Equation 2).

$$\Phi = \frac{\sum_{i=1}^{k} occurrences(T_i)}{\#TM} \quad (2)$$

where $occurrences(T_i)$ represents the number of times a specific topic (T_i) appears in a message, and $\#TM$ depicts the total number of different topics present in the e-mail. Additionally several knowledge approximation techniques such as those described in [68] can be easily modelled by using this Φ function.

In order to test the suitability of our proposal in conjunction with different ML techniques, we have designed an exhaustive experimentation protocol (selected dataset, performance evaluation metrics, ML techniques, etc.). The next section provides a detailed description together with the justification for the decisions made at the different experiment stages.

4. Experimental protocol

In order to evaluate the performance and accuracy of our feature extraction methodology, we designed and executed a straightforward and reproducible benchmarking protocol. This section introduces the experimental details and presents the results achieved. Specifically, Section 4.1 includes a summary of all the publicly available datasets and advocates the suitability of the used dataset, while Section 4.2 presents a briefly description of the ML techniques used to accomplish the experimental protocol together with the configuration parameters used for each ML technique. Finally, Section 4.3 shows the workflow comprising the designed experimental protocol.

4.1 Corpus selection

Taking into account the relevance of the delivery of spam contents as an obstacle to the successful and positive experience of Internet users, many researchers and organizations have dedicated an extraordinary amount of time and effort to meticulously compile their own collection of e-mails. Additionally, and motivated by the need for a standardized framework to ensure the reproductivity of the achieved results, a large number of e-mail datasets are publicly available [8].

As a result of the decisions adopted by owners of e-mail corpora, the nature of the messages compiled and the absence of a unified e-mail gathering protocol, it is easy to note that public datasets are distributed by using different formats, and comprise messages with very different properties. These issues should be taken into account when designing an experimental protocol because of the direct dependence they have with both the model to be tested and the results achieved. To this end, and motivated by the need to have full access to the content of each email, we studied all corpora distributed following the RFC 2822 format [69]. The standardized syntax of emails following this structure makes it possible to (*i*) simplify the access and extraction of the required information from each email, and (*ii*) ensure the reproductivity of the results. Table 1 presents a collection of well-known RFC 2822 corpora emphasizing their most important features (i.e., language, availability of duplicates, message source, and number of spam and legitimate e-mails).

Table 1. *Summary of available corpuses for anti-spam filtering research and development.*

Collection name	Multi language	Contain duplicates	Time period	Message Source	Number of ham e-mails	Number of spam e-mails	Total number of e-mails
SpamAssassin [70]	×	×	[2002 - 2006]	Forums and anonymous donations	4150	1897	6047
2005TrecSpam [26]	×		[2005]		39399	52790	92189
2006TrecSpam [26]	✓	✓	[2006]	Multiple sources	13238	24584	37822
2007TrecSpam [26]			[2007]		25220	50199	75419
CSDMC2010 [71]	×	×	[2010]	Selected for ICONIP 2010	2949	1378	4327
Bruce Guenter [72]	✓	✓	[1997-present]	Own contribution	-	>1M	>1M
Enron Corpus [73]	×	×	[1998 - 2003]	Enron Electrical Company	34519	-	34519

As shown in Table 1, last two corpora comprise only single class messages (only legitimate or spam emails) and contain a large volume of instances, while the others include e-mails belonging to both (spam and legitimate) classes. As stated in [8], selecting an adequate corpus size is mandatory to ensure an appropriate balance between the use of computational resources and the ability to obtain representative results from a statistical point of view. To execute an in-depth analysis of our proposal, we consider it essential to use a large-sized corpus because (*i*) using a large-sized dataset is essential to corroborate the suitability of our feature selection methods (the amount of information is directly related to the quality of the analysis), and (*ii*) the low computational cost of our approximation facilitates the easily handling of a large amount of information. These issues reflect the unsuitability of SpamAssassin to carry out the experiments despite its great popularity in the spam filtering domain [5,8,12,16].

Other key aspects to take into account when choosing a dataset that will create a realistic environment for experimental purposes include (*i*) the existence of duplicates (very common in a spam environment), (*ii*) the presence of both spam and ham messages (needed to build the feature selection model), and (*iii*) the availability of emails comprising a consecutive time series. The inclusion of these aspects will serve to measure the robustness of the model to deal with the different types of concept drift commonly present in the spam filtering domain [74,75]. Additionally, and taking into account the language limitations of WordNet database (only available in English); it is important to choose a corpus compatible with this issue. Motivated by the absence of a corpus addressing all of these issues, we decided to create a customized dataset by joining the English e-mails extracted from Bruce Guenter together

with Enron corpora over a period of five years (1998-2003). The resulting dataset has a realistic proportion between ham and spam messages (64 %ham, 36 %spam) together with a large size (more than 350.000 emails).

4.2 Instance Representation and Evaluated Techniques

Due to the existence of a great amount of ML techniques, together with time and computer resource limitations, and in order to validate the worth of our feature selection method, we selected a set of representative classification methods. The set of ML techniques evaluated comprises both simple and ensemble classifiers that have been largely used to filter spam. Moreover, an appropriate instance representation should be selected for the target models.

Naïve Bayes classifiers are simple and efficient linear classifiers. These techniques take advantage of the Bayes theorem where joint probabilities are achieved through the hypothesis of reciprocally independent events and, therefore, by multiplying individual probabilities. Despite the non-realistic assumption of independence, the use of the Naïve Bayes classifier is very popular in the spam-filtering domain due to its ease of implementation, the accurate results achieved, and its low computational resource requirements [76]. In order to compute the probability of a message being spam when it contains a certain token/topic, we used a multivariate Bernoulli approach [73]. Moreover, for experimentation purposes we chose the implementation provided by the e1071 [68] package available in R statistical software [77].

SVM [78,79] is one of the most widely used ML techniques for regression and classification purposes. SVMs suppose that each instance (represented by n feature values) is a point in an n-dimensional space. Given this situation, the SVM algorithm is able to find a hyperplane to geometrically distinguish between spam and ham instances (represented as points), thus maximizing the distance between the hyperplane and the instances of both classes. SVMs has been successfully used to filter spam [68]. Due to the large number of features, we selected the implementation provided in the e1071 [68] R package with a linear kernel [80] for our experiments.

Logistic regression is a method to compute the probability of a spam instance through a linear combination of input features with a link function. The class of a message (ham/spam) is determined by comparing the estimated probability with 0.5. For this work, we selected the speedglm R package using a binomial family [81].

C4.5 is a software extension of the basic ID3 algorithm designed by Quinlan [82]; it is able to induce classification rules in the form of decision trees from a set of given instances. Although this technique has not been widely used to filter spam in its original form (due to its weak performance), we decided to include it in our study to highlight the importance of a good feature selection method. For our experiments, we used the implementation provided in Weka learning environment[4] (J48) through RWeka extension available in R statistical software [83].

