# TEXT MINING APPLICATIONS TO FACILITATE ECONOMIC AND FOOD SAFETY LAW ENFORCEMENT

Gustavo Magalhães[1], Brigida Monica Faria[1,2], Luis Paulo Reis[2,3] and Henrique Lopes Cardoso[2,3]
[1]*Escola Superior de Saúde – Instituto Politécnico do Porto (ESS-P.Porto), Porto, Portugal*
[2]*Laboratory of Artificial Intelligence and Computer Science (LIACC), University of Porto, Porto, Portugal*
[3]*Faculty of Engineering, University of Porto (FEUP), Porto, Portugal*

**ABSTRACT**

Economic and Food Safety Authority receives on a daily basis reports and complaints regarding infractions, delicts and possible food and economic crimes. These reports and complaints can be in different forms, such as e-mails, online forms, letters, phone calls and complaint books present in every establishment. This paper aims to apply text mining and classification algorithms to textual data extracted from these reports and complains in order to help identify if the responsible entity to analyze the content is, in fact, the Economic and Food Safety Authority. The paper describes text preprocessing and feature extraction procedures applied to Portuguese text data. Supervised multi-class classification methods such as Naïve Bayes and Support Vector Machine Classifiers are employed in the task. We show that a non-semantical text mining approach can achieve good results, scoring around 70% of accuracy.

**KEYWORDS**

Text Mining, Economic and Food Safety, Natural Language Processing, Text Classification, Multi-class Classification

## 1. INTRODUCTION

Natural Language Processing (NLP) is a discipline between Computer Science and Linguistics (Kurdi, 2016). The extraction of features and patterns present on text can be of great value to data analysis and prediction. Natural language understanding systems allow the conversion of human text samples into a representation that a computer can read and understand (Thanaki, 2017). Therefore, text mining is of paramount importance to NLP research as it is defined as the process of extracting explicit knowledge from unstructured data, i.e., text (Jo, 2019). Text mining encompasses processes such as information retrieval, text classification and clustering, in turn allowing for better natural language perceiving and knowledge extraction (Kao and Poteet, 2007). Economic and Food Safety Authority is Portugal's national criminal police body responsible for supervising and preventing criminal delicts in the food and other economic sectors (*Missão, Visão e Valores*, 2019). Several reports and complaints are directed to Economic and Food Safety Authority regarding infractions, delicts and possible food and economic crimes. These reports and complaints can be submitted in several ways, such as via e-mails, online forms, letters, phone calls or complaint books present in every establishment. The main objective of this paper is to classify entries of text related to reports and complaints, present in Economic and Food Safety Authority's databases, concerning their respective competent institution. This is accomplished using data mining techniques (more specifically, text mining) such as text processing and supervised machine learning. Using text mining and machine learning techniques can help immensely Economic and Food Safety Authority's supervision and prevention mission by speeding up reports and complaints follow-up processes. In turn, this allows for faster response mechanisms and better law enforcing. This classification process is, at the time of writing, completely human-based.

## 2. RELATED WORK

Text mining is an area with several applications and generally refers to the process of extracting information from text. This process involves several phases such as structuring the input text like addition of linguistic features, removal of words, stemming and conversion to lower case letters, deriving patterns within the

structured data and finally the evaluation and interpretation of the results (Feinerer, Hornik and Meyer, 2008).Text classification, text clustering, ontology and taxonomy creation, document summarization and latent corpus analysis are standard techniques in this area. Classical applications in text mining come from the data mining such as document clustering and document classification (Faria, Pimenta and Moreira, 2009).

Chaovalit and Zhou (Chaovalit and Zhou, 2005) establishe comparisons between machine learning approaches (corpus based) to semantic orientation. Machine learning approaches are largely dependent on preprocessing and feature selection and wield more accurate results. Semantic analysis, while more resource friendly and efficient, isn't as accurate and can be skewed due to how human speech is constructed and interpreted (Chaovalit and Zhou, 2005). Joachims (Joachims, 1998) uses Support Vector Machines (SVM) for text categorization. Stemming words is suggested, as well as reducing the number of features from the text by computing the number of times these words occur in any given document and excluding those that are considered stop-words. Inversed document frequency (IDF) is also suggested to improve performance. The paper also establishes comparisons between SVMs and other classifiers such as Naïve Bayes and K-Nearest Neighbors (k-NN) and, concludes that SVMs are well suited for text categorization. A study conducted at the School of Economic and Business, Telkom University (Arusada, Putri and Alamsyah, 2017), investigates how training data optimization can improve a multiclass text classification problem. The case study uses human-defined set of categories (supervised learning) and tries to fit sentences to their respective category. The preprocessing steps applied to the text are tokenization (separating words), stop-word removal, stemming and inverse document frequency weighting. The training data is separated into three conditions: each complaint is directly labeled in its category; a label is given for each sentence manually (built upon its keyword) and a sentence with more than one complaint will be separated into suitable categories for each. Both Naïve Bayes and the SVM classifier were observed to shown similar results across the three categories although a bigger discrepancy is shown when using the first condition (each complaint is directly labeled).

