ProBERT: Product Data Classification with Fine-tuning BERT Model

Hamada M. Zahera and Mohamed A. Sherif

Data Science Group, Paderborn University, Germany
{hamada.zahera, mohamed.sherif}@uni-paderborn.de

Abstract. In this paper, we describe our submission to the semantic web challenge on mining the product data in websites (MWPD2020). The dataset provided $19K$ instances of product data collected from various websites. The task is to predict the category, defined as hierarchical taxonomy as provided in the training set, of the product titles in the test set. In our approach, we present a simple BERT-based model (dubbed ProBERT) for classifying product data into one or more categories. We trained our system on products titles and descriptions to learn semantic representation. The participated systems are evaluated using weighted-average precision, recall and F1-score.

1 Introduction

Recently, many e-commerce websites are embedding structured product data into their content; according to the statistics from web data common[1], there are 37% of web pages or 30% of websites contain structured data. Consequently, these structured data can be used for product data integration and optimize product search service [8]. In addition, product categorization becomes essential in providing personalized recommendations and targeting advertisements. However, classifying product data is a challenging task due to the intrinsic noisy nature of the product labels, the size of modern e-commerce catalogues. In addition, each website has its a different structure of their product data, we refer to it as site-specific annotation [5, 1]. For example, one product like a T-shirt can have different annotation labels in different websites (`College>T-Shirts,Clothing >Tops>Shirts,Clothing accessories>Clothing>Tops`). To train robust models in these cases, we need large amount of training data with balanced classes. Therefore, automated product classification is need to further organize these data semantically into a universal categorization system regardless of their site-specific annotation.

In this paper, we explain our method to solve this problem through the semantic web challenge on mining HTML-embedded product data (MWPD2020[2]). The challenge aims to mining product data embedded into websites content. Previous studies [3, 6] focused on categorizing product data on a single e-commerce website and sensitive to

[1] http://webdatacommons.org/structureddata/2018-12/stats/stats.html

[2] https://ir-ischool-uos.github.io/mwpd/

it's site-specific content. In this challenge, the goal is to predict each product's categories based on datasets from different websites. We address this task as a multi-label classification problem, where each product can be assigned more than one class (i.e., label or category) simultaneously.

The latest development in language models (e.g., BERT) have shown impressive gains in a wide variety of natural language tasks ranging from sentence classification to sequence labeling. In our approach, we propose a BERT-based neural model to categorize a product based on it's meta-data such as product name, description or site-specific annotation. In particular, we employ a fine-tune BERT model to represent product data as low-dimensional contextualized vector. We feed our model with product name and description to capture semantic representation for product information. We summarize our main contributions in this paper as follows:

- We presented ProBERT, a BERT-based model for multi-label product classification based on product meta data (e.g name, description and site annotations).
- We conducted different experiments to benchmark the impact of different embeddings approaches. The result indicates that our method can be a good baseline with contextualized embedding (BERT) for product classification.

The rest of this paper is organized as follows: We first explore the dataset used in the challenge in section 2. Then, we present our proposed approach and the official results in sections 3 and 4 respectively. In section 5, we conclude the paper with some discussion about future work.

2 Dataset

The dataset is provided in the `JSON` format and divided into three subsets: (1) training contains approximately 10*k* product instances, (2) validation contains 3*k* instances and (3) 3*k* instances used for evaluated and testing the submitted systems. The product attributes in the dataset as follows:

- ID: refers to the product identification number.
- Name: is the product name (can be an empty string if unavailable).
- Description: is the description of product (truncated to a maximum of 5*k* characters. can be an empty string if unavailable).
- CategoryText: is the website-specific category for a product, or breadcrum (an empty string if unavailable).
- URL: refers to the original web page URL of the product.

Each product may be assigned one or more from the following classification levels, corresponding to the three GS1 GPC classification levels:

- lvl1: the level 1 GS1 GPC classification.
- lvl2: the level 2 GS1 GPC classification.
- lvl3: the level 3 GS1 GPC classification.

3 Approach

In this section, we present ProBERT, our simple BERT-based model for multi-label product classification. BERT is a pre-trained transformer network [2], which set for various NLP tasks new state-of-the-art results including text classification [7] and natural language understanding [4]. When we adopt BERT to NLP tasks in a target domain, a proper fine-tuning strategy, where a task-specific layer is added on top of BERT architecture. In this work, we leverage the BERT-Base pre-trained model with these details: Uncased: 12-layer, 768-hidden, 12-heads, $110M$ parameters. Then, we add a fully-connected layer (i.e Dense). For multilabel classification purpose, we use binary-cross-entropy as in Eq.1 loss function and sigmoid activation function to replace the original softmax. All hyper-parameters remain as default values, except we set *max_seq_length* as 30 words per input sequence.

$$L = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(H(x_i))) + (1 - y_i)\log(1 - H(x_i))] \tag{1}$$

where y_i and $H(x_i)$ denote ground-truth and predicted categories for each product. x_i refers to the feature vector obtained from the BERT model.