[4] Available at http://www.cs.waikato.ac.nz/ml/weka/

Classifier ensembling (Bagging [84,85], Random Forests [86], or Adaboost [87]) is the most extended form of using weak classifiers such as C4.5 to build up accurate models. Bagging and Random Forests schemes are based on combining the results of different weak models created by the process of training with different subsets of the whole amount of training instances. The main difference between the two forms is the method used in Random Forests to select a subset of features from all candidate splits. Furthermore, Boosting algorithms combine the results of different classifiers built by using different collections of attributes from all instances. Given that these approaches have been widely used to filter spam e-mail, we included all of them in our study. For our experiments, we selected the implementation included in R randomForest package [88]. Moreover, Weka implementations of AdaBosst [73] and Bagging models have been used to test Adaboost and Bagging models (through the usage of RWeka package [83]). All of these ensembling methods were tested using C4.5 as the internal classifier.

Finally, recent works have demonstrated the suitability of Rough Sets as a method to filter spam [16]. These methods are based on the use of the Rough Sets theory proposed by Pawlak [89] to identify irrelevant attributes and build up a rule-based classifier system. For experimentation purposes, we selected the implementation provided in the RoughSets R package [90].

4.3 Experimental benchmarking

In order to evaluate the performance of the proposed feature selection model, we selected the previously described classifiers because of their widespread use and previously established suitability in a particular and specific domain such as spam filtering [14–16,22,73].

As seen in Figure 4, our experimental protocol comprises four main stages: (*i*) data pre-processing; (*ii*) feature selection; (*iii*) 10-fold cross-validation over several ML models; and finally, (*iv*) results interpretation and comparison.

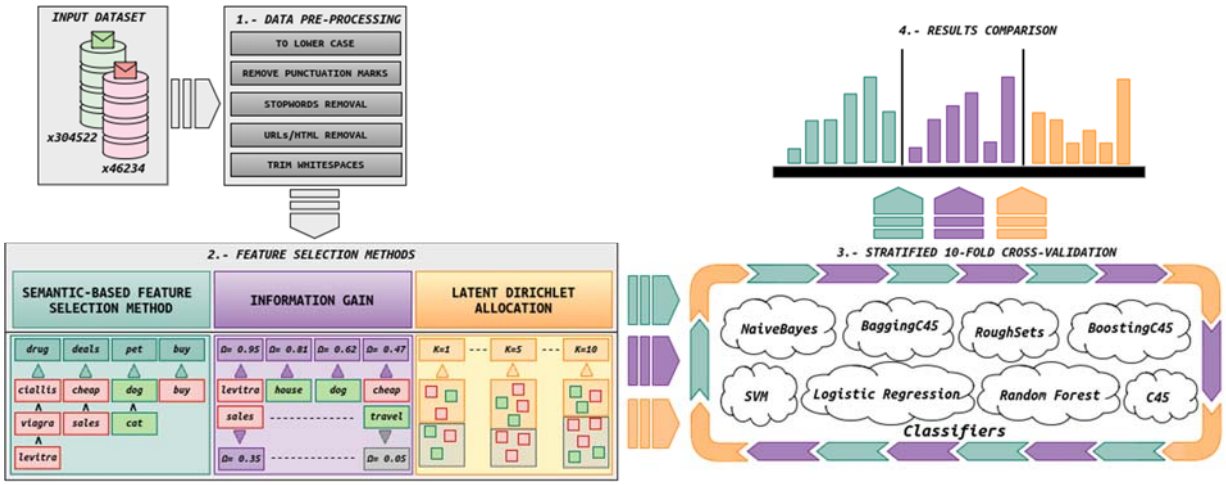

Figure 4. *Experimental protocol design.*

The first stage (*see data pre-processing* in Figure 4) carries out several pre-processing operations over the previously extracted text. In detail, the text is tokenised using spaces ('\s + '), tabs ('\t + ') and newline ('\n') characters as word separators. Additionally, in order to both reduce the computational overhead and improve the performance of the feature selection methods, we (*i*) remove all the possessive forms of tokens (termination 's) and (*ii*) discard tokens composed exclusively of digits, included in an English stopword list [91], containing URL links, or belonging to HTML tags.

The second stage includes the execution of feature selection techniques to obtain the most suitable features. To ensure an adequate use of computational resources without succumbing to an excessive generalization or specification that compromises the performance results of our methodology, we considered it appropriate to use the 181 topics available throughout the first four levels of the WordNet (distributed by levels at follows: 168 level-4, 10 level-3, 2 level-2 and 1 level-1). Additionally, to evaluate the performance of our proposal, we decided to use both, Information Gain and Latent Dirichlet Allocation techniques as reference measures. Information Gain was selected due to its widely demonstrated suitability in the spam-filtering domain resulting from its good balance between performance, use of computational resources and time consumption [63,92]. Moreover, to guarantee an equitable comparison with our method (in terms of amount of information), we decided to choose the same number of features from Information Gain as those used in our proposal (181 features with best entropy).

Latent Dirichlet Allocation is a commonly-used unsupervised method in the text-mining domain due to its ability to group words that best encompass (or cover) each topic [39,40]. However, the need to manually specify the number of topic divisions (defined as k) together with the intense use of computational resources and, consequently, a significant increase in the time consumption, impedes the use of LDA in large datasets (especially with high values of k). This scenario, together with the size of the selected dataset, impedes the execution of LDA using a topic division of 181. Thus, we designed an experimental protocol in order to obtain the k value that guarantees the best compromise between execution time and resource consumption without degrading the performance of the model. Figure 5 below shows the perplexity and time consumption values obtained after computing LDA with an increasing number of topic divisions (up to $k=32$) using 10-fold cross-validation methodology. As seen in Figure 5, the higher value of k implies a significant increase in execution time, while the model perplexity measure (ability to generalize) remains practically unalterable (especially from $k=10$). Keeping this fact in mind, we consider the use of LDA with $k=10$ due to the adequate balance achieved between time consumption and model perplexity.

