BERT Goes Shopping: Comparing Distributional Models for Product Representations

Federico Bianchi
Bocconi University
Milano, Italy
f.bianchi@unibocconi.it*

Bingqing Yu
Coveo
Montreal, CA
cyu2@coveo.com

Jacopo Tagliabue
Coveo AI Labs
New York, United States
jtagliabue@coveo.com†

Abstract

Word embeddings (e.g., word2vec) have been applied successfully to eCommerce products through *prod2vec*. Inspired by the recent performance improvements on several NLP tasks brought by contextualized embeddings, we propose to transfer BERT-like architectures to eCommerce: our model – *Prod2BERT* – is trained to generate representations of products through masked session modeling. Through extensive experiments over multiple shops, different tasks, and a range of design choices, we systematically compare the accuracy of *Prod2BERT* and *prod2vec* embeddings: while *Prod2BERT* is found to be superior in several scenarios, we highlight the importance of resources and hyperparameters in the best performing models. Finally, we provide guidelines to practitioners for training embeddings under a variety of computational and data constraints.

1 Introduction

Distributional semantics (Landauer and Dumais, 1997) is built on the assumption that the meaning of a word is given by the contexts in which it appears: word embeddings obtained from co-occurrence patterns through *word2vec* (Mikolov et al., 2013), proved to be both accurate by themselves in representing lexical meaning, and very useful as components of larger Natural Language Processing (NLP) architectures (Lample et al., 2018). The empirical success and scalability of *word2vec* gave rise to many domain-specific models (Ng, 2017; Grover and Leskovec, 2016; Yan et al., 2017): in eCommerce, *prod2vec* is trained replacing words in a sentence with product interactions in a shopping session (Grbovic et al., 2015), eventually generating vector representations of the products. The key intuition is the same underlying *word2vec* – you can tell a lot about a product by the company it keeps (in shopping sessions). The model enjoyed immediate success in the field and is now essential to NLP and Information Retrieval (IR) use cases in eCommerce (Vasile et al., 2016a; Bianchi et al., 2020).

As a key improvement over *word2vec*, the NLP community has recently introduced *contextualized representations*, in which a word like *play* would have different embeddings depending on the general topic (e.g. a sentence about *theater* vs *soccer*), whereas in *word2vec* the word *play* is going to have only one vector. Transformer-based architectures (Vaswani et al., 2017) in large-scale models - such as BERT (Devlin et al., 2019) - achieved SOTA results in many tasks (Nozza et al., 2020; Rogers et al., 2020). As Transformers are being applied outside of NLP (Chen et al., 2020), it is natural to ask whether we are missing a fruitful analogy with product representations. It is *a priori* reasonable to think that a pair of sneakers can have different representations depending on the shopping context: is the user interested in buying these shoes because they are running shoes, or because these shoes are made by her favorite brand?

In *this* work, we explore the adaptation of *BERT*-like architectures to eCommerce: through extensive experimentation on downstream tasks and empirical benchmarks on typical digital retailers, we discuss advantages and disadvantages of contextualized embeddings when compared to traditional *prod2vec*. We summarize our main contributions as follows:

1. we propose and implement a BERT-based contextualized product embeddings model (hence, **Prod2BERT**), which can be trained with online shopper behavioral data and produce product embeddings to be leveraged by

*Federico and Bingqing contributed equally to this research.

†Corresponding author.

Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4), pages 1–12

downstream tasks;

2. we benchmark Prod2BERT against *prod2vec* embeddings, showing the potential accuracy gain of contextual representations across different shops and data requirements. By testing on shops that differ for traffic, catalog, and data distribution, we increase our confidence that our findings are indeed applicable to a vast class of typical retailers;

3. we perform extensive experiments by varying hyperparameters, architectures and fine-tuning strategies. We report detailed results from numerous evaluation tasks, and finally provide recommendations on how to best trade off accuracy with training cost;

4. we share our code[1], to help practitioners replicate our findings on other shops and improve on our benchmarks.

1.1 Product Embeddings: an Industry Perspective

The eCommerce industry has been steadily growing in recent years: according to U.S. Department of Commerce (2020), 16% of all retail transactions now occur online; worldwide eCommerce is estimated to turn into a $4.5 trillion industry in 2021 (Statista Research Department, 2020). Interest from researchers has been growing at the same pace (Tsagkias et al., 2020), stimulated by challenging problems and by the large-scale impact that machine learning systems have in the space (Pichestapong, 2019). Within the fast adoption of deep learning methods in the field (Ma et al., 2020; Zhang et al., 2020; Yuan et al., 2020), product representations obtained through *prod2vec* play a key role in many neural architectures: after training, a product space can be used directly (Vasile et al., 2016b), as a part of larger systems for recommendation (Tagliabue et al., 2020b), or in downstream NLP/IR tasks (Tagliabue and Yu, 2020). Combining the size of the market with the past success of NLP models in the space, investigating whether Transformer-based architectures result in superior product representations is both theoretically interesting and practically important.

