Product Categorization by Title Using Deep Neural Networks as Feature Extractor

Leonardo S. Paulucio*§, Thiago M. Paixão*†, Rodrigo F. Berriel*, Alberto F. De Souza*,
Claudine Badue* and Thiago Oliveira-Santos*
*Universidade Federal do Espírito Santo (UFES), Brazil
†Instituto Federal do Espírito Santo (IFES), Brazil
§Email: leonardo.paulucio@gmail.com

***Abstract*—Natural Language Processing (NLP) has been receiving increasing attention in the past few years. In part, this is related to the huge flow of data being made available everyday on the internet, which increased the need for automatic tools capable of analyzing and extracting relevant information, especially from the text. In this context, text classification became one of the most studied tasks on the NLP domain. The objective is to assign predefined categories or labels to text or sentences. Important applications include sentence classification, sentiment analysis, spam detection, among many others. This work proposes an automatic system for product categorization using only their titles. The proposed system employs a state-of-the-art deep neural network as a tool to extract features from the titles to be used as input in different machine learning models. The system is evaluated in the large-scale Mercado Libre dataset, which has the common characteristics of real-world problems such as imbalanced classes, unreliable labels, besides having a large number of samples: 20,000,000 in total. The results showed that the proposed system was able to correctly categorize the products with a balanced accuracy of 86.57% on the local test split of the Mercado Libre dataset. It also surpassed the fourth place on the public rank of the MeLi Data Challenge with 91.19% of balanced accuracy, which represents less than 1% of the difference to the winner.**

***Index Terms*—NLP, text classification, sentence classification, product categorization, deep neural networks, machine learning, artificial intelligence**

I. Introduction

Everyday, a massive amount of data is generated and made available on the internet. This huge flow of data has increased the need for automatic tools capable of analyzing and extracting relevant information, especially from the text. In this context, Natural Language Processing (NLP) has been receiving increasing attention in the past few years and many companies have been releasing products that rely on NLP tools. The main goal of NLP is enabling computers to process, manipulate, and, more importantly, understand natural language text or speech [1], whose data are mostly available in a semi- or unstructured way [2]. There are many tasks related to NLP, such as question answering, language translation, summarizing, text similarity, natural language generation, and text classification, being the latter the focus of this work.

This study was financed in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001; Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil (CNPq) – grants 311654/2019-3, 200864/2019-0 and 311504/2017-5; and Fundação de Amparo à Pesquisa do Espírito Santo (FAPES), Brazil – grant 84304057/18.

Text classification can be defined as the task of assigning predefined categories to text or sentences allowing them to be organized, grouped, structured, etc. This task can be performed manually or automatically. The first approach can be done by a human group that analyses the text content and assign it to a proper category. Despite the quality of this categorization approach, it is financially expensive and time-consuming, which makes it impractical for processing large amounts of data. On the other hand, automatic classification can be performed by a system, making the process cheaper and faster, usually at the cost of a lower quality, enabling the processing of an ever-increasing flow of data.

The text classification problem has been widely studied in the literature. Several works investigated the application of statistical and traditional machine learning algorithms: KNN [3], [4], Naive Bayes [5], [6], Regression [7], [8], Neural Networks [9] and Support Vector Machines (SVM) [10], [11].

Deep learning has been successfully applied to several research domains such as: in biology [12], in document recontruction [13], in autonomous driven [14]–[16], multi-domain learning [17], energy consumption [18], and facial expression recognition [19], [20]. In the past decade, deep neural network models started to achieve surprising results in NLP tasks [21]–[23]. Motivated by the success of Convolutional Neural Network (CNNs) in computer vision problems, some works [24]–[26] also investigated their application in the text classification. Li et al. [27] presented a combination of LSTM and CNN called Bi-LSTM-CNN to classify large-scale news text. Lai et al. [28] proposed a Recurrent Convolutional Neural Network to text classification tasks that outperformed previous state-of-the-art approaches in three datasets. Vaswani et al. [29] presented Transformer, a model based on attention mechanisms that dispense with the recurrence and convolutions. The Transformer model obtained superior quality on machine translation tasks while being more parallelizable and requiring significantly less time to train. In 2018, a Bidirectional Encoder Representations from Transformers (BERT) was introduced [30]. Based on the Transformer encoder, the BERT model reached state-of-the-art performance in eleven NLP tasks.