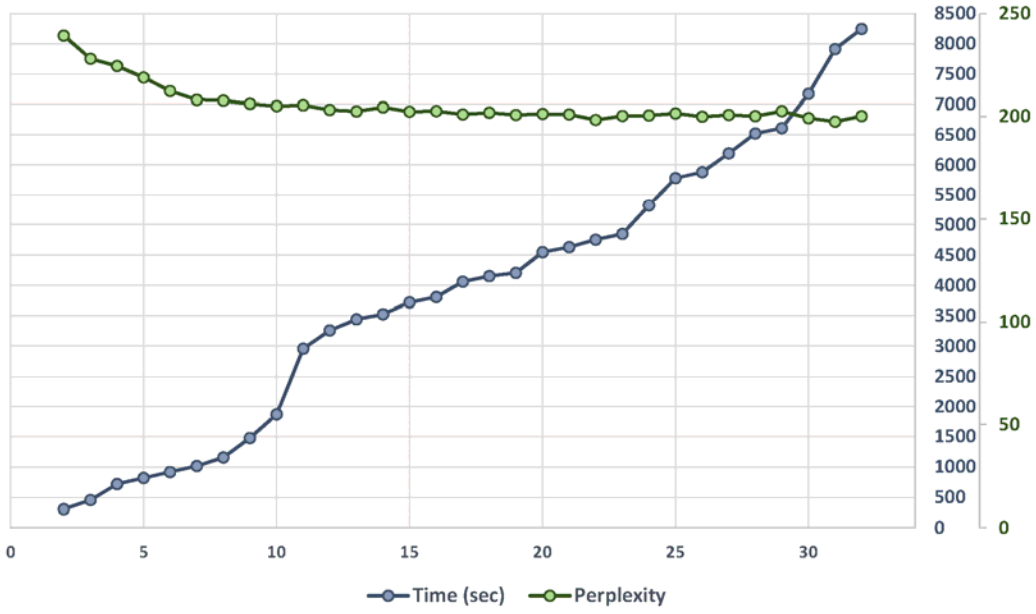

Figure 5. *Perplexity and time consumed values for each k value.*

During the third stage, ML techniques are executed in order to measure the quality of the feature selection methods. Additionally, in order to generalize the prediction results to an independent dataset, we run a k-fold stratified cross-validation scheme (with k=10) [93]. Eight different classifiers are included in the experimental protocol in order to demonstrate the accuracy of the feature selection methods with independence of the classifier used and, therefore, obtain a global perspective on the performance of each method.

Finally, during the last step (see *results comparison* in Figure 4) a single confusion matrix is generated for each ML technique by computing the average mean of the results (FP, FN, TP, TN) obtained from the execution of each experiment.

The next section presents review the results achieved during the described experimental protocol in order to demonstrate the suitability of our proposal.

5. Results and discussion

As previously detailed, our experimental protocol involves the execution of three feature selection methods (*Information Gain, Latent Dirichlet Allocation,* and *Semantic-based Feature Selection*). To measure the accuracy of each method, we executed the same 10-fold cross validation scheme for each classifier. The results of different folds were grouped into a confusion matrix.

One of the primary outcomes of a binary classification experiment is the confusion matrix achieved by classifiers. The confusion matrix brings together the number of different types of errors and hits including: (*i*) false positive errors (FP, legitimate messages classified as spam); (*ii*) false negative errors (FN, undetected spam e-mails); (*iii*) true positive hits (TP, number of spam messages detected); and (iv) true negative hits (TN, number of legitimate messages correctly classified). A cursory review of the confusion matrix provides a general

perspective of the performance of the analysed methods. Table 2 shows a confusion matrix summary that combines the number of hits (Accuracy) and groups the results using percentages to facilitate their comprehension.

Table 2: *Summary of confusion matrices for analysed configurations in percentages*

	Information Gain			**Latent Dirichlet Allocation**			**Topic Guessing**		
Classifier Type	**ACC**	**FN**	**FP**	**ACC**	**FN**	**FP**	**ACC**	**FN**	**FP**
Naïve Bayes	83.3	11.5	5.2	87.0	7.1	5.9	76.9	6.5	16.6
SVM	87.7	12.1	0.2	94.9	3.8	1.3	90.8	7.4	1.8
C4.5	88.7	10.8	0.5	95.5	3.8	0.7	97.6	1.2	1.2
Adaboost	89.1	10.6	0.3	95.4	4.1	0.5	99.1	0.4	0.5
Bagging	88.0	11.7	0.3	95.2	4.1	0.7	95.1	4.0	0.9
Random Forests	89.1	10.7	0.2	95.6	3.8	0.6	99.2	0.6	0.2
Logistic Regression	87.7	10.2	2.1	92.6	3.2	4.2	90.6	7.1	2.3
Rough Sets	89.0	10.9	0.1	96.0	3.8	0.3	99.4	0.4	0.2

From the results included in Table 2, we can appreciate the strong performance achieved by all classifiers with all the feature selection methods (more than 80% of the messages were correctly classified). However, it is also quite clear that the performance of our feature selection method (Topic Guessing) and that of LDA are both significantly better (more than 95% of messages are correctly classified with most classifiers) than the performance acquired by Information Gain.

A core problem derived from analysing results using only the confusion matrix is induced by the unbalanced distribution of ham and spam messages. In fact, if 90% of messages included in experimental dataset were legitimate, an approach that classifies all messages as ham would be a reasonably better approach. To solve this issue, the kappa coefficient [79] may be used to compare real classes of messages and classifier outputs. The kappa coefficient was designed to measure the agreement between two different qualitative diagnostics (which may be the result of two different classification methods). The agreement of the classifier outputs and real classifications (kappa) are shown in Figure 5.

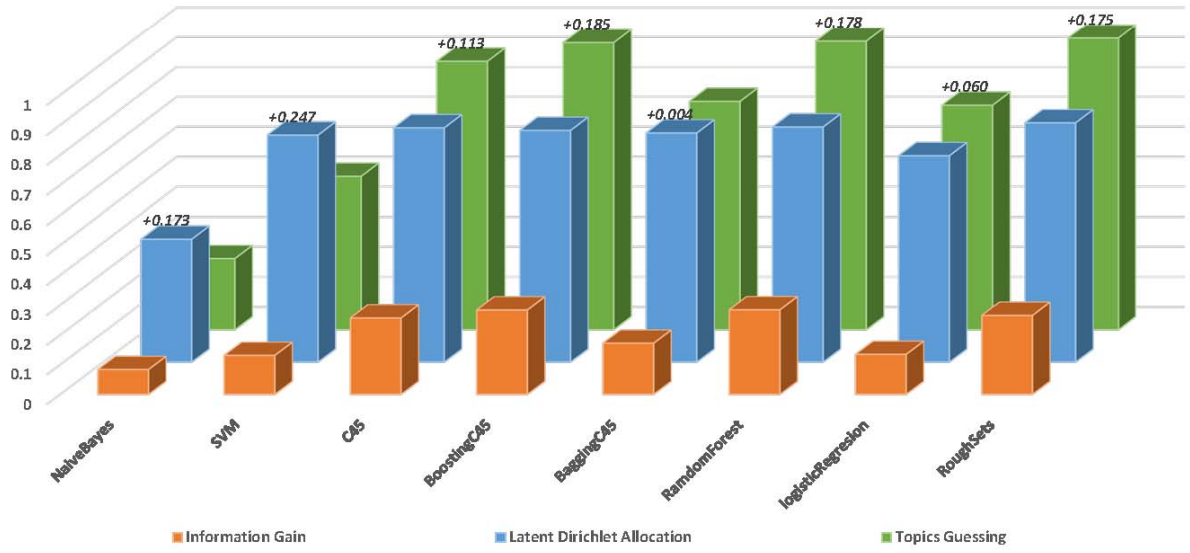

Figure 6. *Kappa values for analysed methods*

When applied to our topic guessing method, the kappa coefficients for the same classifiers achieves a better evaluation in most of the analysed scenarios. Therefore, from a classification perspective (without taking the asymmetric cost of errors into account), our feature selection method is significantly better with regard to Information Gain and achieves higher overall kappa values than LDA in most of the executed classifiers.

