CLER-News: an automatic news classifier

Christian S. C. Cardozo, Elaine C. Tady, Luiz Fernando R. Oliveira, Rafael M. Andrade

[1]COPPE – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

{christiancardozo,elainetady,lfdeoliveira,rafmandrade}@cos.ufrj.br

Abstract. *In this work, the goal is to classify technology news from two Brazilian news portals by analyzing the texts of the titles and short abstracts about each news. In order to do this classification, the TF/IDF and word2vec methods were used, combined with a set of traditional learning algorithms, which accuracies are measured and compared. Computational results shows that the TF/IDF representation provided quite similar validation performance compared to word2vec when combined with the XGBoost and MLP algorithms.*

1. Introduction

The Internet and today's tools provide ways to generate a huge daily amount of information and turns possible for news portals produce higher and higher amounts of content. It's the beginning of the content consumption problem: facing a large number of items disposed in a web page and with a short amount of time available, an user must select a small subset of news to get informed. This choice is usually done based on the degree of interest promoted to the user by the news title and a short description of it.

Meanwhile, the $technology$ area turns into a bigger field which subareas starts to grow on each new industrial advance. *Mobile devices* are a clear example of this change: until recently treated as the same subject, $applications$ and $hardware$ present themselves as two distinct subjects of interest. Indeed, it makes sense to split small improvements between two devices released from new tools for personal organization available on both Apple and Google's applications stores.

Considering these points, this works aims to automatically classify a given technology news into an specific sub area based on it's title and abstract. It is divided into five sections, of which this is the first. Section 2 briefly explore some related works present in the literature. In section 3, the methodology used is presented, while section 4 shows the experimental results obtained. Finally, section 5 presents conclusions and possible future works, followed by the bibliographic references used.

2. Related Subject

Many previous work was done in other researches throughout the area of news classification with many different approaches. The most common approaches, in general, are based on Information Retrieval techniques - such as the use of TF/IDF or word2vec - and Data Mining and Statistical techniques, such as the use of classification algorithms like Decision Trees, K-Means, Naive Bayes, Support Vector Machines and Neural Networks. However, no work was found regarding specifically the classification of technology news, as proposed in this work.

In the article by [Li and Jain 1998], news from *Yahoo!* were classified into seven categories, based on news previously classified in those same categories, using four different classification methods, where Naive Bayes and Decision Tree are among them; in the article by [Wang and Wen 2016], an approach based on *Part-of-speech Tagging* was implemented to filter the keywords of news and classify them into categories based on frequency of the selected keywords; in the article by [Rahmawati and Khodra 2016], Indonesian news were classified using *word2vec* and TF-IDF methods, and a comparison of those methods was performed.

3. Experimental Framework

3.1. Data Acquisition

In order to build the news dataset, two Brazilian news portals about technology were used: $Tecmundo$ (`www.tecmundo.com.br`) and $Computerworld$ (`www.computerworld.com.br`). Both portals does not provide an API that enable users to retrieve a collection of news. It was, then, developed a Web Crawler that collects and stores the title, abstract and category of news based on a given URL.

A total amount of 49,998 news distributed over 17 categories from the $Tecmundo$ portal and 16,653 news distributed over 15 categories from the $Computerworld$ portal were collected, as described in table 1.

Tecmundo		Computerworld	
Class	# news	Class	# news
Market	12304	Business	4989
Geek Culture	8082	Apps	2437
Mobile Devices	6792	Infrastructure	2124
Product	6029	Carreer	1507
Software	5985	Security	1506
Internet	3510	Management	1347
Social Networks	1954	Technologies	769
Science	1846	Rising Tech	568
Security	1783	Cloud-computing	372
Smart Cities	1680	Internet	342
Story Stream	12	Mobility	232
Events	9	IT in Practice	178
Galaxy	6	Big Data	154
Enterprises	2	Telecom	127
Moto	2	Security	1
PS Meeting 2016	1	-	-
BGS 2016	1	-	-

Table 1. All classes that appears in each dataset followed by the number of news that belongs to that class

Two procedures were performed on both news datasets with the objective to balance the classes. First of all, only the five most frequent classes of each dataset were chosen. In that way, the $Tecmundo$ dataset ended up with the classes $Market$, $Geekculture$,

$Mobile$, $Product$ and $Software$, while the $Computerworld$ dataset with the classes $Business$, $Apps$, $Infrastructure$, $Career$ and $Security$.