A text classification task that is particularly interesting is product categorization. Products can be categorized in many ways and based on many factors (e.g., size, price, color,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

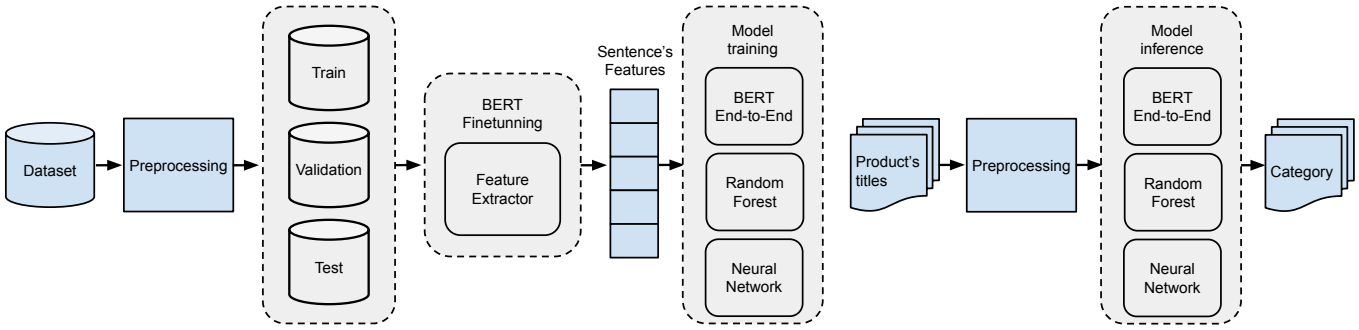


Fig. 1. Overview of the proposed system. First, the dataset is preprocessed following best practices in NLP (e.g., removing punctuation, accents, stop words; tokenization; etc.). Then, the dataset is split into train, validation and test sets for further investigation. These sets are used to fine-tune the BERT model. After that, the BERT model is employed to extract features from the sequences, and these features are used to train other machine learning models. Finally, the model is able to predict the category of a product based on the title.

etc.). This work focuses on categorizing products based on their titles, which may seem a simple task, but it presents some difficult challenges. First, there are many categories to which a product can be assigned. It is common that stores, especially e-commerce, sell a wide range of products, from toys to drinks to food and more. Second, the categories are structured and products are usually assigned to categories and subcategories. For instance, a toy car can be assigned both "TOYS" (category) and "TOY CARS" (subcategory) labels. Third, some categories are expected to include far more products than others, which represents an inherent imbalance among the categories. Lastly, the related datasets for this task are usually noisy. This is even worse for e-commerce websites that are intended to facilitate consumer-to-consumer sales, which is the focus of this work. In this case, the sellers themselves assign categories when putting up a new product advertisement. Regardless of the apparent success of this approach, there are two main problems: (i) the sellers can make honest mistakes assigning wrong categories to certain products, making them less/more visible to the costumers; and (ii) the sellers can deliberately categorize a product incorrectly with the purpose of bringing more attention to it, which may lead to a bad experience to the users potentially decreasing the overall sales.

There are a few works in the literature focused on product categorization. Kozareva [31] proposed an automatic product categorization system using different features such as N-gram, Bi-gram, LDA, etc. For the categorization, the authors investigated two algorithms: one-against-all (OAA) and error correction tournament (ECT). In our work, instead, we focus on deep learning-based features, because they have shown to be more effective in recent years. Cevahir and Murakami [32] used a combination of Deep Belief Nets and Deep autoencoders for categorizing a large-scale e-commerce Japanese product dataset using both titles and descriptions. Compared to them, we work with a problem formulation that is more restricted, in which only the title is available for categorization. Moreover, we are interested in the bilingual categorization, more specifically: Spanish and Portuguese. Another approach using the "divide and conquer" strategy was presented by [33]. The idea was to combine three models for categorizing products, each one responsible for classifying one of the three pieces of information available: titles, descriptions, and images. In our work, as only the title is available, multimedia approaches are beyond the scope of interest. Besides that, there are two main problems that hinder fair comparison with other methods: (i) none of them have publicly available implementations and (ii) most of them are using private datasets.