A Chi-squared test [79] is performed for each classifier using the homogeneity of kappa measure as a null hypothesis. For all different classifiers the p-values were less than 0.001 in the comparison between methods (i.e., Topic Guessing vs Information Gain, Topic Guessing vs LDA and Information Gain vs LDA). Therefore, from a statistical point of view, we found significant differences between our proposal and the other methods for each classifier.

The asymmetric cost of FP and FN errors is one of the most relevant problems that must be taken into account during the evaluation of spam filtering proposals. Several commonly used filtering performance metrics (precision/recall, F-score or Total Cost Ratio -TCR- [8]) allow evaluating different approaches from a cost-sensitive perspective. The use of recall and precision measures is very popular to evaluate spam filters. These measures provide complementary information regarding the ability to find spam messages and avoid FP errors respectively. Both measures take into account any messages correctly identified as spam, and penalize classifiers with lower FP errors. Although these measures should not be used independently, F-score emerged as a method to combine the information provided by both measures in a single measure. Table 3 shows a cost-sensitive evaluation of the configurations (IG = Information Gain, LDA = Latent Dirichlet Allocation and TG = Topic Guessing) using precision/recall and F-score measures.

Table 3. Cost-sensitive evaluation for analysed configurations

	Recall			Precision			F-score		
Classifier Type	**IG**	**LDA**	**TG**	**IG**	**LDA**	**TG**	**IG**	**LDA**	**TG**
Naïve Bayes	0.13	0.46	0.51	0.24	0.50	0.29	0.17	0.48	0.37
SVM	0.08	0.71	0.44	0.82	0.88	0.76	0.15	0.79	0.56
C4.5	0.18	0.71	0.91	0.83	0.93	0.91	0.29	0.81	0.91
Adaboost	0.19	0.69	0.97	0.89	0.95	0.96	0.32	0.80	0.96
Bagging	0.11	0.71	0.70	0.84	0.92	0.91	0.20	0.79	0.79
Random Forests	0.19	0.76	0.95	0.93	0.94	0.99	0.32	0.81	0.97
Logistic Regression	0.12	0.75	0.73	0.40	0.71	0.46	0.18	0.73	0.70
Rough Sets	0.18	0.71	0.97	0.95	0.97	0.99	0.30	0.82	0.98

As seen in Table 3, the higher values are achieved when our proposal (Topic Guessing) is used, even though LDA obtains better precision values (avoiding FP errors) in most scenarios. Moreover, recall values are clearly better for our approach. The summarized version of recall and precision also confirm the satisfactory results of using our topic guessing method, mainly when used in conjunction with RoughSets.

TCR uses a parameter (λ) to establish the cost of an FP error with regard to an FN error. Thus, using λ=9 means that an FP error causes a problem similar to that caused by 9 FN errors. A TCR evaluation under 1 indicates that the proposal should be discarded in the modelled cost scenario. In order to complete a cost-sensitive analysis, Table 4 contains TCR scores achieved by the compared approaches.

Table 4. TCR evaluation for different scenarios

	λ=1			λ=9			λ=999		
Classifier Type	**IG**	**LDA**	**TG**	**IG**	**LDA**	**TG**	**IG**	**LDA**	**TG**
Naïve Bayes	0.4121	1.0173	1.0073	0.0634	0.2200	0.3101	0.0006	0.0022	0.0008
SVM	0.1093	2.5784	0.8315	0.0124	0.8456	0.1124	0.0001	0.0100	0.0071
C4.5	0.2483	2.9301	5.4839	0.0287	1.3172	1.1189	0.0003	0.0191	0.0107
Adaboost	0.2605	2.8708	13.5334	0.0297	1.5785	2.9853	0.0003	0.0278	0.0240
Bagging	0.1476	2.7362	2.0472	0.0168	1.2283	0.2734	0.0002	0.0178	0.0141
Random Forests	0.2492	2.9721	14.9234	0.0281	1.3968	1.9978	0.0003	0.0210	0.0810
Logistic Regression	0.2771	1.7915	1.4079	0.0362	0.3243	0.4775	0.0003	0.0032	0.0058
Rough Sets	0.2216	3.2172	20.9250	0.0249	2.0608	3.0072	0.0002	0.0453	0.0837

As shown in Table 4, none of the configurations where IG feature selection method is used are adequate in a real scenario (TCR evaluation < 1). However, both, LDA and our topic selection method are suitable for most configurations using λ values of 1 and 9. Concretely, the combination of our proposal together with RoughSets achieves the best TCR values with λ values of 1 and 9.

The receiver operating characteristics (ROC) analysis is a useful tool to graphically represent and analyse overall classifier performance [78]. Figure 6 graphically represents a ROC curve where each classifier has been represented as a point (false positive rate, true positive rate) to highlight relative trade-offs between true positives and false negatives. Graphically, the best classifier results are those at the top-right of the graph. Moreover, the points over the diagonal represent the results achieved randomly (i.e. a flip of a coin will decide the class of the message) so a reasonably good classifier should be represented over the diagonal.

