A Study on Multiple Tasks for e-Commerce Marketplaces

Cristian Cardellino and **Rafael Carrascosa**
Mercado Libre Argetina S.R.L.
cristian.cardellino@mercadolibre.com rafael.carrascosa@mercadolibre.com

Abstract

In an e-Commerce marketplace, there are multiple tasks that need to be addressed in a day-to-day basis. Some tasks such as product recommendations and product search take a fundamental role in the overall experience a user has on the site. There are however a multitude of lesser known tasks which are also relevant for the business and that need to be addressed with a comparatively smaller investment in teams and quality datasets. Examples of such tasks are the detection of counterfeit or forbidden items, the estimation of package sizes, etc. In this work we study a set of different baseline models and how they work across different tasks that come from real world data.

1 Introduction

The e-Commerce environment has been growing at a fast rate in recent years. As such, new tasks propose new challenges to be solved. Some key tasks like product search and recommendation receive a lot of attention and resources (dedicated teams, large amounts of quality data, etc.). On the other hand, there are many other tasks in an e-Commerce, which are very relevant as well but not as widely discussed, which need to be resolved with a, comparatively, smaller investment of company resources. Examples of these are counterfeit detection, package size estimation, etc.

In the present work we study a set of different baselines to assess how well they work across multiple tasks in a e-Commerce marketplace. We do so in a bilingual fashion, since our marketplace has Spanish and Portuguese as their main languages of interaction (based on the countries it is currently active). The models come from supervised as well as semi-supervised algorithms and focus specially in the use of the products titles. We do a thorough analysis of the advantages and disadvantages of each model and conclude with the best line of work coming from the results. We also plan to release some of our real world datasets for the community to do benchmarking on them.

We explore the use of several models for benchmarking a series of tasks on industrial datasets of our marketplace, including fastText (Joulin et al. 2016), Meta-Prod2Vec (Vasile, Smirnova, and Conneau 2016), Text Convolutional Networks (Kim 2014) and BERT (Devlin et al. 2018).

The main contributions of our paper are the following: 1) set of downstream tasks for benchmarking, 2) a study of different strong baselines for multiple tasks, 3) the study is set on products from a Spanish and Portuguese marketplace, thus extending the area of research to other languages.

This paper is structured as follows: Section 2 establishes the related work, with special focus on presenting the models used in this work. Section 3 presents the datasets and the tasks we are addressing in the current work. Section 4 defines all our experimental configuration. Section 5 shows our results and finally 6 concludes with a brief discussion of our findings.

2 Related Work

In the past few years, the use of models pre-trained on large unlabeled datasets have seen a dramatic increase. The first of such models was Word2Vec (Mikolov et al. 2013) and some of its variants that work with subword information such as fastText (Joulin et al. 2016), which can establish good representations that can be fed into different supervised algorithms. More recently, the use of large self-supervised models with different applications such as BERT (Devlin et al. 2018) have proven very useful on many different supervised tasks with limited data available. In particular we explore the use of DistilBERT (Sanh et al. 2019) as a lighter weight alternative to BERT.

For e-Commerce, there is an extensive research work for some of the main tasks like product search and recommendation. A good example of such is the Prod2Vec (Grbovic et al. 2015) algorithm as well as the improvement using metadata that is Meta-Prod2Vec (Vasile, Smirnova, and Conneau 2016). Both algorithms are based on the use of Word2Vec (Mikolov et al. 2013) with a sequence of products in a user shopping session. We will be exploring the use of Meta-Prod2Vec as a way to represent a product and use it as features of a classifier.

Finally, we came across the use of Text Convolutional Neural Networks (Text CNN) (Kim 2014) which combined with Sentence Piece Tokenization (Kudo and Richardson 2018) as a way to avoid out-of-vocabulary (OOV) errors proved to be a very good candidate across the multiple tasks.

3 Datasets

In this section we will establish the datasets that were used in our experimental evaluation. We have our set of "Downstream Tasks" that are composed by labeled datasets for three kind of supervised tasks: binary classification, multiclass classification and regression. On the other hand we have an unlabeled sessions dataset that is used to calculate the embeddings used by the Meta-Prod2Vec algorithm.