Anticipating some of the themes below, it is worth mentioning that our study sits at the intersection of two important trends: on one side, neural models typically show significant improvements at large scale (Kaplan et al., 2020) – by quantifying expected gains for "reasonable-sized" shops, our results are relevant also outside a few public companies (Tagliabue et al., 2021), and allow for a principled trade-off between accuracy and ethical considerations (Strubell et al., 2019); on the other side, the rise of multi-tenant players[2] makes sophisticated models potentially available to an unprecedented number of shops – in this regard, we design our methodology to include *multiple* shops in our benchmarks, and report how training resources and accuracy scale across deployments. For these reasons, we believe our findings will be interesting to a wide range of researchers and practitioners.

2 Related Work

Distributional Models. *Word2vec* (Mikolov et al., 2013) enjoyed great success in NLP thanks to its computational efficiency, unsupervised nature and accurate semantic content (Levy et al., 2015; Al-Saqqa and Awajan, 2019; Lample et al., 2018). Recently, models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) shifted much of the community attention to Transformer architectures and their performance (Talmor and Berant, 2019; Vilares et al., 2020), while it is increasingly clear that big datasets (Kaplan et al., 2020) and substantial computing resources play a role in the overall accuracy of these architectures; in our experiments, we explicitly address robustness by *i*) varying model designs, together with other hyperparameters; and *ii*) test on multiple shops, differing in traffic, industry and product catalog.

Product Embeddings. *Prod2vec* is a straightforward adaptation to eCommerce of *word2vec* (Grbovic et al., 2015). Product embeddings quickly became a fundamental component for recommendation and personalization systems (Caselles-Dupré et al., 2018; Tagliabue et al., 2020a), as well as NLP-based predictions (Bianchi et al., 2020). To the best of our knowledge, *this* work is the first to explicitly investigate whether Transformer-based architectures deliver higher-quality product representations compared to non-contextual embeddings. Eschauzier (2020) uses Transformers on cart

[1] Code available at https://github.com/vinid/prodb

[2] As an indication of the market opportunity, in the space of AI-powered search and recommendations we recently witnessed Algolia (Techcrunch, 2019a) and Lucidworks raising 100M USD (Techcrunch, 2019c), Coveo raising 227M CAD (Techcrunch, 2019b), Bloomreach raising 115M USD (Techcrunch, 2021).

co-occurrence patterns with the specific goal of basket completion – while similar in the masking procedure, the breadth of the work and the evaluation methodology is very different: as convincingly argued by Requena et al. (2020), benchmarking models on unrealistic datasets make findings less relevant for practitioners outside of "Big Tech". Our work features extensive tests on real-world datasets, which are indeed representative of a large portion of the mid-to-long tail of the market; moreover, we benchmark several fine-tuning strategies from the latest NLP literature (Section 5.2), sharing – together with our code – important practical lessons for academia and industry peers. The closest work in the literature as far as architecture goes is *BERT4Rec* (Sun et al., 2019), i.e. a model based on Transformers trained end-to-end for recommendations. The focus of *this* work is not so much the gains induced by Transformers in sequence modelling, but instead is the quality of the representations obtained through unsupervised pretraining – while recommendations are important, the breadth of *prod2vec* literature (Bianchi et al., 2021b,a; Tagliabue and Yu, 2020) shows the need for a more thorough and general assessment. Our methodology helps uncover a tighter-than-expected gap between the models in downstream tasks, and our industry-specific benchmarks allow us to draw novel conclusions on optimal model design across a variety of scenarios, and to give practitioners actionable insights for deployment.

3 Prod2BERT

3.1 Overview

The Prod2BERT model is taking inspiration from BERT architecture and aims to learn context-dependent vector representation of products from online session logs. By considering a shopping session as a "sentence" and the products shoppers interact with as "words", we can transfer masked language modeling (MLM) from NLP to eCommerce. Framing sessions as sentences is a natural modelling choice for several reasons: first, it mimics the successful architecture of *prod2vec*; second, by exploiting BERT bi-directional nature, each prediction of a masked token/product will make use of past and future shopping choices: if a shopping journey is (typically) a progression of intent from exploration to purchase (Harbich et al., 2017), it seems natural that sequential modelling may capture relevant dimensions in the underlying vocabulary/catalog. Once trained, Prod2BERT becomes capable of predicting masked tokens, as well as providing context-specific product embeddings for downstream tasks.