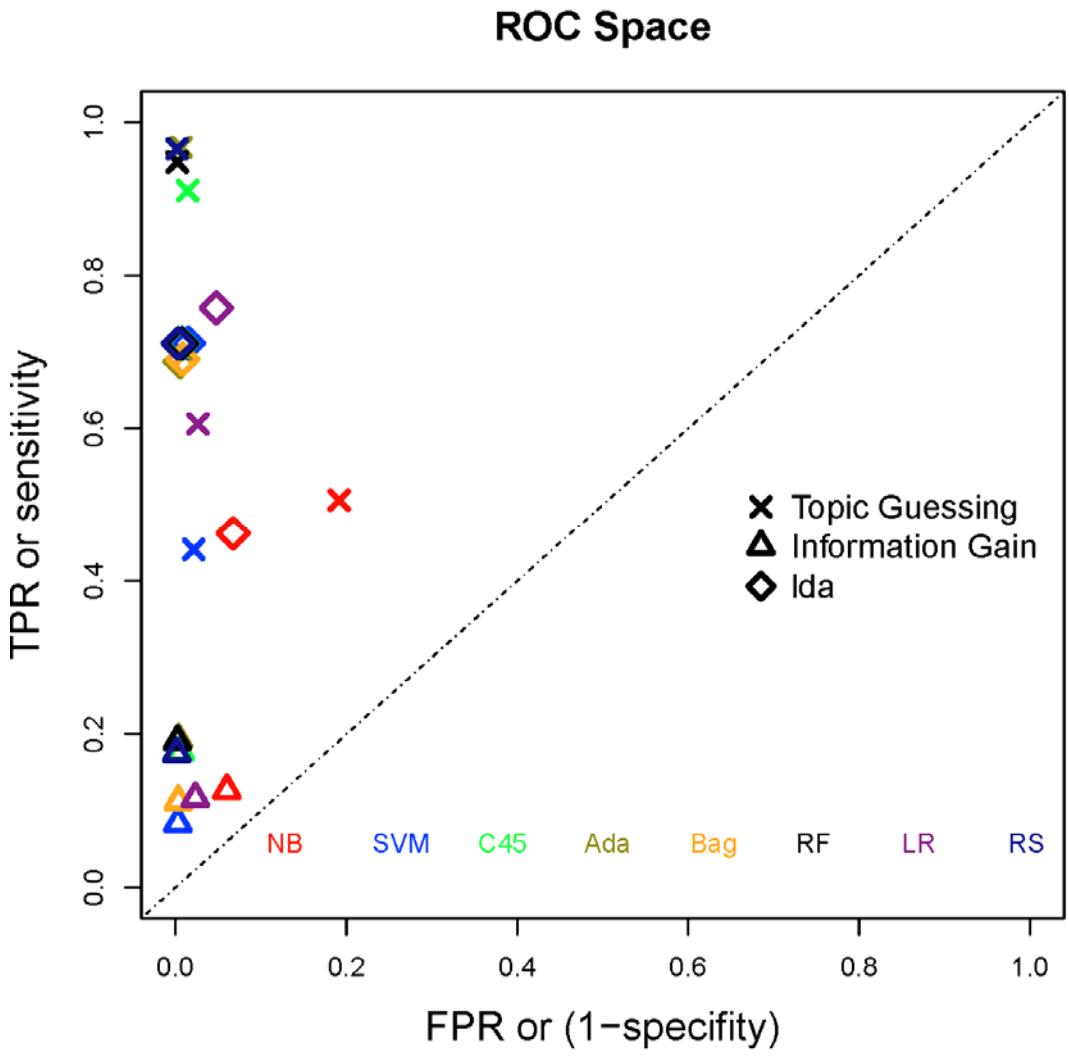

Figure 7. *The ROC space and plots of the eight prediction classifiers.*

As shown in Figure 7, all the classifiers using our Topic Guessing method obtain a better predictive power with respect to Information Gain technique. Although LDA achieves better performance values for Naïve Bayes, SVM and Logistic Regression classifiers, from a global

perspective, it is readily apparent that Topic Guessing facilitates the performance of the classifiers. Specifically, the best classifiers able to take advantage of our topic guessing method are Adaboost, Random Forests, and above all, Rough Sets. In fact, and taking into account the achieved results, we can conclude the importance of combining Topic Guessing as a Feature Selection Technique, together with Rough Sets, in order to increase the accuracy of the spam filters. The next section compiles the conclusions achieved from this study and details future research directions.

6. Conclusions and future work

This study introduced a novel feature-selection method able to take advantage of semantic information to detect topics, and presented its application to spam filtering. We compared the results achieved by using our proposal against Information Gain (the most widely used feature selection method in spam filtering domain) and Latent Dirichlet Allocation (a well-known unsupervised technique in the text-mining domain). The analysed classifiers achieved a significant increase in performance when the novel approach is applied. These results support the possibility of filtering spam by using topics instead of words (tokens). Moreover, the new feature selection method is able to automatically discard/identify noise from messages (because it finds words that are not found in semantic ontologies).

With regard to the results shown in Section 5, our proposal enables classifiers to discover more spam messages than when using classic approaches (see Table 1 and Figures 5 and 6). Although the number of errors could slightly increase with some ML techniques, cost-sensitive measures showed a worthwhile increase in performance (see Tables 3 and 4). We believe that these observations are derived from the grouping nature of our method. Thus, traditional filter methods are based on discarding (more or less relevant) words (tokens), while our proposal is able to group message knowledge (words/tokens) into topics, thus avoiding the loss of information.

Our method also presents additional benefits such as the ability to know the features (columns) before executing any feature selection method. In fact, all messages used in our experiments were represented with 181 features (with h=4). This issue implies more parallelization capabilities of the proposed method with regard to other feature selection methods (filter, wrapped or embedded). Wrapped methods also present additional computer requirements to evaluate the suitability of different groups of features. Finally, our method (as with other filter and wrapped approaches) can be combined with the use of embedded methods (Rough Sets or SVM). In fact, the promising results achieved by combining our feature selection approach together with the Rough Sets theory makes it possible to increase the classification capabilities of new filtering systems. Additionally, with regard to the use of computational resources, the topic guessing approach has more requirements than filter methods but fewer than wrapped methods.

Although results achieved seem promising, we are sure that new improvements will be included to this approach. In fact, the use of a manually specified level (h) to establish possible topics is not the most adequate way of operating. We are aware that some

hierarchical clustering methods [94] could provide a reliable way of finding adequate topics that could be used to represent messages using the semantic ontology. Moreover, although this seems obvious, we should also explore precisely how obfuscation tricks could be efficiently detected during the feature selection stage. Furthermore, the exploration of different feature representation schemes with each ML technique and the inclusion of different languages are also valuable areas of study. Finally, we are aware of the applicability of this feature selection method in many other disciplines, such as the study of user interests and, hence, recommendation systems or the automatic classification of contents in all domains.

Acknowledgements.

D. Ruano-Ordás was supported by a Postdoctoral fellowship from the Xunta de Galicia. Additionally, this work was partially funded by Consellería de Cultura, Educación e Ordenación Universitaria (Xunta de Galicia) and FEDER (European Union).

SING group thanks CITI (Centro de Investigación, Transferencia e Innovación) from University of Vigo for hosting its IT infrastructure.

Bibliography

[1] D. Tamir, How To Protect Yourself From Instaspam, Readwrite. (2015). http://readwrite.com/2015/04/15/instagram-spam-instaspam-how-to-avoid (accessed April 30, 2017).

[2] M. Allton, How to Turn Off Social Media Spam, Soc. Media Hat. (2013). https://www.thesocialmediahat.com/blog/how-turn-social-media-spam-02122013 (accessed April 30, 2017).