The second procedure aims to create a numeric balance between each dataset set of classes by randomly picking 5000 news of each class on the $Tecmundo$ dataset and 1500 news on the $Computerworld$ dataset, as summarized on table 2. Considering the two procedures, the final datasets are formed by 25,000 observations of news on $Tecmundo$ and 7,500 on $Computerworld$.

Tecmundo		Computerworld	
Class	# news	Class	# news
Market	5000	Business	1500
Geek Culture	5000	Apps	1500
Mobile Devices	5000	Infrastructure	1500
Product	5000	Carreer	1500
Software	5000	Security	1500

Table 2. Category frequency after class balance procedure

3.2. Preprocessing and Features Generation

Each dataset was preprocessed in order to prepare the data that will be used. For each news on the datasets, the texts of the title and abstract are merged to form a single string. The words in this string are, then, tokenized, followed by the remotion of the stop words. This step is performed by using the NLTK Portuguese stopwords list. The tokens were set to lowercase, the special characters were removed and characters with accentuation were replaced by the corresponding character without the accentuation. Finally, a stemmer is applied for each token using the RSLPStemmer from NLTK package.

At this point, each news is a preprocessed string. It is, then, applied a series of procedures in order to generate the features for each observation that will be presented to the learning algorithms. The TF/IDF algorithm was used to calculate the weights between the terms and documents. The values are calculated using dictionaries of trigrams, bigrams and unigrams of the words present in the texts. In the $Computerworld$ dataset, the TF/IDF matrix has 219,469 columns, while the $Tecmundo$ dataset provides a matrix with 661,515 columns. To reduce the dimensionality of the matrix, the SVD method was used with $k = 30$ on both datasets. Figure 1 and 2 shows, respectively, the eigenvalues curves of SVD from $Computerworld$ and $Tecmundo$ datasets. Both curves justify the choose of value 30 since it significantly reduces the TF/IDF matrices dimensionality without loss of the most relevant information on the datasets.

The Word2vec method, which is a multilayer neural network with an input layer, an output layer, and a hidden layer that learns word vectors through context, was also experimented. For this case, the stemmer procedure were not applied at tokens. The chosen architecture was the Skipgram, in which a word is used to predict the context [Asr et al. 2016]. The text of each category was transformed into an average vector of words. To represent each word was used a portuguese base of wikipedia with 300 dimensions and 50246 vocabulary size, and to represent a word that did not appear in the external base, an average vector of all words (outside the vocabulary) was used.