To assert more confidently that our results generalize to new settings, we experimented with data coming from two countries where our marketplace is present, on in Portuguese and the other in Spanish. Each country has it's own set of datasets, and although the tasks are the same, there are different amounts of data for each country.

Downstream tasks

We collected 3 datasets built to work on different supervised tasks from our marketplace.

Counterfeit Products A dataset of products in our marketplace that were reported to be counterfeit or in violation of trademarks or intellectual property. This dataset has two classes, fake or original. The dataset has 117,475 products for Portuguese and 66,908 for Spanish. It is balanced for both classes.

Forbidden Products A dataset of products that are prohibited to being sold in our marketplace. There are two tasks that come from this dataset, the forbidden product detection task, which is a binary classification, and the multiclass classfication of the policy that classifies the reason why the product is prohibited. This last task has two levels of granularity, where some policies can have a subpolicy (e.g. the forbidden article which falls in the policy "document" can be either the document of a person or of a car). The dataset is comprised of 1,085,639 products for Portuguese and 568,159 products for Spanish. In each country, 25% of the dataset is labeled as forbidden.

Products Dimensions A dataset with products dimensions (height, length and width). We use this dataset for two tasks: A binary classification task that establish if it is not possible to automatically ship the product, named the product is non-machinable, which is based on whether any of the three dimensions of the product surpasses the 70 cm long. The other is a regression task, as we want to estimate the "volumetric weight" of a package. This is done following the formula used by DHL[1], which is the most popular. As we use the metric system and the dimensions are in centimeters, the volumetric weight represents 5,000 cm^3/kg: $\text{VW} = \frac{\text{length} \times \text{height} \times \text{widht}}{5000}$. The dataset has 3,655,548 products for Portuguese and 826,133 products for Spanish. This dataset has only 3% of products that are not elegible for free shipping meaning that is not only the largest one, but also the most unbalanced.

The tasks datasets were divided into train, test and validation with a 70/15/15 split. The validation set is for hyperparameter tuning and the final results are taken from the test set. These tasks have, in all cases, three main components: the title of the product, the category, and the label of the task, however except for the case of the Meta-Prod2Vec algorithm, we only use the title of the product as feature. The titles of all the datasets were normalized by removing all especial characters (including accents of both Spanish and Portuguese), lowercased and removed stopwords for each language.

[1] https://www.dhl.com/en/tools/volumetric_weight_express.html

Sessions Dataset

The sessions dataset has information on items search and views done by different users in our marketplace, as well as the information on the items themselves (metadata like the title, the price and the category), it is not associated to any of the downstream tasks established before, but was used to train a Meta-Prod2Vec model in order to represent the items of the downstream tasks. For the task we splitted the datasets into sessions of 5 minutes length given a sequence of products. Each product the user visits is part of the browsing session of the user. A session ends when at least 5 minutes have passed without new visits. The product titles in this dataset were also normalized as it was the case for the downstream tasks datasets. The dataset consists of 1,723,826 from our Portuguese marketplace and 378,451 from our Spanish marketplace. It has 354,907 user stories from the Portuguese marketplace and 58,256 from the Spanish.

4 Experimental Evaluation

Models

fastText For classification tasks (binary and multiclass), we use fastText's text categorization (Joulin et al. 2016) engine. The model hyperparameters are set by the `autotuneValidationFile` that the model takes, passing the validation split of each dataset for that case.

Bag-of-Words + Linear Regression For the regression tasks we cannot use fastText since it does not support it. For this case we use the a Bag-of-Words model with words, bigrams and trigrams and a Linear Regression with L2 regularization, with a learning rate and regularization rate parameter of 0.01 selected based on the validation dataset for volumetric weight estimation. Our experiments discarded the use of this combination for classification (using a logistic regression instead of linear regression) because it would consistently be worse in performance than the fastText baseline.