[3] A.J. Trivedi, P.Q. Judge, S. Krasser, Analyzing Network and Content Characteristics of Spim Using Honeypots, in: Proc. 3rd USENIX Work. Steps to Reducing Unwanted Traffic Internet, USENIX Association, Berkeley, CA, USA, 2007: p. 3:1--3:9. http://dl.acm.org/citation.cfm?id=1361436.1361439.

[4] D. Guarini, Here's How To Stop Snapchat Spam, Huffingt. Post. (2014). http://www.huffingtonpost.com/2014/01/13/snapchat-spam_n_4590515.html (accessed April 30, 2017).

[5] J. Fdez-Glez, D. Ruano-Ordás, R. Laza, J.R. Méndez, R. Pavón, F. Fdez-Riverola, WSF2: A Novel Framework for Filtering Web Spam, Sci. Program. 2016 (2016). doi:10.1155/2016/6091385.

[6] J. Fdez-Glez, D. Ruano-Ordas, J.R. Méndez, F. Fdez-Riverola, R. Laza, R. Pavón, A dynamic model for integrating simple web spam classification techniques, Expert Syst. Appl. 42 (2015) 7969–7978. doi:10.1016/j.eswa.2015.06.043.

[7] M. Golden, The End of Gnutella?, Free. to Tinker. (2011). https://freedom-to-tinker.com/2011/08/03/end-gnutella/ (accessed April 30, 2017).

[8] N. Pérez-Díaz, D. Ruano-Ordás, F. Fdez-Riverola, J.R. Méndez, SDAI: An integral evaluation methodology for content-based spam filtering models, Expert Syst. Appl. 39 (2012) 12487–12500. doi:10.1016/j.eswa.2012.04.064.

[9] Josh, FSpamlist, (2007). https://fspamlist.com (accessed April 30, 2017).

[10] D. Gudkova, M. Vergelis, T. Demidova, Nadezhda Shcherbakova, Spam and phishing

in Q2 2016, Securelist. (2016). https://securelist.com/analysis/quarterly-spam-reports/75764/spam-and-phishing-in-q2-2016 (accessed April 30, 2017).
[11] Apache Software Foundation, The Apache SpamAssassin project - The #1 Enterprise Open-Source Spam Filter, 2017 (2007). http://spamassassin.apache.org (accessed November 7, 2016).
[12] N. Pérez-Díaz, D. Ruano-Ordás, F. Fdez-Riverola, J.R. Méndez, Wirebrush4SPAM: A novel framework for improving efficiency on spam filtering services, Softw. - Pract. Exp. 43 (2013) 1299–1318. doi:10.1002/spe.2135.
[13] A. Gray, M. Haahr, Personalised, Collaborative Spam Filtering, in: Proc. Conf. Email Anti-Spam, Mountain View, CA, USA, 2004.
[14] E. Blanzieri, A. Bryl, A survey of learning-based techniques of email spam filtering, Artif. Intell. Rev. 29 (2008) 63–92. doi:10.1007/s10462-009-9109-6.
[15] G. V Cormack, Email Spam Filtering: A Systematic Review, Found. Trends® Inf. Retr. 1 (2008) 335–455. doi:10.1561/1500000006.
[16] N. Pérez-Díaz, D. Ruano-Ordás, J.R. Méndez, J.F. Gálvez, F. Fdez-Riverola, Rough sets for spam filtering: Selecting appropriate decision rules for boundary e-mail classification, Appl. Soft Comput. J. 12 (2012). doi:10.1016/j.asoc.2012.05.024.
[17] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, P. Samarati, An Open Digest-based Technique for Spam Detection, in: Proc. 2004 Int. Work. Secur. Parallel Distrib. Syst., 2004: pp. 559–564. http://spdp.di.unimi.it/papers/pdcs04.pdf.
[18] S. Kitterman, Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Internet Eng. Task Force. (2014). https://tools.ietf.org/html/rfc7208 (accessed April 30, 2017).
[19] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas, DomainKeys Identified Mail (DKIM) Signatures, (2007). https://www.ietf.org/rfc/rfc4871.txt (accessed November 7, 2016).
[20] Rhyolite Software, Distributed Checksum Clearinghouses, (2001). https://www.dcc-servers.net/dcc/ (accessed November 7, 2016).
[21] A. Ramachandran, D. Dagon, N. Feamster, Can DNS-Based Blacklists Keep Up with Bots?, in: Third Conf. Email Anti-Spam, Mountain View, CA, 2006. doi:10.1.1.123.2270.
[22] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos, P. Stamatopoulos, Learning to Filter Spam E-Mail: {A} Comparison of a Naive Bayesian and a Memory-Based Approach, CoRR. cs.CL/0009 (2000). http://arxiv.org/abs/cs.CL/0009009.
[23] E. Conrad, Detecting Spam with Genetic Regular Expressions, SANS Inst. InfoSec Read. Room. (2007). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.183.4622&rep=rep1&type=pdf.
[24] J.R. Méndez, D. Glez-Peña, F. Fdez-Riverola, F. Díaz, J.M. Corchado, Managing irrelevant knowledge in CBR models for unsolicited e-mail classification, Expert Syst. Appl. 36 (2009) 1601–1614. doi:http://dx.doi.org/10.1016/j.eswa.2007.11.037.
[25] S. Halder, R. Tiwari, A. Sprague, Information extraction from spam emails using stylistic and semantic features to identify spammers, in: 2011 IEEE Int. Conf. Inf. Reuse Integr., 2011: pp. 104–107. doi:10.1109/IRI.2011.6009529.
[26] NIST, Text Retrieval Conference, (2007). http://trec.nist.gov/ (accessed November 7, 2016).
[27] T.A. Almeida, T.P. Silva, I. Santos, J.M.G. Hidalgo, Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering, Knowledge-Based Syst. 108 (2016) 25–32. doi:https://doi.org/10.1016/j.knosys.2016.05.001.