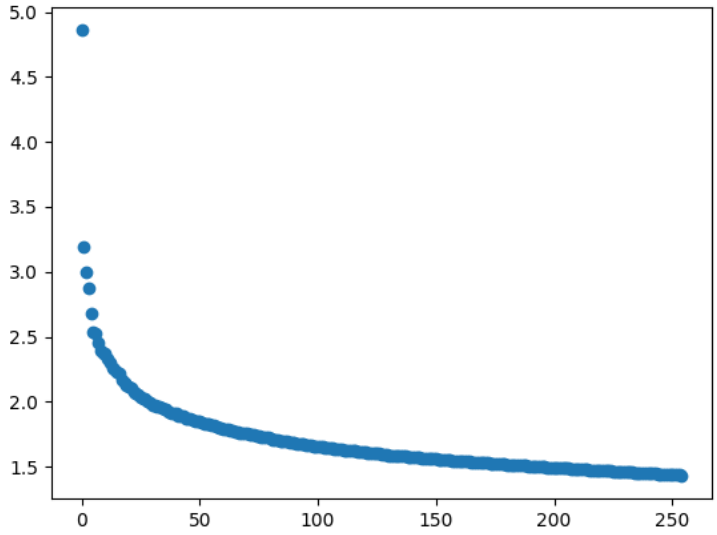

Figure 1. Firsts 255 eigenvalues from ComputerWorld dataset's SVD

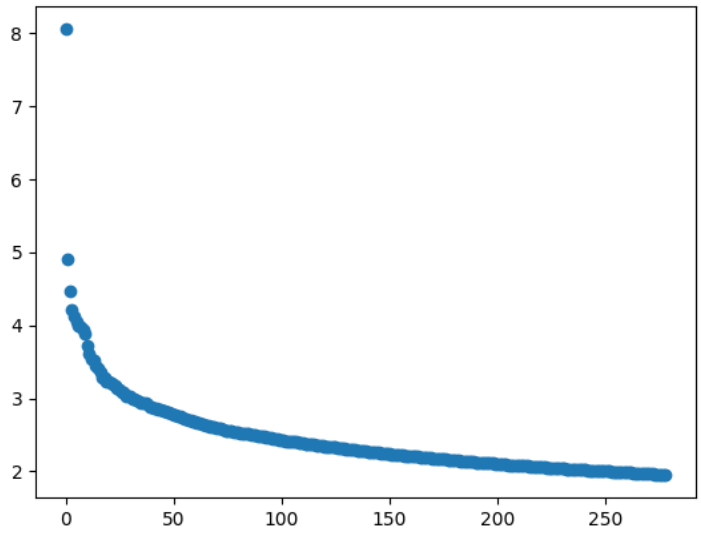

Figure 2. Firsts 280 eigenvalues from Tecmundo dataset's SVD

3.3. Learning Algorithms

3.3.1. Naive Bayes

Naive Bayes is a classification algorithm that is based on the Bayes' theorem, which applies the assumption that the model's features are independent. This model is used as baseline in this work and gives a minimum accuracy expectation about the problem. A Minmax scaler was applied on the features so that they could be used as input to

the Multinomial Naive Bayes algorithm from Scikit-learn. The default parameters from the package were used.

3.3.2. Decision Tree

This model consists in using a decision tree structure in order to make a conclusion about the data observed. It uses the value of the observation's features to navigate through the tree's branches until reaching a leaf node, which contains the chosen class for that observation. For this model, the Decision Tree Classifier from Scikit-learn were used with the default parameters.

3.3.3. Feedforward Neural Network

Neural networks are computational systems inspired on how the biological neurons work. It is based on a set of connections between artificial neurons organized in a layer-shape format where information is transmitted from a layer to another. A $feedforward$ network is so that the connections between layers do not perform a cycle. A set of parameters was tested looking for better results. Table 3 shows the tested parameters configurations for the MLP Classifier from Scikit-learn.

Configuration	# layers	# neurons per layer	Activation	Solver	Maximum iterations
0	1	100	relu	adam	500
1	2	100	relu	adam	500
2	1	500	relu	adam	500
3	2	500	relu	adam	500
4	1	1000	relu	adam	500
5	2	1000	relu	adam	500

Table 3. Parameters Configurations for MLP Classifier

3.3.4. XGBoost

According to the site documentation (`xgboost.readthedocs.io`), XGBoost "is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable". The XGBoost library provide a parallel implementation of the boosting algorithm which guarantees efficiency in machine learning tasks.

The XGB Classifier from XGBoost package was used and a set of combinations of parameter configurations was made. The empirical exploration was made varying three parameters: learning rate, maximum depth and number of estimators. The objective function parameter used was always $multi : softmax$. Considering the sets of values used for each parameter described on table 4, an amount of 48 different configurations was tested.

Parameter	Set of Configurations
Learning rate	{ 0.1, 0.15, 0.2 }
Maximum depth	{ 3, 5, 8, 10 }
Number of estimators	{100, 250, 500, 1000}

Table 4. Parameters Configurations for XGBoost

3.3.5. Computational Environment

In order to run the proposed experiments, it was used a PC with an Intel Core i7-2600K processor, 3.40GHz, 8GB of RAM memory and the Ubuntu 14.04 operating system. The algorithms were implemented using the Python 3.6 language, as well as the Scikit-learn, XGBoost and NLTK packages. The code and all data used on this work is available at the GitHub repository https://github.com/LuizFernando-RO/cler-bri17.