# 3. METHODOLOGY

This chapter presents the main phases following the Knowledge Discovery in Databases (KDD) methodology. KDD methodology divides in 9 phases from business understanding (application domain), data set selection, data cleaning and preprocessing, feature selection, model creation, model selection, data mining (pattern recognition), results interpretation and applying the knowledge acquired to the business model (Fayyad, Piatetsky-Shapiro and Smyth, 1996). Knowledge discovery should be efficient and able to extract valuable information from the data. It should also be efficient in space and in time to coupe with large data sets (Feldman and Dagan, 1995). Each of the KDD phases are described below along with their respective procedures.

## 3.1 Data Preparation and Text Preprocessing

The dataset used totaled 120193 entries ranging from the years 2014 to 2018, each having 49 attributes. While there are many attributes in the dataset, most are directly or indirectly related to the content of the complaint or report. The targeted attribute of this article is the "competence" as it is one of the first steps in automating the complaints and reports process. This attribute is divided into 7 categories: ASAE, ASAE and courts, ASAE and another Entity, another Entity, another Entity and Courts, and Courts. From these entries, 5000 samples were selected for further analysis and classification due to computational requirements. Of these, train and test samples were selected with the test size being 30% of the train size.

Preprocessing encompasses several steps such as tokenization, standardization, cleansing of the text data, removal of stop words and stemming or lemmatization (Murugan, Hill and Nolan, 2019). The objective of text processing is returning cleansed tokens from raw text. Tokens are single words or groups of words that can be fed to machine learning algorithms to be able to understand how the data is structured and how it can help predict attributes related to it (Murugan, Hill and Nolan, 2019). As a first step, the text data was tokenized using the NLTK (Natural Language Toolkit) package. NLTK allows the building of Python programs to work with human language data (Bird, Klein and Loper, 2009). This package uses regular expressions (context-independent syntax that represent a variety of character sets that define a search pattern) to tokenize the text (IEEE, 2016). This package also splits standard contractions, treats most punctuation characters as separate tokens, splits of commas and single quotes when these are followed by whitespaces and separates periods that appear at the end of line (*nltk.tokenize.treebank — NLTK 3.4.1 documentation*, no date). The tokens retrieved were evaluated against a set of Portuguese stop words and removed if present. The same was done

regarding punctuation and special characters as well as any single character. Furthermore, stemming was applied on the tokens. Stemming is the process in which a word is reduced to its stem, base or root. This stem is shared between words from the same etymological family (Viera and Virgil, 2007). It works by cutting of the end or the beginning of the word, sometimes making the word unreadable or lose its context. Snowball stemmer was chosen due to its support for the Portuguese language, unlike its previous iteration Porter stemmer (Porter, 1980). The Snowball stemmer is faster and is an improvement over the original Porter stemmer (Porter, 2019).

One popular method of feature weighting is term frequency-inverse document frequency (Tf-idf). "Tf-idf" values terms proportionally to their frequency but reduces their weight based on the frequency they appear on the corpus (inverse document frequency). This is illustrated on equation 1 where "n" stands for the total number of documents in the document set and df(t) the document frequency of t (number of documents in the document collection that contain the term t).

$$idf(t) = log\frac{1+n}{1+df(t)} + 1 \quad \text{(Eq.1)}$$

Term frequency-inverse document frequency weighting was chosen due to its ability to inherently give less importance to terms that appear frequently in most documents and a higher weight to those that appear often in a small collection of documents (Jones, 1973). The vocabulary extracted was based on the length of the features extracted.

## 3.2 Multinomial Class Prediction

Several multi-class prediction algorithms were employed: Multinomial Naïve Bayes (NB), Support Vector Machines (SVM), Decision Trees (DT), Random Forest (RF), K-Nearest Neighbors (k-NN) and Logistic Regression (LR). The results were then compared. A grid search was conducted which featured options varying from 1-gram (unigrams) to 2-grams (bigrams) and 3-grams (trigrams) with N-gram being a sequence of "N" words. This was done to select the best n-gram range for each classifier. Table 1 shows these results.