In this paper, we propose an automatic system for product categorization based only on their titles. The system employs a state-of-the-art deep neural network to extract features from the titles. Then, different machine learning models are investigated to handle the task of interest. The proposed system was evaluated in a large-scale dataset released for the MeLi Data Challenge with 20 million product titles and more than 1,500 categories. Our system was able to surpass the fourth place on the private Mercado Libre test set, achieving 91.19% of accuracy (less than 1% to the first place).

II. Product Categorization System

The proposed system for product categorization (illustrated in Fig. 1) comprises three main steps. The initial preprocessing step aims to remove potential noise and standardize the input samples. Then, the feature extract is fine-tuned to better extract the features in the task of interest. Finally, the models are trained and ready for inference, i.e., they can receive a product title as input and output a category.

A. Data Preprocessing

The input data (product titles) are made available in a freeway form, which also means they may include unnecessary and unwanted characters. To remove such characters and standardize the input, some procedures usually adopted in the NLP applications are employed. First, punctuation, accents, and numbers are removed. Second, the product titles are converted to lower case. Third, stop words from the languages of interest (in our case, Spanish and Portuguese) are removed. Lastly, a tokenization process is conducted and each token is associated with a numerical identifier to make the input

consistent with the BERT expected input (the multilingual BERT tokenizer was used in this step).

B. Feature Extractor

After preprocessing, the state-of-the-art BERT architecture is leveraged to extract features from the processed titles. The BERT architecture is composed of a set of stacked encoders from the Transformer architecture and uses attention mechanisms instead of recurrent connections as those seen in RNNs. BERT was chosen because of the proven high accuracy in text classification tasks, including the availability of a model pre-trained on a large and multilingual text corpus and that will be fine-tuned to our application of product categorization.

To enable fine-tuning, a fully-connected layer (classification layer) comprising C outputs was added on top of BERT-base, where C is the number of product categories. The resulting network is trained end-to-end as a classifier model by processing the preprocessed titles and comparing it with the ground-truth category labels. After fine-tuning, only the feature extractor – which projects preprocessed product titles into a $\mathbb{R}^{768}$ feature space – is kept.

C. Models Training and Inference

The proposed system is trained with supervision, i.e., in addition to the input samples, the corresponding category of each input must be known. During training time, the system receives as input a product's title in a freeway form. First, the title is preprocessed as explained in Subsection II-A. Then, a 768-dimensional feature is extracted from the preprocessed title. Finally, these features are fed to a classifier for training. Once the models are trained, they can be used to assign a category to a given product's titles, i.e., the model can predict to which category a given product belongs.

At inference time, the same preprocessing is applied to the product's titles. As we also investigated the application of ensembles, the final prediction is a combination of the predictions of many models. In our system, the models are combined based on the addition of the predicted probabilities (see Eq. 1). This vector has C positions. Each position i represents the probability of a product's title belonging to the i-th class. Then, all probabilities produced by each model are summed to make the ensemble. In the end, the position with the highest score will be the predicted class.

$$\text{Predicted Category} = \arg\max_{c} \sum_{m} \{\hat{y}_c | c \in [1..C]\}^m, \quad (1)$$

where $\{\hat{y}_c\}^m$ are the probabilities predicted by the m-th model for the C classes given a product's title.

III. Experimental Methodology

This section covers the following topics related to the assessment of the performance and robustness of the proposed system: dataset, implementation details of the investigated models, conducted experiments and performance metrics, and the experimental platform.