Text CNN + Sentece Piece Tokenization We use Kim's Text CNN (Kim 2014) with a combination of Sentence Piece Tokenization (Kudo and Richardson 2018) to avoid dealing with out-of-vocabulary (OOV) tokens and limit the projected space of the tokens. The tokenizer was trained with the normalized titles of all the products coming from all the datasets (the labeled and the unlabeled ones) with a total of 30,000 tokens. The Text CNN model is used both for classification and regression tasks. It proved to be a very strong model, especially in the cases of regression and for multitask classification (where it showed very good results over fastText for the classes with less training data). The hyperparameters were set based on the validation dataset of the counterfeit product detection task and those same are used across all the other tasks. The model embeddings of the sentence piece

tokens are learned along the tasks, they have a dimension of 512, with 4 kernels of size 2, 3, 4, and 5 tokens (these are 1D CNNs), a total of 128 filters for each kernel size, with ReLU activations and a Global Max Pooling operation. The Text CNN feature map has a final dimension of 512. It is trained using the LARS optimizer (You, Gitman, and Ginsburg 2017) with a batch size of 4096, a learning rate of 0.001 and a weight decay of 0.0001, for a maximum of 50 epochs, with an early stopping of 5 epochs without improving the loss on the validation data of the task. We did some preliminary experiments using pre-trained word embeddings, but we failed to see better results than what sentence piece tokens trained specifically for the task obtained, and we have the plus of avoiding OOV words.

DistilBERT We evaluate our downstream tasks using the multilingual version of DistilBERT (Sanh et al. 2019). We decided to go with this version since it was faster and more suitable for a production environment. Still, the use of DistilBERT proved to be much more resource intensive that Text CNNs, even though it was only used for inference, i.e. we only use the embedding for the `[CLS]` token. For this we use the implementation of the Huggingface Library (Wolf et al. 2019). We feed the encoded `[CLS]` to a logistic regression classifier using the liblinear solver (Fan et al. 2008). For the case of regression tasks, we use the same configuration as the Linear Regression that was used for the Bag-of-Words representation.

Meta-Prod2Vec + KNN Finally, we wanted to asses the impact of a representation obtained via the Meta-Prod2Vec algorithm. Unlike the rest of the models presented here, which only use the text of the product title, this algorithm depends on other metadata, the category of the product, that was available when training the algorithm with the sessions data, but that might not be present on the downstream tasks. It was used alongside a K Nearest Neighbor classifier or regressor, depending on the task. The value of K is 5 and we use cosine similarity as a metric for the algorithm. To train the model we used the sessions dataset for each country (i.e. we trained one Meta-Prod2Vec model for each country), with the id of the product as the main vector and the title and category of the item as the metadata. The model dimension is 512 (i.e. matching the dimensions of the feature vector of Text CNNs), and trained using negative sampling and a window of 5 products for 5 epochs. When using it on the downstream tasks, if the product id is present on the model we use that as a vector, otherwise we use the mean of the vectors of the metadata.

Metrics

The selected metrics were chosen to evaluate model performance and diagnostic, we are particularly interested in assess the case where classes are imbalanced. We have three metrics depending on the task:

Area Under ROC (AUCROC) Used for binary tasks. It establishes a threshold independent metric and it is a good standard to use in unbalanced datasets.

F1-Score Macro Average Used in multiclass classification tasks as we are interested not only in the most common class, but also in less frequent classes (which are usually the more interesting) to asses the impact on the long tail of the distribution.

R2 Score Used in regression tasks. It is a good way to compare across different tasks with different units as it gives an upper bound to see how good the model is performing.

5 Results and Discussion

Table 1 shows the results for each task, in each language, and each model. The table is divided in three sections, one for each type of task: binary classification, multiclass classification and regression. The tasks are measured by AUCROC, F1-Score Macro Average and R2 Score respectively. The best results for each language in each task are highlighted with bold numbers.

For the cases of Text CNN and fastText, as they can change according to the random initialization of its parameters, we decided to run 10 trials of each of them for each model in each task and each site, changing the random initialization seed, and we report the average results and their standard deviation. For the other models this level of detail was not relevant as they converged to a single solution.

The binary tasks are the detection of counterfeit/forbidden items and the non eligibility for free shipment. The multiclass tasks the dataset is the same, it has two levels of granularity, the more general "policy" and the more specific "subpolicy". Finally, the regression task is done on the same dataset as the "shipment" task, but with the volumetric weight as target.