4. Results and Discussion

In this experiment, the goal is to verify how the classification models performs in terms of accuracy. The datasets were divided in 90% to training and 10% to validation. To train the models, the K-fold cross-validation was used with K value equal to 3. The K-fold procedure gives the average accuracy for the three steps and the validation test gives the accuracy in the 10% data prediction. Table 5 shows the results related to the $Tecmundo$ dataset using the TF/IDF and the table 7 using word2vec, while tables 6 and 8 shows the results related to the $Computerworld$ dataset.

Table 5 shows that the best accuracy results in K-fold and in the validation set were obtained using MLP to the $Tecmundo$ dataset.

	Average accuracy in K-fold (%)	Accuracy in validation set (%)
Naive Bayes	49.11	49.00
Decision Tree	45.85	45.97
MLP	61.47	64.44
XGBoost	59.92	60.68

Table 5. Classification results on Tecmundo using TF/IDF

Table 6 shows that the best accuracy results in K-fold and in the validation set were obtained using MLP to the $Computerworld$ dataset.

	Average accuracy in K-fold (%)	Accuracy in validation set (%)
Naive Bayes	62.01	62.93
Decision Tree	56.12	57.73
MLP	71.15	72.53
XGBoost	70.55	70.00

Table 6. Classification results on Computerworld using TF/IDF

Table 7 shows that the best accuracy results in K-fold and in the validation set were obtained using XGBoost to the $Tecmundo$ dataset.

	Average accuracy in K-fold (%)	Accuracy in validation set (%)
Naive Bayes	48.70	48.64
Decision Tree	34.74	34.76
MLP	54.64	56.20
XGBoost	55.34	56.76

Table 7. Classification results on Tecmundo using word2vec

Table 8 shows that the best accuracy results in K-fold and in the validation set were obtained using XGBoost to the $Computerworld$ dataset.

	Average accuracy in K-fold (%)	Accuracy in validation set (%)
Naive Bayes	58.22	62.53
Decision Tree	41.45	42.66
MLP	69.37	72.26
XGBoost	66.94	72.26

Table 8. Classification results on Computerworld using word2vec

Since MLP and XGBoost presented the best results among the experiments, the confusion matrix related to these two learning algorithms are shown bellow. Figures 3 and 4 show the confusion matrix related to the $Tecmundo$ dataset using TF/IDF, while figures 5 and 6 show the one related to the $Computerworld$ dataset using TF/IDF. Figures 7 and 8 show the confusion matrix related to the $Tecmundo$ dataset using word2vec, while figures 9 and 10 show the one related to the $Computerworld$ dataset using word2vec.

Figure 3 shows that, when using MLP with TF/IDF to predict $Tecmundo$ classes, the classes $Market$ and $Product$ were the ones with more misclassified elements among themselves. In particular, the class $Market$ was the one with more misclassified elements among all classes, with 46.53% of correct predictions.

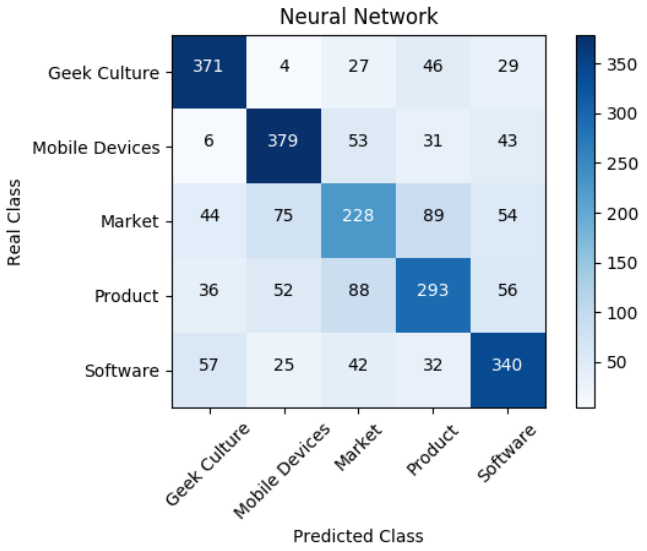

Figure 3. Confusion matrix of MLP in the Tecmundo dataset using TF/IDF

Figure 4 shows that, when using XGBoost with TF/IDF to predict $Tecmundo$ classes, the classes $Market$ and $Product$ were again the ones with more misclassified elements among themselves. In particular, the class $Market$ was again the most misclassified class among all, with only 40.4% of correct predictions.

