Machine Learning Text Classification Algorithms

Helda CURMA
Institute of Statistics
Albania
hmitre@instat.gov.al

Valentina SINAJ
Department of Statistics and Applied Informatics
Faculty of Economy, University of Tirana
Albania
sinajv@yahoo.com

Malvina XHABAFTI
Department of Statistics and Applied Informatics
Faculty of Economy, University of Tirana
Albania
xhabaftimalvina1@gmail.com

***Abstract*— Technological changes are rapidly affecting the present and future prospects. This fundamental and very rapid transformation brings new opportunities and challenges. Technological developments, methodological advances, numerous data generated by digitization processes, have increased the need for automation of statistical processes using "machine learning" techniques. Automatic text classification and coding is one of the challenges faced in statistical information processing. In order to achieve high quality text classification, machine learning techniques as well as natural language processing are needed. This paper will be focused on different machine learning models used for text classification. The models will be classified by the input used to train the algorithm, such as supervised, semi supervised and unsupervised. Frequently used algorithms in different machine learning techniques for text classification will be explored from a theoretical perspective, taking into account the fact that new learning algorithms emerge day by day. Advantages and disadvantages of these learning algorithms will be discussed. The review made on this paper will explain the relations between different text classification algorithms and future research in this area.**

***Keywords*— Text classification; Algorithms; Machine learning;**

I. Introduction

The digitalization processes have provided us a vast amount of data, which if used in appropriate way can serve as an important source of knowledge. This information must be structured and one of the biggest challenges is structuring unstructured text. This doesn't mean that the text has no structure, as a matter of fact it is quite complex, but the structure and organization of data makes a big difference when it comes to effective automatic processing in a computer. The text classification is the process of categorizing into predefined groups, raw texts. In other words, it is the phenomenon of labelling the unstructured texts with appropriate tags that are predicted from a set of predefined categories. Instead of humans having to read and evaluate huge volumes of text in order to understand the context, text classification helps derive relevant understanding. Figure 1 shows the process of text classification based on machine learning algorithms to train the classifier.

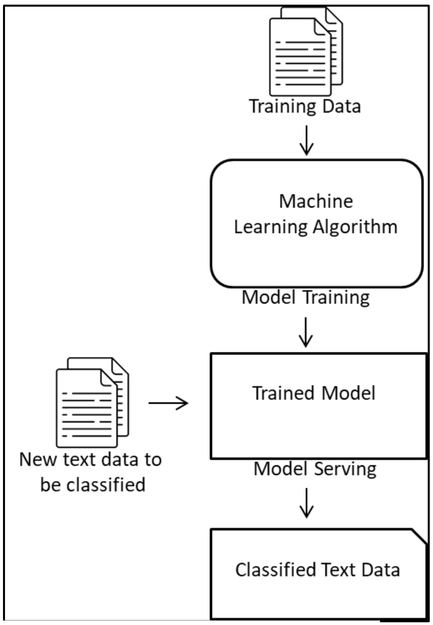

Figure 1: Text Classification process

Text classification needs have increased over the years and Machine Learning has been helping to exploit the full potential of this kind of information. Text classification is an important problem in many areas and text classifiers are used to quickly and cost-effectively classify different type of content. It is being used in document indexing, spam filtering, email categorization, social media sentiment analysis, survey coding, etc.

According to [1], text classification has been gaining power due to developments in the fields of text mining and natural language processing (NLP) and some practical applications go beyond simple task of categorization/classification into summarization and evaluation of open answers to specific questions. Natural language processing is a part of text mining. It performs analysis of the text based on the linguistic knowledge which is fundamental to help a machine “read and understand” text. Over the years statistical methods and machine learning have replaced traditional methods which were based in simple decision rules. The idea behind is that machine learning algorithms based on statistical methods understand linguistic patterns and utilize these patterns to make useful predictions and insights according to [2]. Machine learning algorithms are used extensively nowadays. The text classifiers in this learning process are not black boxes anymore, but can be theoretically explained and understand, instead of approaches where classification is done on ad-hoc basis and predictions quality is limited.

Over the years several machine learning algorithms have proven to be appropriate for text classification problems, including k-Nearest Neighbor, Naïve Bayes classifiers, Decision Trees, Rule Based Classifiers, Support Vector Machine, Hierarchical and Partitional Clustering, Self-Organizing Maps, Self-training, Co-training, Graph based methods, etc.

Machine learning problems can be classified by the input used to train the algorithm as shown in Figure 2. The main models are the following: supervised, semi supervised and unsupervised. This classification is done based on the nature of training data. This paper presents the main text classification algorithms based on these three models.