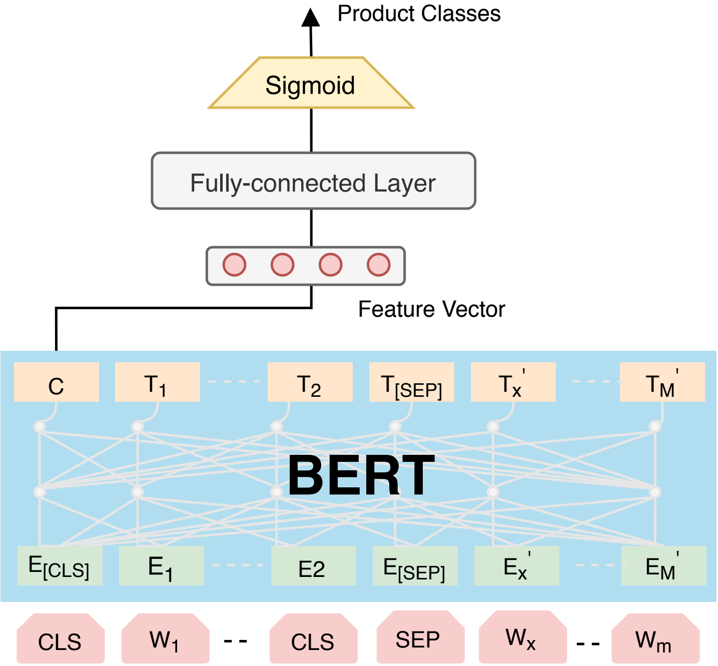


Fig. 1: ProBERT: A Fine-tuned BERT Model for Multi-label Product Categorization.

The general architecture of BERT is shown in Figure 1. We use a combined text of product title and description as an input features. Then, we do standard preprocessing which lower-casing and lemmatization of text. Then, a special preprocessing is performed for BERT processing; first inserting two special tokens. (CLS) is appended to the beginning of the text, another special token (SEP) is inserted after each sentence as an indicator of sentence boundary. The modified text is then represented as a sequence of tokens $X = [w_1, w_2, \ldots, w_n]$. Each token w_i is assigned three kinds of embeddings: token embedding, segmentation embedding and position embedding. These three embeddings are summed to a single input vector (C), which captures the overall meaning of the input.

4 Experiments

4.1 Evaluation

The evaluation metrics used in this challenge are precision, recall and F1. F1-score in Eq. 2 is the harmonic mean of precision and recall scores. The organizers used *macro-averaged* F1 score as the main metric to compare and rank the participating systems.

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{2}$$

4.2 Results

The organizers provided an overview of the performance of baselines with different embedding approaches (FastText, CBOW and Skipgram) on the validation dataset. As shown in Table 1, the baselines are evaluated based on both weighted-average and macro-average F1-scores. The experimental results are promising and shows that the systems based on embedding methods can achieve good F1-scores. Hence, we proposed our approach to employ the state-of-art contextualized embedding such as BERT to benchmark the system performance.

Table 1: Experimental Results

Model	**Weighted Avg. P, R, F1**			**Macro Avg. P, R, F1**		
Baseline	85.553	84.167	84.255	66.164	60.709	61.542
Baseline+embeddings(CBOW)	86.498	86.000	85.734	70.639	63.925	65.551
Baseline+embeddings(Skipgram)	85.453	84.911	84.575	70.574	62.740	64.693

The results are reported in terms of three evaluation metrics: (precision, recall and F1-score). F1-score is the score ultimately used to compare and rank the participating systems. Table 2 shows the results of five participating teams and the baseline (FastText). Our team (DICE_UPB) submitted one system based on fine-tuning BERT model.

The performance is close to the baseline system in terms of F1 score (81.84% compared to baseline 84.26%). However, we found that feature engineering needs a special preprocessing rather than the standard preprocessing, due to the nature of product data such as: highly imbalanced in labels as shown in Figures 2a and 2b; noisiness in the descriptions. We suggest to perform the same preprocessing as [8] and change our strategy of fine-tuning BERT model to address these challenges properly.