A. Dataset

The Mercado Libre dataset was released to the MeLi Data Challenge [34] and it is originally split into training and test sets. The training set consists of a list of 20,000,000 samples. For each sample (i.e., a product), four features are given: the title of the product, the language of the title, the category, and the label quality. The title of the product was given by the seller and is available without any preprocessing, i.e., it contains punctuation, cased letters, accentuation, etc. The language of the title can be either *spanish* or *portuguese* as most of its market is in Latin America. The category is a unique textual description of one of the 1588 categories of the dataset, including their subcategories (e.g., Printers, 3D-Printers, Souvenirs, Cameras, Umbrellas, etc.). It is important to note that a category is assigned regardless of the language of the title, i.e., the title of a cellphone product is expected to be categorized as "CELLPHONES" no matter whether the title is in Spanish or Portuguese. Lastly, the label quality is used to specify whether the category is reliable, i.e., verified by the Mercado Libre team (they did not grant 100% accuracy), or unreliable, i.e., the category was picked by the seller and not reviewed by the Mercado Libre team and one should expect a larger error rate in the category assignment when compared to the reliable ones. In total, only 1,184,245 samples (≈6%) were considered reliable. The reliable cases, however, are not uniformly distributed across categories, which means that some categories have a higher number of reliable cases and others have a low number (or even zero). In addition, the number of samples for each category are not balanced either. Fig. 2 shows the distribution of the dataset with respect to the categories and label reliability, in which the class imbalance and the low number of reliable samples can be observed.

The test set is a list of 246,955 products, each one containing 3 features: id, title, and language. The language and title represent the same as in the training set. The id column is useful only to identify each sample in the private submission system of the challenge. Participants of the challenge were evaluated by the performance on this restricted test set, therefore the correct categories are not publicly available.

B. Training and Implementation Details

Here we present the models that were investigated and the implementation details associated with each one.

BERT Fine-tuning: It was leveraged the *bert-base-multilingual* model that is composed of 12 stacked encoders, totalling 110M parameters. This model was trained on 104 languages with the largest Wikipedias. The adopted source code is publicly available (see Subsection III-E) and was recommended by the authors as an alternative to the original implementation. The model was fine-tuned with a batch size of 128 and a fixed learning rate of 0.001. The chosen optimizer was the Stochastic Gradient Descent (SGD) and selected Cross Entropy was cross entropy. Class weighting was adopted to alleviate data imbalance. The other hyper-parameters were the same used on the pre-trained model.

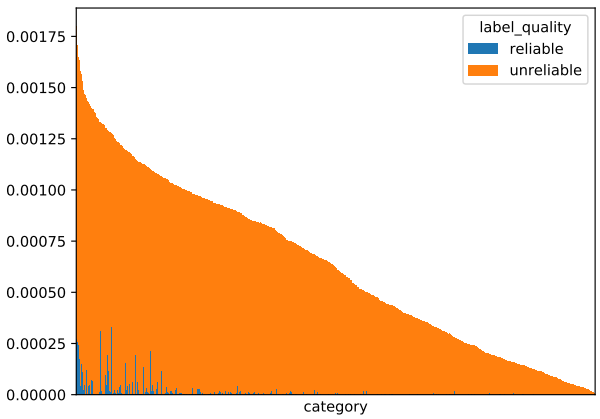


Fig. 2. Mercado Libre dataset distribution. The dataset is highly imbalanced and only $\approx 6\%$ is considered reliable.

Random Forest: Random Forest (RF) is basically an ensemble of Decision Trees. In our experiments, it was used only 50 trees due to the large amount of memory required by the construction of the trees. Apart from the number of trees, we used the framework's default values for the other hyper-parameters such as the criterion used for the branch split, maximum depth, and others.

Neural Network: Each neural network of the ensemble of NNs is a simple fully connected layer. The model was trained with a batch size of 128 with a fixed learning rate of 0.001. The loss function and optimizer employed were the Cross Entropy and SGD, respectively.

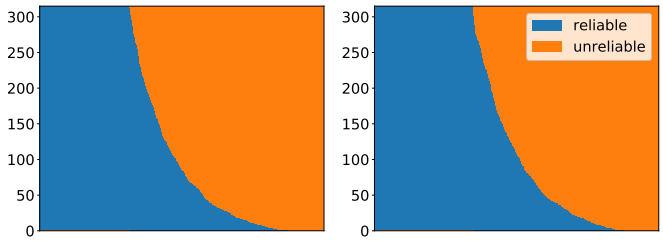
The local data splits, preprocessing and training codes will be made publicly available[1]. Other details such as the libraries and frameworks used are described in Subsection III-E.