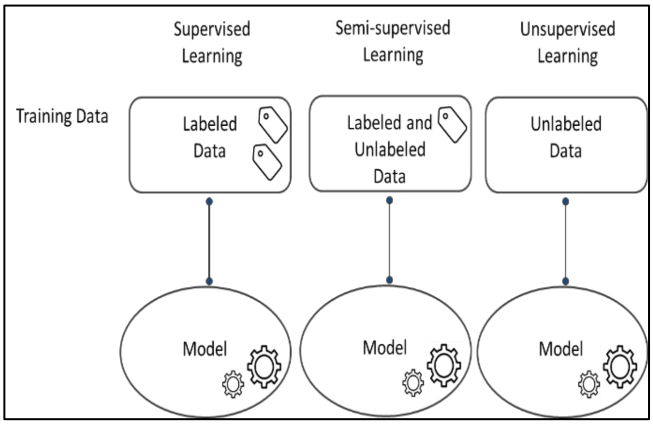

Figure 2: Supervised vs. Semi-supervised vs. Unsupervised Machine learning

II. TEXT CLASSIFICATION ALGORITHMS IN SUPERVISED LEARNING

First, Supervised learning is basically a synonym for classification. The supervision in the learning comes from the labelled examples in the training data set according to [3]. In supervised learning the classifier is learned from the training data to predict unknown data.

Text classification algorithms in supervised learning use the training data, where each text is labelled, to learn a classifier which classifies new texts. Weight vectors are used in order that the classifier can categories new text. These weight vectors are constructed by the training algorithm.

A. K-Nearest Neighbour classifier

K-Nearest Neighbor (KNN) is a very popular non parametric classification algorithm. In the context of text classification this algorithm classifies text based on the k nearest neighbor vote. According to [4] the ”k-Nearest Neighbor” (k-NN) algorithm represents a classification method, in which a new object is labelled based on its closest k neighboring objects. The “distance” between the training dataset and the new object is calculated and the k nearest objects are then chosen.

This is a non-parametric, very simple, frequently used algorithm. It is considered a “lazy” algorithm as the number of computations increment rapidly with the size of the training dataset. Noise on the training data can impact severely the accuracy of this algorithm. That is why improvements of this algorithm have been proposed over the years, to lower the computational burden.

B. Bayesian classifiers/Naive Bayes classifiers

Naive Bayes classifier is a probabilistic model used for classification. It is a parametric algorithm which is based on the Bayes theorem. It is based on the supposition of independence of attributes that is why it is called Naïve. It classifies an object based on the estimation of probabilities for each attribute and classes in order that the probabilities of the classified class are maximal. The (naive) Bayesian classifier is not impacted by isolated noise in the training data and irrelevant attributes. The drawback of this method is the assumption of independence of all attributes, which may be overcome by the usage of the Bayesian network.

C. Decision Trees

Decision Tree is a learning technique that can be used for classification problems. It is a classifier structured as a tree (based on true false queries), where internal nodes represent attributes, branches represent decision rules and leaf’s represents the text classification. Decision trees are simple and easy to be interpreted. Pruning is a significant theory used to reduce classification complexity by eliminating less meaningful, or irrelevant data, and finally to prevent overfitting and to improve the classification accuracy according to [5].

D. Rule Based Classification

A rule based classification is classification that based on a condition makes a prediction. The condition is based on attributes of a class and the result of it is the class definition or another condition testing. In the case of text categorization, the rules are based on the syntactic models of the lexicon to avoid errors in the classification. Rule learning algorithms, have become a successful strategy for classifier induction. Rule based classifiers provide the desirable property of being interpretable and, thus, easily modifiable based on the user's a priori knowledge according to [6]. Relatively slow learning method due to the dependency among nested rules.

E. Support Vector Machines

Support Vector Machine (SVM) is a non-parametric algorithm based on statistical learning. According to [7] it uses linear models to implement nonlinear category boundaries by transforming a given instance space into a linearly separable one through nonlinear mappings. SVM constructs a separating hyperplane (based on so called support vectors) to maximize the distance between the training samples of two categories (positive and negative). The distance between the two tangential planes is the margin of the classifier, which is to be maximized. Two separate training sets are needed in the case of SVM, one for positive and one for negative category. SVM is well performing in high dimensional spaces and when the number of dimensions is greater than the number of samples. Training time increases rapidly in the case of large data sets. Noise is also a factor of underperformance. Alternatively, to the linear classifier, nonlinear ones can be created by using a kernel function. Over the years different kernel functions have been proposed and used for text classification as well.

III. Text classification algorithms in unsupervised learning (clustering)

Before Unsupervised learning is machine learning type where there is no supervisor to guide the model. Models themselves, based on the data in their disposal, find the hidden inferences. This technique does not use a training data set or labelled data. So, in the case of text classification the classifier clusters the text without prior knowledge. In unsupervised learning underlying text patterns are revealed, and the text is clustered based on these likenesses. Two different approaches are used: hierarchical clustering, and non-hierarchical or partitional clustering.