Table 1. Best N-grams found for each classifier

| Classifier | N-grams |
|---|---|
| Linear Support Vector | Unigrams and Bigrams |
| Random Forest | Unigrams |
| Decision Tree | Unigrams |
| Logistic Regression | Unigrams and Bigrams |
| Multinomial Naïve Baye | Trigrams |
| K-Neighbors Classifier | Unigrams |

The selected algorithms were all implemented from the Scikit-learn packages and their default values were not altered to provide a baseline evaluation. To evaluate the models, several scoring measures were calculated: Accuracy, precision, recall and F1. Accuracy is a ratio of correctly predicted observations to the total observations. It is a simple scoring measure and usable in cases where there is similar ratio of false positives and false negatives. Precision measures how precise the model is according to how many of the positive samples are correctly labeled as positive. Recall represents the ratio of correctly predicted positive observations (True Positives) to the total positive observations. It should be used when there is a high False Negative cost. F1 Score is the weighted average of precision and recall. It takes in consideration False Negatives and False Positives and should be used in cases where there is uneven class distribution. Precision, recall and averaged F1 were all calculated with the class weight "macro" setting which takes label imbalance in consideration, as is our case. Central Processing Unit time (CPU time) stands for the amount of time that a central processing unit (CPU) was used since instructions were given. It is measured in clock ticks and is more accurate than the measurement of real time elapsed (Gnu, 2019).

# 4. RESULTS

Out of the six evaluated classifiers, the most accurate was the support vector machine with linear kernel (70.2%), followed by Logistic Regression Classifier (66.3%) and the Random Forest Classifier (60.5%). Regarding precision, the most precise model was the K-Neighbors Classifier with results of approximately 70% (69.9%), followed by the SVM with 61.9%, and closely followed by the RF with 58.2%.

The Recall scoring measure was observed to have similar results at the lower end of the spectrum between Multinomial NB and K-Neighbors Classifier (23.8% and 20.3% respectively). The SVM reached predictions of 50% (49.9) with the second best being the Decision Tree (39.2%) which had a slightly higher percentage than the LR with 38.2%. The weighted average between precision and recall (F1 score) calculated for the SVM was 53.1%, the best result of all the classifiers, followed by the RF classifier (40.5%) and DT (40.2%).
All the results outlined above can be seen in Table 2.

Table 2. Scoring for each model fitted.

| Model | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Linear Support Vector | 0.702 | 0.619787 | 0.499354 | 0.531114 |
| Random Forest | 0.604667 | 0.582377 | 0.362279 | 0.405056 |
| Decision Tree | 0.561333 | 0.415173 | 0.391984 | 0.401897 |
| Logistic Regression | 0.662667 | 0.44028 | 0.38222 | 0.396264 |
| Multinomial Naive Bayes | 0.538667 | 0.565695 | 0.237704 | 0.233666 |
| K-Neighbors Classifier | 0.484 | 0.699205 | 0.202965 | 0.175576 |

It was observed that some of the predicted labels had individual F1 scores of 0, which meant that a prediction was not established. This, in turn lowers the weighted F1 score. On one hand, the Support Vector Machine model was able to achieve high F1 scoring, approximately 53%, and high accuracy (70.2%). It was also observed that it had the highest scores for each measure but precision, achieving the second best result with approximately 62%. On the other hand, the SVM model was the second slowest model to train with 2.2 Central Processing Unit time while the fastest model trained was the Random Forest (0.82). The central processing unit time (CPU time) for each model trained was also calculated, as seen in Figure 1.

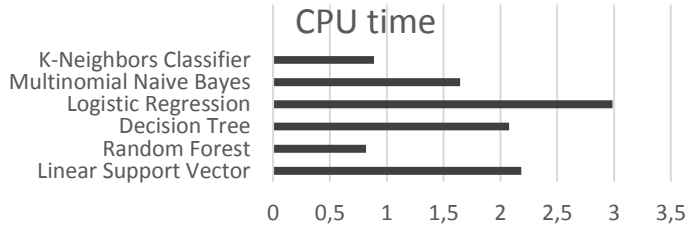


Figure 1. Central Processing Units Time (CPU Time) for each model fitted

Multinomial Naïve Bayes, although expected to be faster than the Random Forest Classifier, had mixed results regarding CPU time (1.64). Inconsistent results were also observed regarding prediction scoring with low recall and F1-score which can be explained by labels with no predictions. Although baseline results demonstrated high accuracy (~70%) regarding the Linear Support Vector model, F1 scoring was observed to be average (~53%). The LSVC was the second slowest model to train despite achieving the best results. However, CPU time is a measure that can be influenced by several factors external to the models and was only calculated to establish comparisons between the models and how each model fared regarding their computational performance.