[28] E.M. Bahgat, I.F. Moawad, Semantic-Based Feature Reduction Approach for E-mail Classification, in: A.E. Hassanien, K. Shaalan, T. Gaber, A.T. Azar, M.F. Tolba (Eds.), Proc. Int. Conf. Adv. Intell. Syst. Informatics 2016, Springer International Publishing, Cham, 2017: pp. 53–63. doi:10.1007/978-3-319-48308-5_6.
[29] S. Suganya, C. Gomathi, S. Mano Chitra, Syntax and Semantics based Efficient Text Classification Framework, Int. J. Comput. Appl. 65 (2013) 18–21.
[30] D. Padmaraju, V. Varma, Applying Lexical Semantics to Improve Text Classification, in: Second Symp. Indian Morphol. Phonol. Lang. Eng., Kharagpur, 2005: pp. 94–98.
[31] S. Salcedo-Sanz, G. Camps-Valls, F. Perez-Cruz, J. Sepulveda-Sanchis, C. Bousono-Calzon, Enhancing genetic feature selection through restricted search and Walsh analysis, IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev. 34 (2004) 398–406. doi:10.1109/TSMCC.2004.833301.
[32] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, Knowl. Inf. Syst. 12 (2007) 95–116. doi:10.1007/s10115-006-0040-8.
[33] F. Alonso-Atienza, J.L. Rojo-Álvarez, A. Rosado-Muñoz, J.J. Vinagre, A. García-Alberola, G. Camps-Valls, Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection, Expert Syst. Appl. 39 (2012) 1956–1967. doi:http://dx.doi.org/10.1016/j.eswa.2011.08.051.
[34] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28. doi:http://dx.doi.org/10.1016/j.compeleceng.2013.11.024.
[35] Y. Saeys, I. Inza, P. Larrañaga, A Review of Feature Selection Techniques in Bioinformatics, Bioinformatics. 23 (2007) 89–118. doi:10.1093/bioinformatics/btm344.
[36] V. Bolón-Canedo, N. Sánchez-Maroño, J. Cerviño-Rabuñal, Scaling Up Feature Selection: A Distributed Filter Approach, in: C. Bielza, A. Salmerón, A. Alonso-Betanzos, J.I. Hidalgo, L. Martínez, A. Troncoso, E. Corchado, J.M. Corchado (Eds.), Adv. Artif. Intell. 15th Conf. Spanish Assoc. Artif. Intell. CAEPIA 2013, Madrid, Spain, Sept. 17-20, 2013. Proc., Springer Berlin Heidelberg, Berlin, Heidelberg, 2013: pp. 121–130. doi:10.1007/978-3-642-40643-0_13.
[37] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
[38] J. Ma, Y. Zhang, J. Liu, K. Yu, X. Wang, Intelligent SMS Spam Filtering Using Topic Model, in: 2016 Int. Conf. Intell. Netw. Collab. Syst., 2016: pp. 380–383. doi:10.1109/INCoS.2016.47.
[39] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet Allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[40] P. Nakov, Latent Semantic Analysis of Textual Data, in: Proc. Conf. Comput. Syst. Technol., ACM, New York, NY, USA, 2000: pp. 5031–5035. doi:10.1145/365143.365382.
[41] T. Hofmann, Probabilistic Latent Semantic Indexing, in: Proc. 22Nd Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., ACM, New York, NY, USA, 1999: pp. 50–57. doi:10.1145/312624.312649.
[42] B. Grün, K. Hornik, topicmodels: An R Package for Fitting Topic Models, J. Stat. Software, Artic. 40 (2011) 1–30. doi:10.18637/jss.v040.i13.
[43] W. Duch, Filter Methods, in: I. Guyon, M. Nikravesh, S. Gunn, L.A. Zadeh (Eds.), Featur. Extr. Found. Appl., Springer Berlin Heidelberg, Berlin, Heidelberg, 2006: pp. 89–117. doi:10.1007/978-3-540-35488-8_4.
[44] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and

clustering, IEEE Trans. Knowl. Data Eng. 17 (2005) 491–502. doi:10.1109/TKDE.2005.66.
[45] E.T. Jaynes, Information Theory and Statistical Mechanics, Phys. Rev. 106 (1957) 620–630. https://link.aps.org/doi/10.1103/PhysRev.106.620.
[46] C. Bae, W.-C. Yeh, Y.Y. Chung, S.-L. Liu, Feature selection with Intelligent Dynamic Swarm and Rough Set, Expert Syst. Appl. 37 (2010) 7026–7032. doi:http://dx.doi.org/10.1016/j.eswa.2010.03.016.
[47] Y. Qian, J. Liang, W. Pedrycz, C. Dang, Positive approximation: An accelerator for attribute reduction in rough set theory, Artif. Intell. 174 (2010) 597–618. doi:http://dx.doi.org/10.1016/j.artint.2010.04.018.
[48] A.K. Uysal, An Improved Global Feature Selection Scheme for Text Classification, Expert Syst. Appl. 43 (2016) 82–92. doi:10.1016/j.eswa.2015.08.050.
[49] M. Bennasar, Y. Hicks, R. Setchi, Feature selection using Joint Mutual Information Maximisation, Expert Syst. Appl. 42 (2015) 8520–8532. doi:https://doi.org/10.1016/j.eswa.2015.07.007.
[50] S.R. Sanjay, S.S. Sane, Article: JMIM: A Feature Selection Technique using Joint Mutual Information Maximization Approach, IJCA Proc. Emerg. Trends Comput. ETC 2016 (2017) 5–10.
[51] K. Javed, S. Maruf, H.A. Babri, A two-stage Markov blanket based feature selection algorithm for text classification, Neurocomputing. 157 (2015) 91–104. doi:https://doi.org/10.1016/j.neucom.2015.01.031.
[52] J. Doak, An Evaluation of Feature Selection Methods and Their Application to Computer Security, (1992). http://www.escholarship.org/uc/item/2jf918dh.
[53] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324. doi:http://dx.doi.org/10.1016/S0004-3702(97)00043-X.
[54] L. Yu, H. Liu, Redundancy Based Feature Selection for Microarray Data, in: Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., ACM, New York, NY, USA, 2004: pp. 737–742. doi:10.1145/1014052.1014149.
[55] W. Cohen, Learning rules that classify e-mails, in: AAAI Symp. Mach. Learn. Inf. Access, 1996: pp. 18–25.
[56] H. Drucker, D. Wu, V.N. Vapnik, Support vector machines for spam categorization, IEEE Trans. Neural Networks. 10 (1999) 1048–1054. doi:10.1109/72.788645.
[57] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, A Bayesian Approach to Filtering Junk E-Mail, in: AAAI Work. Learn. Text Categ., Madison, Wisconsi, 1998.
[58] J. Clark, I. Koprinska, J. Poon, A neural network based approach to automated e-mail classification, in: Proc. IEEE/WIC Int. Conf. Web Intell. (WI 2003), 2003: pp. 702–705. doi:10.1109/WI.2003.1241300.
[59] X. Yao, Y. Liu, A New Evolutionary System for Evolving Artificial Neural Networks, Trans. Neur. Netw. 8 (1997) 694–713. doi:10.1109/72.572107.
[60] H. Wang, Y. Yu, Z. Liu, SVM Classifier Incorporating Feature Selection Using GA for Spam Detection, in: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embed. Ubiquitous Comput. -- EUC 2005 Int. Conf. EUC 2005, Nagasaki, Japan, December 6-9, 2005. Proc., Springer Berlin Heidelberg, Berlin, Heidelberg, 2005: pp. 1147–1154. doi:10.1007/11596356_113.
[61] J.R. Méndez, I. Cid, D. Glez-Peña, M. Rocha, F. Fdez-Riverola, A Comparative Impact Study of Attribute Selection Techniques on Na{ï}ve Bayes Spam Filters, in: P. Perner (Ed.), Adv. Data Mining. Med. Appl. E-Commerce, Mark. Theor. Asp. 8th Ind. Conf. ICDM 2008 Leipzig, Ger. July 16-18, 2008 Proc., Springer Berlin Heidelberg, Berlin, Heidelberg, 2008: pp. 213–227. doi:10.1007/978-3-540-70720-2_17.
[62] J.R. Méndez, F. Fdez-Riverola, F. Díaz, E.L. Iglesias, J.M. Corchado, A Comparative