C. Experiments

Two experiments were performed, and they are referred to as Local and Private experiment. In the first, given that the labels of the (private) test set are not available, we propose to split the public training dataset and validate the models using a *local* test set. In the second, the models were evaluated on the private test set via a submission server. Before the evaluation, the public training split of the Mercado Libre dataset ($\mathcal{D}$) was randomly subdivided into training ($\mathcal{D}_{train}$), validation ($\mathcal{D}_{val}$), and test ($\mathcal{D}_{test}$) partitions, each one comprising, respectively, 70%, 10%, and 20% of the samples of $\mathcal{D}$. The split was made so that the class distribution remained the same as in $\mathcal{D}$, which implies that the category distribution of each split is equally imbalanced. After that, the feature extractor was trained on $\mathcal{D}_{train}$ and the accuracy on $\mathcal{D}_{val}$ was measured at the end of every epoch. The training stopped on the epoch in which the validation accuracy started to decrease. After trained, the performance of this model was used as a baseline on both local and private test sets. Moreover, this model was used to extract the features for training the models in the experiments below. Because of memory limitations, a different procedure (data partitioning) was carried out for training the ensembles (both Random Forest and Neural Network).

Local experiment To carry out the experiments on the ensemble of random forests, 10 different "folds" were created from $\mathcal{D}_{local\_train} = \{\mathcal{D}_{train} \cup \mathcal{D}_{val}\}$. Each fold $\mathcal{D}^i_{local\_train}$ was created with 315 random samples of each category, with a total of 500,220 samples per fold. Due to the existence of unreliable samples, random reliable samples were drawn before the unreliable ones. Those categories with fewer than 315 reliable exemplars are filled with random unreliable samples. Then, with a fixed feature extractor, an ensemble of random forests was trained by training a model for each fold. For training the ensemble of neural networks (10 in total), $\mathcal{D}^i_{local\_train}$ was split into training (90%) and validation (10%) sets, so that the validation set was used to stop the training following the same protocol used in the fine-tuning of the feature extractor. The performance metrics were measured on $\mathcal{D}_{test}$.

Private experiment The main goal of this experiment is to enable the comparison with the results available on the public final results of the MeLi Data Challenge. The experiment on the private test set followed the same protocol of the local test set, except that the folds were created from the whole original $\mathcal{D}$ instead of $\mathcal{D}_{local\_train}$ and the metrics were measured by submitting the predictions to the server of the challenge (it was reopened after the challenge finished). The folds also comprised 500,220 samples with 315 samples of each category, however more reliable samples were available in comparison to the Local experiment. Therefore, the distribution of the folds used in the local experiment was slightly different (a small shift) from one of the folds used in the private experiment, as can be seen in Fig. 3.



(a) Local (b) Private

Fig. 3. Distribution of the reliable and unreliable samples among the categories in the folds of the local (a) and private (b) experiments.

D. Performance Metrics

The final goal of the proposed system is to correctly categorize the product's titles. In this context, two metrics were used to quantify the performance of the system. The first is the Balanced Accuracy metric, defined in Eq. 2:

[1]https://github.com/lspaulucio/product-categorization-ijcnn-2020/blob/master/README.md

$$\text{BACC} = \sum_{c=1}^{C} \frac{\text{RECALL}_c}{C}, \tag{2}$$

where $\text{RECALL}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}$, C is the number of categories, and TP_c and FN_c are the true positives and false negatives of the category c, respectively. This metric was chosen because, besides taking categories imbalance into account, it was also used in the MeLi Data Challenge. Therefore, our results on the private test set can be directly compared to those of the public challenge scoreboard.