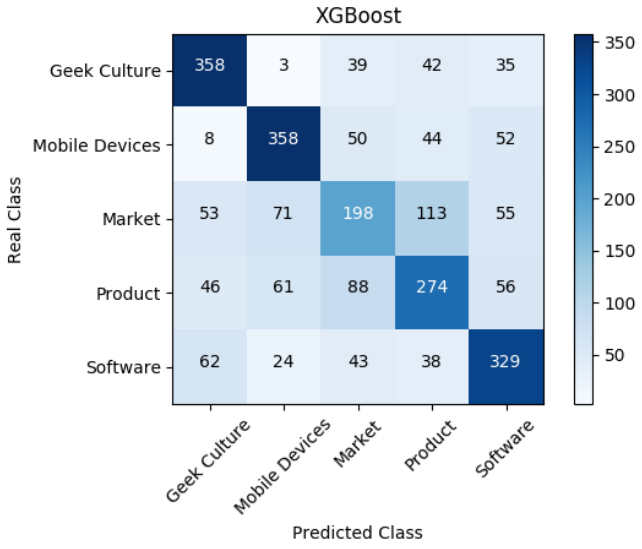

Figure 4. Confusion matrix of XGBoost in the Tecmundo dataset using TF/IDF

Figure 5 shows that, when using MLP with word2vec to predict $Computerworld$ classes, the classes $Apps$ and $Business$ were the ones with more misclassified elements among themselves. The class $Apps$ was the worse predicted class, with 54.34% of correct predictions.

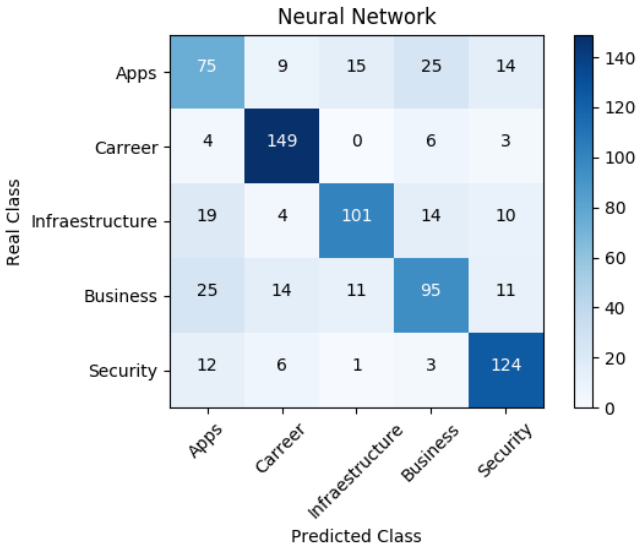

Figure 5. Confusion matrix of MLP in the ComputerWorld dataset using TF/IDF

Figure 6 shows that, when using XGBoost with word2vec to predict $Computerworld$ classes, the classes $Apps$ and $Business$ were again the ones with more misclassified elements among themselves. The class $Apps$ was again the worse predicted class, with 51.44% of correct predictions.

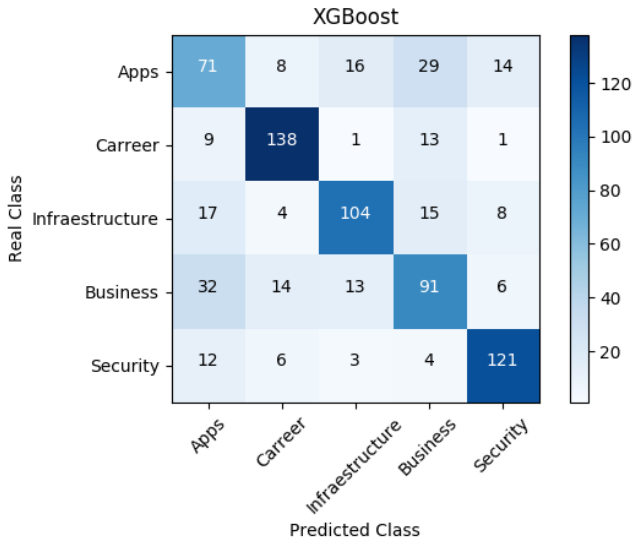

Figure 6. Confusion matrix of XGBoost in the ComputerWorld dataset using TF/IDF

Figure 7 shows the classification to $Tecmundo$ using a multilayer perceptron with word2vec to generate the features, and considering the real class and predicted class the highest number of hits were in $Geek\ Culture$ and $Mobile\ Devices$.