# 5. CONCLUSION AND FUTURE WORK

This paper provides a comparison between established methods of text classification using supervised machine learning regarding their ability to predict categories in a multi-class example and their computational performance. There are many different algorithms that can be used to classify textual data, with varying degrees of success and speed. The algorithms chosen here were selected due to their established capabilities and lower processing requirements. It was concluded that the support vector machine classifier outperformed the others

by a significant margin regarding every measure. Since the macro setting was considered when calculating the F1 score, the metric was biased towards the least populated classes. In turn, this can mean that the smaller labels were misclassified. In addition, the SVM classifier was shown to have higher precision than recall which means that the classifier predicts many actual true positive labels as true positives (high precision) but fails to predict a lot of the relevant items from the total amount of relevant items (low recall). Even though the number of tools available to conduct text mining procedures in the Portuguese language is expanding, these are still lacking when comparing to the English language. This, in turn, makes it harder to conduct text mining procedures on Portuguese unstructured data and even more so when conducting lexical analysis due to the lack of Portuguese lemmatizers. Furthermore, grammatical errors are prone to increase the number of features of the data and, because of this, spelling correction applied on the input data can help reduce dimensionality and increase the classifiers' performance but at the cost of preprocessing run times. The process described in this article can be helpful to automate classification tasks. One such case is the attribution of "competence" to each complaint and report directed to Economic and Food Safety Authority. This application has a positive impact on the speed which complaints and reports are analyzed and processed, meaning faster responses, swifter inspections and law enforcing.

## ACKNOWLEDGEMENT

The authors would like to thank the IA.SAE – "Inteligência Artificial na Segurança Alimentar e Económica" project, funded by the FCT/MCTES through national funds (PIDDAC), as part of the "Programa Iniciativa Nacional Competências Digitais" e.2030 – INCoDe.2030, enrolled in the National Reform Plan.

## REFERENCES

Arusada, M. D. N., Putri, N. A. S. and Alamsyah, A., 2017. *Training data optimization strategy for multiclass text classification*, *5th International Conference on Information and Communication Technology, ICoIC7 2017*, 0(c). doi: 10.1109/ICoICT.2017.8074652.

Bird, S., Klein, E. and Loper, E., 2009. *Natural Language Processing with Python - Analyzing Text with the Natural Language Toolkit*.

Chaovalit, P. and Zhou, L., 2005. *Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches*, 00(C), pp. 112c-112c. doi: 10.1109/hicss.2005.445.

Faria, B. M., Pimenta, R. and Moreira, J., 2009. *Projects of Fisioterapia and Terapia Ocupacional: A Classification Approach using Text Mining in R*, in Rocha, Á. et al. (eds) *Sistemas e Tecnologias de Informação*. APPACDM. Póvoa de Varzim, Portugal: 4ª Conf. Ibérica de Sistemas e T. de Informação, Sistemas e Tecnologias de Informação, pp. 367–372.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P., 1996. *From Data Mining to Knowledge Discovery in Databases*, *Al Magazine*.

Feinerer, I., Hornik, K. and Meyer, D., 2008. *Text Mining Infrastructure in R*, Journal of Statistical Software, 25(5).

Feldman, R. and Dagan, I., 1995. *Knowledge Discovery in Textual Databases (KDT)*, The First International Conference on Knowledge Discovery and Data Mining (KDD), pp. 112–117. doi: 10.1.1.47.7462.

IEEE, 2016. *Standard for Inf. Technology — Portable Operating System Interface ( POSIX ® ) Technical*, *IEEE Xplore*.

Jo, T., 2019. *Text Mining Concepts, Imp., Big Data Challenge*, in Springer, pp. 3–6. doi: 10.1007/978-3-319-91815-0.

Joachims, T. (no date) 'Text Categorization with SVM: Learning with Many Relevant Features', pp. 2–7.

Jones, K. S. (1973) 'Index term weighting', *Inf. Storage and Retrieval*, 9(11), pp. 619–633. doi: 10.1016/0020-0271(73)90043-0.

Kao, A. and Poteet, S. R., 2007. *Natural language processing and text mining*. doi: 10.1007/978-1-84628-754-1.

Kurdi, M. Z., 2016. *Natural Language Processing and Computational Linguistics*.

*Missão, Visão e Valores*, 2019. Available at: https://www.asae.gov.pt/asae20/missao-visao-e-valores.aspx (Accessed: 29 April 2019).

Murugan, A., Hill, C. and Nolan, T., 2018. *Practical Text Analytics: Maximizing the Value of Text Data*.

*nltk.tokenize.treebank — NLTK 3.4.1 documentation*, 2019. Available at: https://www.nltk.org/_modules/nltk/tokenize/treebank.html#TreebankWordTokenizer (Accessed: 24 April 2019).

Porter, M. F., 1980. *An algorithm for suffix stripping*, Program. doi: 10.1108/eb046814.

Thanaki, J., 2017. *Python Natural Language Processing*, Packt Publishing Ltd, ISBN: 978-1-78712-142-3.

Viera, A. F. G. and Virgil, J., 2007. *Uma revisão dos algoritmos de radicalização em língua portuguesa*, Inf. Research.