The second metric is the Top-K accuracy (Eq. 3), where a prediction is counted as correct if the expected category is among the k largest probabilities of the predicted vector:

$$\text{Top-K} = \frac{1}{N} \sum_{i=1}^{N} [y_i \in p_i^k], \tag{3}$$

where N is the number of samples, Y_i is the ground-truth category of the i-th sample, p_i^k is the set of the k categories with the largest probabilities in the prediction, and $[\cdot]$ is the Iverson bracket. Four k values were considered: $\{3, 5, 7, 10\}$. This metric is particularly useful due to the fact that several categories are very similar, mainly because of the category-subcategory semantic (e.g., "VIDEO_GAMES" vs. "GAME_CONSOLE", "INSTRUMENT_AMPLIFIERS" vs. "AUDIO_AMPLIFIERS", "SHIRT" vs. "T-SHIRT"). This metric is also used in other challenges, such as the ImageNet Challenge [35].

E. Experimental Platform

The BERT model was trained on an Intel Xeon X5690 @ 3.47GHz $\times$ 24 with 32 GB of RAM, and 1 Titan Xp GPU with 12 GB of memory with NVIDIA CUDA 9.1 and cuDNN 7.0 installed. The dataset preprocessing and training of the machine learning models was performed on an Intel Xeon E7-4850 v4 (2.10GHz) with 128 vCPU (only 100 were used) and 256GB of RAM. The operating system running on both machines was Linux Ubuntu 16.04. The PyTorch framework [36] was adopted together with the BERT model implementation provided by [37], which is publicly available[2]. The exception is the Random Forest, in which the implementation available in the scikit-learn library [38] was used. The training sessions took, on average, 36 hours per epoch for the BERT fine-tuning, and 2 hours for inference on test sets. The training and inference for the Random Forest models took, on average, 10 hours for each Random Forest model (10 models were used in the ensemble). The neural network ensemble took, on average, 2.5 hours for training and inference.

IV. Results

Fig. 4 shows the results for the Local experiment. As can be seen, the end-to-end BERT model achieved the best performance considering all metrics. In the second place, the Neural Network ensemble achieved slightly better performance than the Random Forest model in all metrics. Using the Top-3 metric the accuracy of all models increased more than 5% compared to the balanced accuracy metric. For the Top-5, -7, and -10 metrics (in this order), however, the models' accuracy does not increase sharply much. This may have happened due to the heavy class imbalance associated with a large number of categories, which may have led some product categories not to be learned properly.

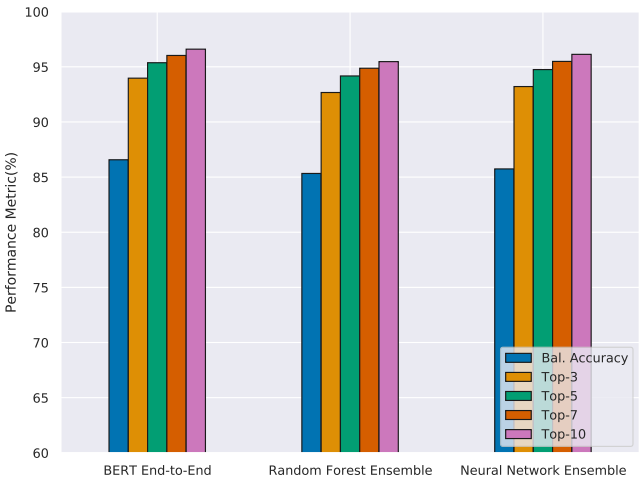


Fig. 4. Results of the Local experiment. All metrics used are presented for each model.

The results for the Private experiment are shown in Table I. The Random Forest ensemble achieved the best performance, although it performed only 1% higher than the worst-performing method (end-to-end BERT), followed by the Neural Network ensemble. The probable reason for this is that the folds used in this experiment were created using all the public training split (i.e., $\mathcal{D}_{train}$, $\mathcal{D}_{val}$ and $\mathcal{D}_{test}$), and it presents more reliable samples compared to $\mathcal{D}_{train}$ (70%) only, which was used to train the BERT since the training occurs during the fine-tuning process. Nevertheless, the features extracted with the BERT enabled the Random Forest ensemble to reach 91.190% on Mercado Libre submission system surpassing the fourth place obtained in the challenge, and getting close to the first three places for less than 1%, as shown in the second column of Table I. Despite the difference observed between the results achieved on local and private test sets may seem strange at first, they might have occurred due to the existence of samples with unreliable categories on the local test set. Thus, some of the wrong predictions made by the model may actually be the correct ones.