Performance Study of Feature Selection Methods for the Anti-spam Filtering Domain, in: P. Perner (Ed.), Adv. Data Mining. Appl. Med. Web Mining, Mark. Image Signal Min. 6th Ind. Conf. Data Mining, ICDM 2006, Leipzig, Ger. July 14-15, 2006. Proc., Springer Berlin Heidelberg, Berlin, Heidelberg, 2006: pp. 106–120. doi:10.1007/11790853_9.
[63] A. Sharaff, N.K. Nagwani, K. Swami, Impact of Feature Selection Technique on Email Classification, Int. J. Knowl. Eng. 1 (2015). http://www.ijke.org/vol1/10-E001.pdf.
[64] Z. Zhu, An Email Classification Model Based on Rough Set and Support Vector Machine, 2008 Fifth Int. Conf. Fuzzy Syst. Knowl. Discov. 5 (2008) 236–240. doi:10.1109/FSKD.2008.658.
[65] P. Ozarkar, M. Patwardhan, Efficient Spam Classification by Appropriate Feature Selection, Glob. J. Comput. Sci. Technol. Softw. Data Eng. 13 (2013).
[66] G.A. Miller, WordNet: A Lexical Database for English, Commun. ACM. 38 (1995) 39–41. doi:10.1145/219717.219748.
[67] A. Gangemi, R. Navigli, P. Velardi, The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet BT - On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sic, in: R. Meersman, Z. Tari, D.C. Schmidt (Eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2003: pp. 820–838. doi:10.1007/978-3-540-39964-3_52.
[68] T.S. Guzella, W.M. Caminhas, A review of machine learning approaches to Spam filtering, Expert Syst. Appl. 36 (2009) 10206–10222. doi:10.1016/j.eswa.2009.02.037.
[69] P. Resnick, RFC 2822 Internet Message Format, (2001) 1–57. https://www.ietf.org/rfc/rfc2822.txt (accessed November 7, 2016).
[70] Apache SpamAssassin Project, SpamAssassin Public Mail Corpus, (2005). https://spamassassin.apache.org/publiccorpus/ (accessed November 7, 2006).
[71] CSMINING Group, Spam email dataset, (2010). http://csmining.org/index.php/spam-email-datasets-.html (accessed November 7, 2006).
[72] B. Guenter, SPAM Archive, (1997). http://untroubled.org/spam/ (accessed November 7, 2016).
[73] V. Metsis, I. Androutsopoulos, G. Paliouras, Spam Filtering with Naive Bayes - Which Naive Bayes?, CEAS 2006 - Third Conf. Email Anti-Spam. (2006). https://classes.soe.ucsc.edu/cmps290c/Spring12/lect/14/CEAS2006_corrected-naiveBayesSpam.pdf.
[74] S.J. Delany, P. Cunningham, A. Tsymbal, L. Coyle, A case-based technique for tracking concept drift in spam filtering, Knowledge-Based Syst. 18 (2005) 187–195. doi:10.1016/j.knosys.2004.10.002.
[75] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (2014) 1–37. doi:10.1145/2523813.
[76] P. Domingos, M. Pazzani, On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, Mach. Learn. 29 (1997) 103–130. doi:10.1023/A:1007413511361.
[77] R Core Team, R: A Language and Environment for Statistical Computing, (2014). http://www.r-project.org/.
[78] B.E. Boser, I.M. Guyon, V.N. Vapnik, A Training Algorithm for Optimal Margin Classifiers, in: Proc. Fifth Annu. Work. Comput. Learn. Theory, ACM, New York, NY, USA, 1992: pp. 144–152. doi:10.1145/130385.130401.
[79] C. Cortes, V. Vapnik, Support-Vector Networks, Mach. Learn. 20 (1995) 273–297. doi:10.1023/A:1022627411411.
[80] C. Hsu, C. Chang, C. Lin, A practical guide to support vector classification, (2010).
[81] M. Enea, Fitting Linear Models and Generalized Linear Models with large data sets in

R, in: Stat. Methods Anal. Large Data-Sets, Chieti, Pescara, 2009: pp. 411–414.
[82] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
[83] K. Hornik, C. Buchta, A. Zeileis, Open-source Machine Learning: R Meets Weka, Comput. Stat. 24 (2009) 225–232. doi:10.1007/s00180-008-0119-7.
[84] L. Breiman, Bagging Predictors, Mach. Learn. 24 (1996) 123–140. doi:10.1023/A:1018054314350.
[85] S.J. Delany, P. Cunningham, A. Tsymbal, A comparison of ensemble and case-base maintenance techniques for handling concept drift in spam filtering, LAIRS Conf. (2006) 340–345.
[86] B.U. Gaikwad, P.P. Halkarnikar, Random Forest Technique for E-mail Classification, Int. J. Sci. Eng. Res. 5 (2014) 145–152.
[87] X. Carreras, L. Marquez, Boosting Trees for Anti-Spam Email Filtering, in: Proc. 4th Conf. Recent Adv. Nat. Lang. Process., 2001: pp. 58–64. http://arxiv.org/abs/cs/0109015.
[88] A. Liaw, M. Wiener, Classification and Regression by randomForest, R News. 2 (2002) 18–22. http://cran.r-project.org/doc/Rnews/.
[89] Z. Pawlak, Rough sets, Int. J. Comput. {&} Inf. Sci. 11 (1982) 341–356. doi:10.1007/BF01001956.
[90] L.S. Riza, A. Janusz, Rough Sets: Data Analysis Using Rough Set and Fuzzy Rough Set Theories, R News. (2015).
[91] R.A. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
[92] L. Zhang, J. Zhu, T. Yao, An Evaluation of Statistical Spam Filtering Techniques, 3 (2004) 243–269. doi:10.1145/1039621.1039625.
[93] R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in: Proc. 14th Int. Jt. Conf. Artif. Intell. - Vol. 2, Morgan Kaufmann Publishers Inc., Montreal, Quebec, Canada, 1995: pp. 1137–1143.
[94] L. Pitt, R.E. Reinke, Criteria for Polynomial-Time (Conceptual) Clustering, Mach. Learn. 2 (1988) 371–396. doi:10.1023/A:1022825229661.