In the case of hierarchical clustering a dendrogram is created, where clusters are made by using previously established clusters according to [4]. While in the case of non-hierarchical clustering (partitional clustering), the objects belong to only one cluster, so objects are classified in non-overlapping clusters.

A. Hierarchical Clustering

In hierarchical clustering, clusters are represented hierarchically and at varying levels of granularity according to [8]. The way the hierarchy tree is build divide this technique in two main methods. The bottom up approach which is also called the agglomerative clustering and the top-down approach called divisive clustering. The agglomerative approach starts with clusters having individual data points and merging these clusters to create the tree. The way clusters are merged may be different and based on it algorithms are divided in single-linkage, complete-linkage, average-linkage. In the case of single-linkage (nearest neighbor) the distance between two clusters is the distance between the two more similar texts. The complete-linkage (furthest neighbor) takes into account the distance between the two least similar texts, while the average-linkage the average of similarities of pairs of texts from every cluster according to [4] (unweighted pair-group average)/(weighted pair-group average). The divisive approach starts with one big cluster having all data points which is then splitted in smaller clusters to create the tree. The clusters with the biggest distance between texts are splitted recursively, until a certain criterion is met. In this case there is a trade of between the balancing of the tree, as well as the weight of every node.

Hierarchical clustering techniques are understandable and simple, but are very sensitive in terms of noise and outliers and in handling different sized clusters and large data sets.

B. Partitional Clustering

Partitional clustering techniques are totally different from hierarchical clustering. In these models the work starts by priorly defining a number k of clusters. This is why this is also called k-means clustering. The problem statement for partitional clustering is the definition of k clusters, which are as distinct as possible from each other. Even though the algorithms for defining k mutually exclusive clusters are in place based on creation of centroids for each cluster, there is an open problem on the definition of the best a priori number k. Different methods can be used to solve this issue such as cross validation or establishment of a criteria based on which results given from the use of different k numbers are compared. Text is moved between clusters to achieve clusters where variability of texts of the same is minimal, while variability of texts of different clusters is maximized.

Partitional clustering techniques create more symmetric clusters in terms of shapes, but are sensitive to the presence of outliers in the dataset and there is the problem of defining an appropriate k value.

C. Self Organizing Maps

Self-Organizing Maps (SOMs) are using clustering algorithms and can be used to accomplish classification. They use a type of neural network called Kohonen's self-organizing feature map. According to [9], the SOM learning can be explained as a map of nodes, each including a model vector, where learning is an iterative process based on each input of data objects. The closest map node is calculated for the data object and the model vector of the best matching unit is adjusted towards the input vector. The training is performed in iterations and is performed on the training data set. A class label is specified for every node of the map based on the number of training samples of different classes mapped on that node. After it the algorithm has the duty to map objects on the map surface. SOM are simple to understand especially due to their visual implementation, but more complex in terms of computations. For text classification also hybrid implementation of this algorithm with other ones are recommended.

IV. Text classification algorithms in semi-supervised learning

Semi-supervised learning, as the name implies, is a hybrid technique between supervised and unsupervised learning. Conceptually situated between supervised and unsupervised learning, it permits harnessing the large amounts of unlabeled data available in many use cases in combination with typically smaller sets of labelled data according to [10]. These machine learning techniques makes use of both labelled and unlabeled data. The main idea of semi supervision algorithms is to treat data objects based on the fact that they have or not labels. The labelled ones will use traditional supervision while for the unlabeled data, the difference in predictions between the data object with similar training examples is taken into account during the classification process. A supervised learning algorithm performs better in the case when the training data set is labelled, but semi-supervised algorithms are very powerful when labels are limited and unlabeled set of data is large.

A. *Self-training*

One of the most used and simple to understand semi-supervised learning techniques is self-training. As the name implies in this case the training is done by taking into consideration the same model prediction. The prediction that the model itself does on unlabeled data (pseudo labels) as well as the labelled data are included in the training process in an iterative manner until there are no confident predictions on the unlabeled data or a constant number of iterations is reached. Self-training are limited models in terms of performance. It has rather a good performance on small dataset or in the case when it is used in combination with other techniques in the bigger data sets. While the biggest down side of self-training is that the model cannot correct its mistakes. Once an unlabeled data is mistakenly labelled with a high level of confidence it will be included in the training process and its effect will escalate in all predictions. In case labelled and unlabeled data belong to different domains the model performance will be very reduced and predictions done will be incorrect.