V. Conclusions

Text classification is a big challenge in the NLP domain. Despite the large amount of data available, mainly from the internet, the manual approach is impracticable, which makes the automation of such tasks extremely relevant. In this context, we proposed a product categorization system that predicts a product category based solely on its title. The system

[2]https://huggingface.co/transformers/model_doc/bert.html

TABLE I
BALANCED ACCURACY SCORES OBTAINED BY MODELS ON BOTH LOCAL AND PRIVATE EXPERIMENTS.

Model	Balanced Accuracy (in %)	
	Local Test Set	Private Test Set
BERT (end-to-end)	86.57	90.19
Random Forest Ensemble	85.33	91.19
Neural Network Ensemble	85.74	90.89
Public Rank of the Challenge		
• Top-1	–	91.73
• Top-4	–	91.04

employs a deep neural network model, the BERT, to extract sentence's features that are used to train machine learning models which are also compared with the BERT end-to-end (baseline).

The proposed system was evaluated in a large-scale real-world dataset with more than 20 million samples, the Mercado Libre dataset, released as part of the MeLi Data Challenge. In the local test set, the ensemble of Neural Networks performed better than the one of Random Forests. The latter, however, achieved the best results on the private test set: 91.19% of balanced accuracy. This performance is better than the fourth place in the public rank and has less than 1% of difference to the winner. The results show that the BERT model was able to extract relevant features from the products' titles allowing other machine learning models to achieve a performance comparable to the BERT end-to-end.

Future work includes a more comprehensive investigation of additional NLP preprocessing techniques, such as stemming and lemmatization, as well as of the use of other pre-trained word vectors, such as fastText [39] and GloVe [40]. Finally, we will also investigate how to leverage metric learning approaches (e.g., siamese networks, triplets networks) to improve the BERT embeddings.

ACKNOWLEDGMENT

The authors thank the support of NVIDIA Corporation for their donation of the Titan Xp GPU used in this research.

REFERENCES

[1] G. G. Chowdhury, "Natural Language Processing," *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 51–89, 2003.

[2] A. R. Sharma and P. Kaushik, "Literature Survey of Statistical, Deep and Reinforcement Learning in Natural Language Processing," in *International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017, pp. 350–354.

[3] S. Tan, "Neighbor-weighted K-nearest neighbor for unbalanced text corpus," *Expert Systems with Applications*, vol. 28, no. 4, pp. 667–671, 2005.

[4] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *Information Retrieval*, vol. 1, no. 1-2, pp. 69–90, 1999.

[5] D. D. Lewis and M. Ringuette, "A Comparison of Two Learning Algorithms for Text Categorization," in *Third Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 81–93.

[6] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature Selection for Text Classification with Naïve Bayes," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5432–5435, 2009.

[7] N. Fuhr, S. Hartmann, G. Lustig, M. Schwantner, K. Tzeras, and G. Knorz, "AIR/X - a Rule-Based Multistage Indexing System for Large Subject Fields," in *RIAO'91: Intelligent Text and Image Handling – Volume 2*, 1991, pp. 606–623.

[8] Y. Yang and C. G. Chute, "An example-based mapping method for text categorization and retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 12, no. 3, pp. 252–277, 1994.

[9] E. Wiener, J. O. Pedersen, A. S. Weigend *et al.*, "A Neural Network Approach to Topic Spotting," in *Symposium on Document Analysis and Information Retrieval (SDAIR)*, vol. 317, 1995, p. 332.

[10] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in *European Conference on Machine Learning (ECML)*, 1998, pp. 137–142.

[11] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive Learning Algorithms and Representations for Text Categorization," in *Conference on Information and Knowledge Management (CIKM)*, 1998, pp. 148–152.