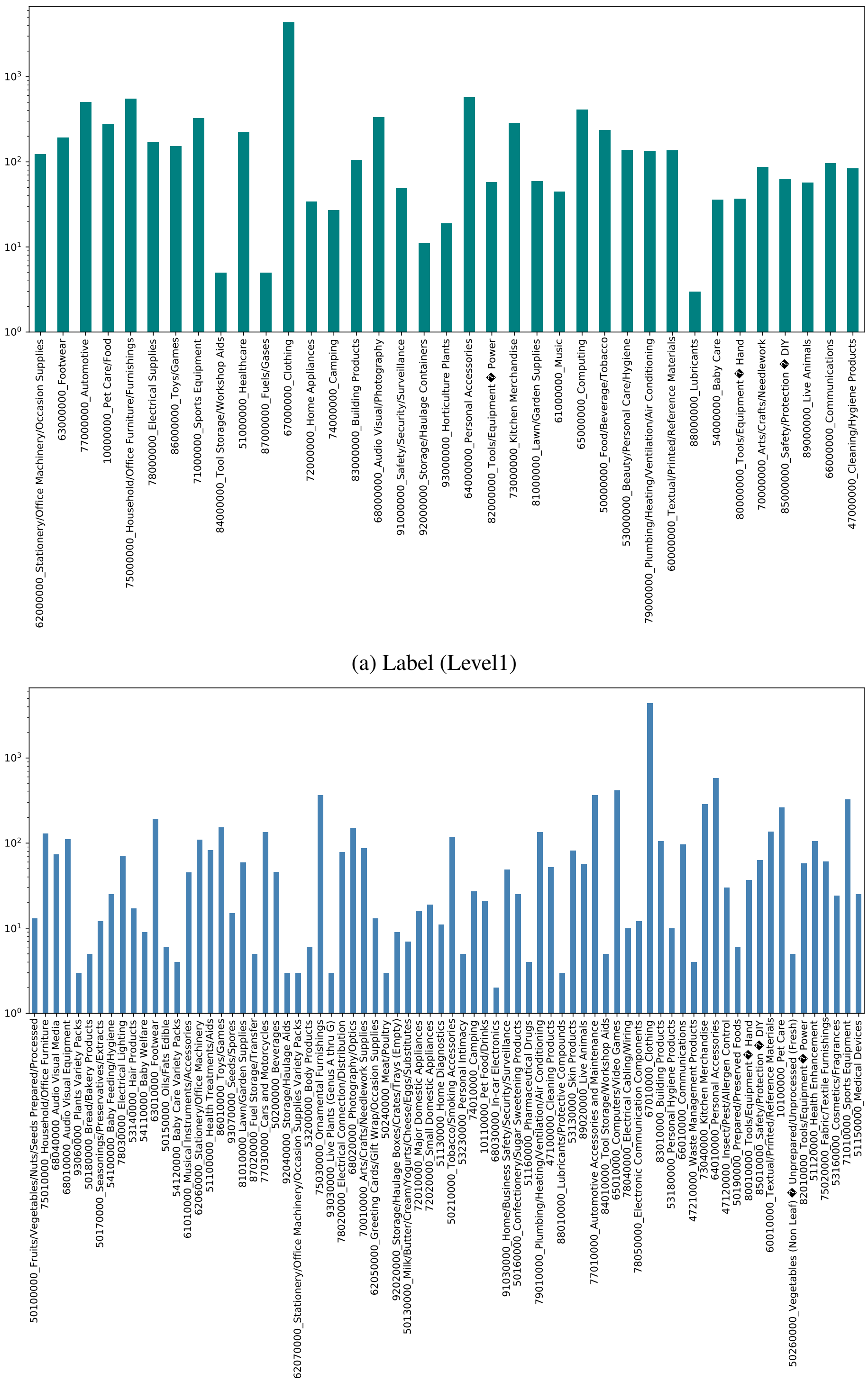
Table 2: System Evaluation Results. R2 refers to the systems which participated in the second round. Best Results in Bold

System	**Precision**	**Recall**	**F1-score**
Rhinobird	**89.01**	**89.04**	**88.62**
Rhinobird (R2)	88.97	88.72	88.43
Team ISI	87.16	86.85	86.54
ASVinSpace	86.96	86.30	86.10
Megagon	84.98	84.98	84.98
Baseline FastText	85.55	84.17	84.26
DICE_UPB	85.30	81.49	81.84

5 Conclusion and Future Work

In this paper, we described our approach (ProBert) to classify product data based on micro annotations. Our approach leverage a simple BERT model that represents a single feature vector from product's title and description, then predicts it's categories. Our experiments suggest that ProBERT is a good baseline to benchmark the task of automatic products classification. In the future, we plan to re-evaluate our approach with different preprocessing and fine-tuning strategies. Also, we will investigate more deep models with different architectures (e.g., graph-based neural model).

Acknowledgments This work has been supported by the EU H2020 project KnowGraphs (GA no. 860801) as well as the BMVI projects LIMBO (GA no. 19F2029C) and OPAL (GA no. 19F2028A).



(a) Label (Level1)

(b) Label (Level2)

Fig. 2: Label distributions (*log scaled*) in the training dataset.

Bibliography

[1] Cevahir, A., Murakami, K.: Large-scale multi-class and hierarchical product categorization for an e-commerce giant. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 525–535 (2016)

[2] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[3] Gupta, V., Karnick, H., Bansal, A., Jhala, P.: Product classification in e-commerce using distributional semantics. arXiv preprint arXiv:1606.06083 (2016)

[4] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q.: Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351 (2019)

[5] Kozareva, Z.: Everyone likes shopping! multi-class product categorization for e-commerce. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1329–1333 (2015)

[6] Xia, Y., Levine, A., Das, P., Di Fabbrizio, G., Shinzato, K., Datta, A.: Large-scale categorization of japanese product titles using neural attention models. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 663–668 (2017)

[7] Zahera, H.M., Elgendy, I.A., Jalota, R., Sherif, M.A.: Fine-tuned bert model for multi-label tweets classification. In: TREC (2019)

[8] Zhang, Z., Paramita, M.: Product classification using microdata annotations. In: International Semantic Web Conference. pp. 716–732. Springer (2019)