B. *Co-training*

Co-training is a semi-supervised learning technique where classifiers are trained on two or multiple different views of the data. Pseudo labels of unlabelled instances are exchanged in an iterative way according to [12]. Co-training implies that the two different views are independent and sets with labels from each domain are appropriate to train the model. The initial models are trained based on their respective view sets and pseudo labelled data that have a certain level of confidence on one of the two models are used in the training set of the other model. This is an iterative process from which one model provides labels where the other model is uncertain. Even in the case of co-training when mistakes occur in wrongly labelling data mistakes are not corrected, so this data is not removed during the training process. The co-training process can be used also in more than two different views of the data. Co-training performs well on smaller data sets for text classification and it is appropriate to different classifiers, but is strongly reliant on the independence of the features of the data set.

C. *Graph based*

Graph based methods in semi-supervised learning as in accordance with [12], start with constructing a graph where the nodes represent all the samples and the weighted edges reflect the similarity between a pair of nodes. Nodes of the graph connected by edge and associated with large weights are likely to have the same classification label. So nearby samples on a low-dimensional manifold are assumed to share similar labels.

Graph based semi-supervised learning due to the graph structure can be easily understandable. Moreover, these algorithms can be used for binary classification but also multiclass scenarios. The downside of these methods is that they require heavy computations, while modified versions improve overall performance.

D. *Expectation-Maximization based*

Another technique for training a classifier based on labelled and unlabeled data is by estimating parameters of a generative model through iterative Expectation-Maximization (EM) techniques a. In the expectation according to [13]. Maximization algorithms, the maximum probability is estimated based on the presence of hidden variable. The probability for a certain value of hidden variables is calculated, and based on it the model is optimized, and this is done in iterations until there is no significant progress. The basic expectation and maximization steps of these algorithms are considered simple to be implemented for different machine learning problems and iterations will improve likelihood. In terms of convergence, it is slow and only related to local optimum (global optimum is not guaranteed).

V. Hybrid methods

Hybrid machine learning algorithms can be used to improve accuracy, lower processing time and need for heavy calculations in the case of text classification. These hybrid algorithms can be created combining in a hierarchic manner high performance classifiers as a two or more stages algorithm or by combining key principles form more than one basic machine learning algorithm. A machine learning base classifier can be combined as well with a rule-based one, in order to improve quality for the classification results. Different hybrid algorithms for text classification use: Naïve Bayes classifier and Support Vector Machine; Genetic Algorithm and Support Vectors Machine; KNN and SVM; KNN and Random Forest Tree, etc.

VI. Conclusions

In this review paper different machine learning algorithms have been explored based on input used to train the algorithm. The three different methods: supervised, unsupervised and semi-supervised learning have been taken into consideration to categorize algorithms that can be used to perform text classification. The different algorithms described have been extensively studied over the years. Existing literature supports their application for text classification problems achieving good performance. Important while choosing an appropriate algorithm for text classification is the actual text data to be classified as well as algorithm strengths and weakness. There is no one-size-fits-all solution. Further research need to be done in creation of more generic systems, which take into account the input data, to be used for text classification while retaining accuracy, reliability and overall quality.

References

[1] Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A Survey on Text Classification Algorithms: From Text to Predictions. Information, 13(2), 83.

[2] Zhang, Y., & Teng, Z. (2021). Natural language processing: a machine learning perspective. Cambridge University Press.

[3] Han J, P. J. (2011). Data mining: concepts and techniques. Amsterdam: Elsevier.

[4] Gorunescu, F. (2011). Data Mining: Concepts, models and techniques (Vol. 12). Springer Science & Business Media.

[5] Ying, X. (2019, February). An overview of overfitting and its solutions. In Journal of physics: Conference series (Vol. 1168, No. 2, p. 022022). IOP Publishing.

[6] Du, Maghesh. (2010). Automatic Induction of Rule Based Text Categorization. International Journal of Computer Science & Information Technology. 2. 10.5121/ijcsit.2010.2615.

[7] Berry, M. W., & Kogan, J. (Eds.). (2010). Text mining: applications and theory. John Wiley & Sons.

[8] Aggarwal, Charu & Reddy, Chandan. (2013). DATA CLUSTERING Algorithms and Applications.

[9] Saarikoski, J. (2014). On text document classification and retrieval using self-organising maps.

[10] Engelen, Jesper & Hoos, Holger. (2020). A survey on semi-supervised learning. Machine Learning. 109. 10.1007/s10994-019-05855-6.

[11] Ma, F., Meng, D., Dong, X., & Yang, Y. (2020). Self-paced multi-view co-training. Journal of Machine Learning Research.

[12] Song, Zixing & Yang, Xiangli & Xu, Zenglin & King, Irwin. (2021). Graph-based Semi-supervised Learning: A Comprehensive Review.

[13] Nigam, K., McCallum, A., & Mitchell, T. M. (2006). Semi-Supervised Text Classification Using EM.