Text CNN has an advantage over the other models in most of the tasks, with fastText being a close follower for classification tasks and Meta-Prod2Vec being a good model when it comes to regression tasks. It is interesting to point out that, unlike fastText which actively uses the validation split to actively do hyperparameter tuning for each task, none of the other models were tuned specifically for each task, and rely only on the hyperparameter tuning of the counterfeit task, which implies there is a lot of room to improve upon the found results. In the multiclass tasks, Text CNN shows a good performance over fastText in classes with less training data, which are usually the more interesting ones.

Another interesting thing to see is that DistilBERT falls short for almost all the tasks, specially in comparison to any of the others. Perhaps a larger, language specific model (e.g. BERT large but trained for Spanish or Portuguese) could incur in better results, but the DistilBERT model was particularly expensive to train, even though we only use inference.

6 Conclusions

In this work we presented a set of problems that are part of our marketplace environment. The solutions to these business problems need to be effective using less resources than would be available for tasks like recommendation or search, plus the preference for this systems is that they be able to resolve multiple tasks with the least amount of effort. The results shown in the previous section hint of Text CNN being a very good all around solution for the many problems. Is particularly interesting in the case of multiclass classification because it can achieve very good performance on rarely seen

Task	Language	Model	AUCROC
Counterfeit	Portuguese	DistilBERT	94.5
		FastText	97.6 ± 0.04
		MP2V	88.7
		Text CNN	$\mathbf{98.7 \pm 0.04}$
	Spanish	DistilBERT	90.6
		FastText	96.0 ± 0.09
		MP2V	81.6
		Text CNN	$\mathbf{96.9 \pm 0.02}$
Forbidden	Portuguese	DistilBERT	88.5
		FastText	94.8 ± 0.94
		MP2V	86.4
		Text CNN	$\mathbf{96.3 \pm 0.04}$
	Spanish	DistilBERT	87.8
		FastText	92.5 ± 1.64
		MP2V	82.3
		Text CNN	$\mathbf{93.6 \pm 0.09}$
Shipment	Portuguese	DistilBERT	74.1
		FastText	83.2 ± 0.04
		MP2V	75.1
		Text CNN	$\mathbf{84.2 \pm 0.09}$
	Spanish	DistilBERT	72.7
		FastText	84.4 ± 0.21
		MP2V	77.5
		Text CNN	$\mathbf{85.5 \pm 0.16}$
Task	Language	Model	F1 Macro
Policy	Portuguese	DistilBERT	56.0
		FastText	69.9 ± 0.70
		MP2V	47.6
		Text CNN	$\mathbf{70.6 \pm 0.54}$
	Spanish	DistilBERT	62.4
		FastText	70.1 ± 1.12
		MP2V	62.3
		Text CNN	$\mathbf{70.8 \pm 0.34}$
Subpolicy	Portuguese	DistilBERT	37.9
		FastText	51.8 ± 6.48
		MP2V	33.8
		Text CNN	$\mathbf{55.2 \pm 1.09}$
	Spanish	DistilBERT	38.3
		FastText	49.2 ± 2.98
		MP2V	45.2
		Text CNN	$\mathbf{51.5 \pm 1.22}$
Task	Language	Model	R2 Score
V. Weight	Portuguese	BoW	0.22
		DistilBERT	0.04
		MP2V	0.30
		Text CNN	$\mathbf{0.37 \pm 0.01}$
	Spanish	BoW	0.08
		DistilBERT	0.01
		MP2V	0.12
		Text CNN	$\mathbf{0.13 \pm 0.00}$

Table 1: Results per task, language and model.

labels, which are usually the more interesting ones. The use of Meta-Prod2Vec vectors and a Nearest Neighbor model can also render competitive results for the regression cases.

Finally, the results show that the use of transformers models (i.e. DistilBERT) does not guarantee good results specially when working with non English languages.

References

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.* 9:1871–1874.

Grbovic, M.; Radosavljevic, V.; Djuric, N.; Bhamidipati, N.; Savla, J.; Bhagwan, V.; and Sharp, D. 2015. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, 1809–1818.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Kudo, T., and Richardson, J. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 66–71. Association for Computational Linguistics.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108.

Vasile, F.; Smirnova, E.; and Conneau, A. 2016. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, 225–232.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv* abs/1910.03771.

You, Y.; Gitman, I.; and Ginsburg, B. 2017. Scaling SGD batch size to 32k for imagenet training. *CoRR* abs/1708.03888.