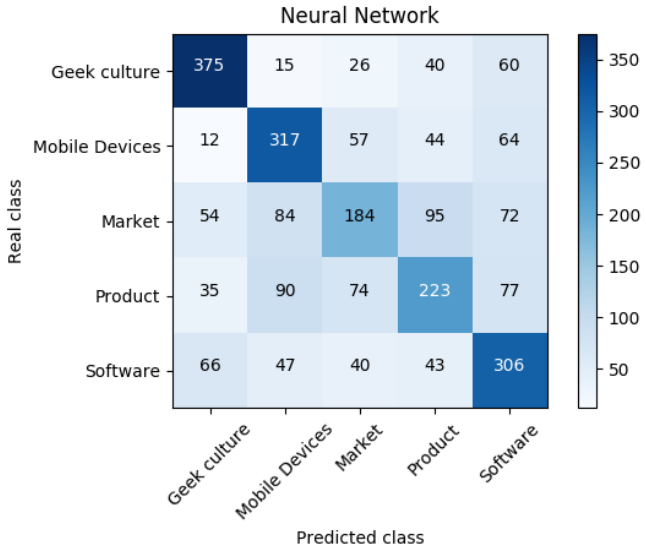

Figure 7. Confusion matrix of MLP in the Tecmundo dataset using word2vec

Figure 8 shows the classification to $Tecmundo$ using XGBoost with word2vec to generate the features, and considering the real class and predicted class the highest number of hits were in $Geek\ Culture$ and $Software$.

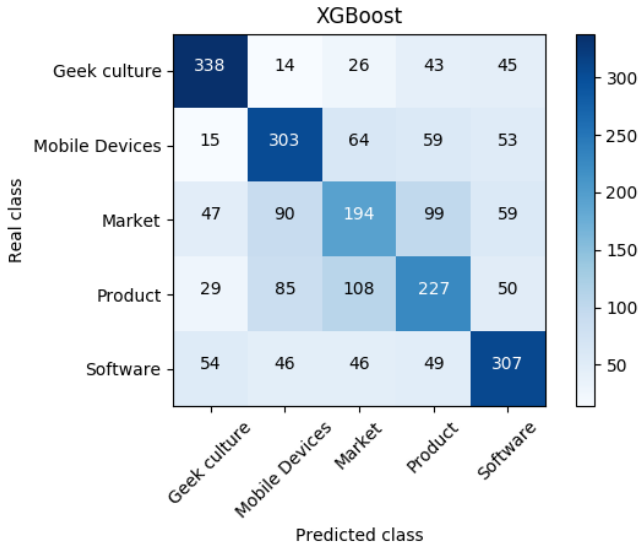

Figure 8. Confusion matrix of XGBoost in the Tecmundo dataset using word2vec

Figure 9 shows the classification to $Computerworld$ using a multilayer perceptron with word2vec to generate the features, and considering the real class and predicted class the highest number of hits were in $Carreer$ and $Security$.