[12] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2063–2079, 2018.

[13] T. M. Paixão, R. F. Berriel, M. C. S. Boeres, A. L. Koerich, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, "Fast(er) Reconstruction of Shredded Text Documents via Self-Supervised Deep Asymmetric Metric Learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] R. F. Berriel, F. S. Rossi, A. F. de Souza, and T. Oliveira-Santos, "Automatic Large-Scale Data Acquisition via Crowdsourcing for Crosswalk Classification: A Deep Learning Approach," *Computers & Graphics*, vol. 68, p. 32–42, 2017.

[15] V. F. Arruda, T. M. Paixão, R. F. Berriel, A. F. D. Souza, C. Badue, N. Sebe, and T. Oliveira-Santos, "Cross-Domain Car Detection Using Unsupervised Image-to-Image Translation: From Day to Night," in *International Joint Conference on Neural Networks (IJCNN)*, 2019.

[16] L. T. Torres, T. M. Paixão, R. F. Berriel, A. F. D. Souza, C. Badue, N. Sebe, and T. Oliveira-Santos, "Effortless Deep Training for Traffic Sign Detection Using Templates and Arbitrary Natural Images," in *International Joint Conference on Neural Networks (IJCNN)*, 2019.

[17] R. Berriel, S. Lathuili'ere, M. Nabi, T. Klein, T. Oliveira-Santos, N. Sebe, and E. Ricci, "Budget-Aware Adapters for Multi-Domain Learning," in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[18] R. F. Berriel, A. T. Lopes, A. Rodrigues, F. M. Varejão, and T. Oliveira-Santos, "Monthly Energy Consumption Forecast: A Deep Learning Approach," in *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[19] M. V. Zavarez, R. F. Berriel, and T. Oliveira-Santos, "Cross-database facial expression recognition based on fine-tuned deep convolutional network," in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2017, pp. 405–412.

[20] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos, "Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order," *Pattern Recognition*, vol. 61, pp. 610 – 628, 2017.

[21] W.-t. Yih, K. Toutanova, J. C. Platt, and C. Meek, "Learning discriminative projections for text similarity measures," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL)*, 2011, pp. 247–256.

[22] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research (JMLR)*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[23] F. Wei, H. Qin, S. Ye, and H. Zhao, "Empirical study of deep learning for text classification in legal document review," in *IEEE International Conference on Big Data (Big Data)*, 2018, pp. 3317–3320.

[24] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[25] X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 649–657.

[26] Z. Wang and Z. Qu, "Research on Web text classification algorithm based on improved CNN and SVM," in *IEEE International Conference on Communication Technology (ICCT)*, 2017, pp. 1958–1961.

[27] C. Li, G. Zhan, and Z. Li, "News Text Classification Based on Improved Bi-LSTM-CNN," in *IEEE International Conference on Information Technology in Medicine and Education (ITME)*, 2018, pp. 890–893.

[28] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification," in *Twenty-ninth Conference on Artificial Intelligence (AAAI)*, 2015.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[31] Z. Kozareva, "Everyone Likes Shopping! Multi-class Product Categorization for e-Commerce," in *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015, pp. 1329–1333.

[32] A. Cevahir and K. Murakami, "Large-scale Multi-class and Hierarchical Product Categorization for an E-commerce Giant," in *Proceedings of the International Conference on Computational Linguistics (COLING) Technical Papers*, 2016, pp. 525–535.

[33] P. Wirojwatanakul and A. Wangperawong, "Multi-Label Product Categorization Using Multi-Modal Fusion Models," *arXiv preprint arXiv:1907.00420*, 2019.

[34] MercadoLibre. (2019) MeLi Data Challenge 2019 - Mercadolibre. [Online]. Available: https://ml-challenge.mercadolibre.com/

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems (NeurIPS))*, 2019, pp. 8024–8035.

[37] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "HuggingFace's Transformers: State-of-the-art Natural Language Processing," *arXiv preprint arXiv:1910.03771*, 2019.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 2825–2830, 2011.

[39] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2018.

[40] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.