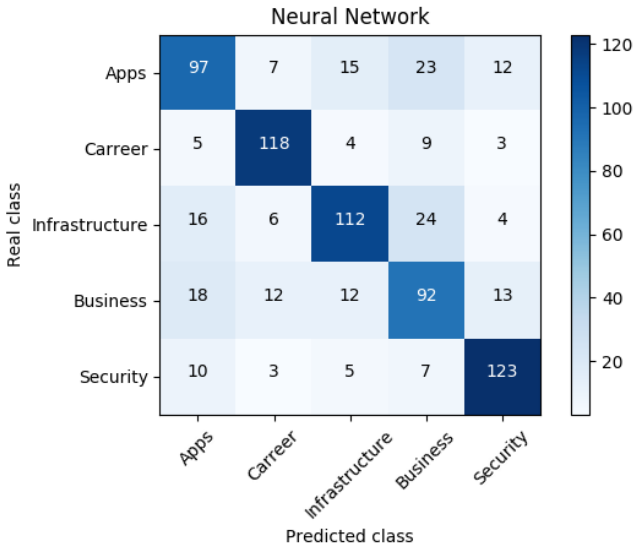

Figure 9. Confusion matrix of MLP in the ComputerWorld dataset using word2vec

Figure 10 shows the classification to *Computerworld* using XGBoost with word2vec to generate the features, and considering the real class and predicted class the highest number of hits were in Carreer and Security.

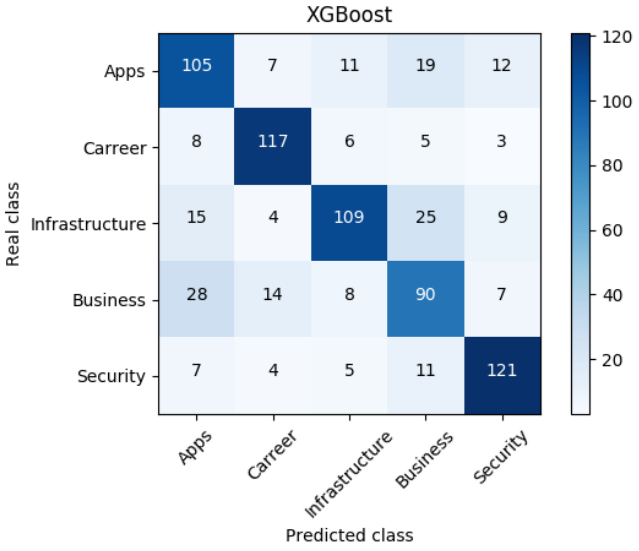

Figure 10. Confusion matrix of XGBoost in the ComputerWorld dataset using word2vec

5. Conclusions and Future Works

In this work, the objective was to classify a set of technology news from two Brazilian news portals - *Tecmundo* and *Computerworld* - by analyzing the news' title and a short abstract. The classification was made through the use of information retrieval techniques combined with four classification algorithms presented in the literature.

After analyzing the computational results, it was possible to observe that both TF/IDF and word2vec methods presented a similar performance, with TF/IDF having a slightly better classification accuracy. Considering the learning algorithms, MLP and XGBoost presented the best results when compared to Naive Bayes and Decision Tree.

A business condition that should be considered is the possibility that the portal news were badly classified on purpose due to a coordination request. This situation could happen in order to ensure that all sub-areas are regularly updated.

5.1. Future Works

During the development of this work, some interesting topics came up as complementary analysis and future improvements. These possible future works are described as follows:

- dataset development: it is worthy noticing that a big amount of news was necessary to train the models and, unfortunately, it was necessary to reduce the amount of data due to balancing problems. Some limitations were faced when building the dataset that would require some time to be trespassed;

- badly classified news analysis: the business condition mentioned creates a set of corrections that could be made. Besides that, it is possible to notice the sub areas like $Galaxy$ and $Moto$ that contains only a short amount of news that could be associated to another sub area, like $Mobile\ Devices$ in the given example;
- cross prediction: where news from one dataset would be classified using the model trained on other dataset. This task demands the two datasets to have a group of matching sub areas;

References

Asr, F., Willits, J., and Jones, M. (2016). Comparing predictive and co-occurrence based models of lexical semantics trained on child-directed speech. In *Proceedings of CogSci*.

Li, Y. H. and Jain, A. K. (1998). Classification of text documents. *The Computer Journal*, 41(8):537–546.

Rahmawati, D. and Khodra, M. L. (2016). Word2vec semantic representation in multi-label classification for indonesian news article. In *Advanced Informatics: Concepts, Theory And Application (ICAICTA), 2016 International Conference On*, pages 1–6. IEEE.

Wang, F. and Wen, S. (2016). Study of automatic annotation based on news tags. In *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016 IEEE*, pages 733–737. IEEE.