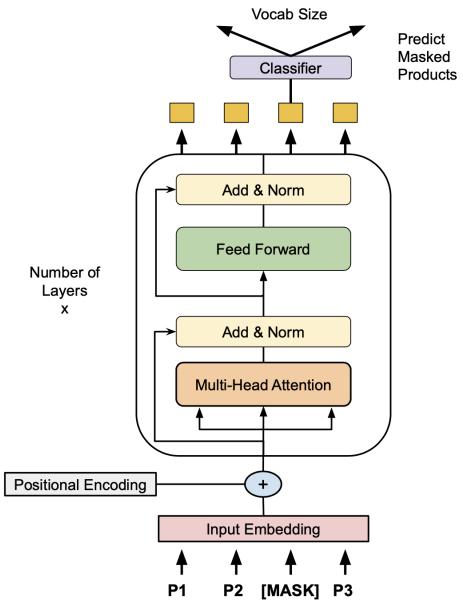

Figure 1: Overall architecture of Prod2BERT pretrained on MLM task.

3.2 Model Architecture

As shown in Figure 1, Prod2BERT is based on a transformed based architecture Vaswani et al. (2017), emulating the successful BERT model. Please note that, different from BERT's original implementation, a white-space tokenizer is first used to split an input session into tokens, each one representing a product ID; tokens are combined with positional encodings via addition and fed into a stack of self-attention layers, where each layer contains a block for multi-head attention, followed by a simple feed forward network. After obtaining the output from the last self-attention layer, the vectors corresponding to the masked tokens pass through a softmax to generate the final predictions.

3.3 Training Objective

Similar to Liu et al. (2019); Sun et al. (2019), we train Prod2BERT from scratch with the MLM objective. A random portion of the tokens (i.e., the product IDs) in the original sequence are chosen for possible replacements with the *MASK* token; and the masked version of the sequence is fed into the model as input: Figure 2 shows qualitatively the relevant data transformations, from the original

Figure 2: Transformation of sequential data, from the original data generating process – i.e. a shopping session –, to telemetry data collected by the SDK, to the masked sequence fed into Prod2BERT.

shopping session, to the telemetry data, to the final masking sequence. The target output sequence is exactly the original sequence without any masking, thus the training objective is to predict the original value of the masked tokens, based on the context provided by their surrounding unmasked tokens. The model learns to minimize categorical cross-entropy loss, taking into account only the predicted masked tokens, i.e. the output of the non-masked tokens is discarded for back-propagation.

3.4 Hyperparameters and Design Choices

There is growing literature investigating how different hyperparameters and architectural choices can affect Transformer-based models. For example, Lan et al. (2020) observed diminishing returns when increasing the number of layers after a certain point; Liu et al. (2019) showed improved performance when modifying masking strategy and using duplicated data; finally, Kaplan et al. (2020) reported slightly different findings from previous studies on factors influencing Transformers performance. Hence, it is worth studying the role of hyperparameters and model designs for Prod2BERT, in order to narrow down which settings are the best given the specific target of our work, i.e. *product representations*. Table 1 shows the relevant hyperparameter and design variants for Prod2BERT; following improvement in data generalization reported by Liu et al. (2019), when $duplicated = 1$ we augmented the original dataset repeating each session 5 times.[3] We set the embedding size to 64 after preliminary optimizations: as other values offered no improvements, we report results only for one size.

[3]This procedure ensures that each sequence can be masked in 5 different ways during training.

Parameter	Values
# epochs [e]	10, 20, 50, 100
# layers [l]	4, 8
masking probability [m]	0.15, 0.25
duplicated [d]	1, 0

Table 1: Hyperparameters and their ranges.

4 Methods

4.1 Prod2vec: a Baseline Model

We benchmark Prod2BERT against the industry standard *prod2vec* (Grbovic et al., 2015). More specifically, we train a CBOW model with negative sampling over shopping sessions (Mikolov et al., 2013). Since the role of hyperparameters in *prod2vec* has been extensively studied before (Caselles-Dupré et al., 2018), we prepare embeddings according to the best practices in Bianchi et al. (2020) and employ the following configuration: $window = 15$, $iterations = 30$, $ns\_exponent = 0.75$, $dimensions = [48, 100]$. While *prod2vec* is chosen because of our focus on the quality of the learned representations – and not just performance on sequential inference *per se* – it is worth nothing that kNN (Latifi et al., 2020) over appropriate spaces is also a surprisingly hard baseline to beat in many practical recommendation settings. It is worth mentioning that for both *prod2vec* and Prod2BERT we are mainly interested in producing a dense space capturing the latent similarity between SKUs: other important relationships between products (substitution (Zuo et al., 2020), hierarchy (Nickel and Kiela, 2017) etc.) may require different embedding techniques (or extensions, such as interaction-specific embeddings (Zhao et al., 2020)).

4.2 Dataset

We collected search logs and detailed shopping sessions from two partnering shops, **Shop A** and **Shop B**: similarly to the dataset released by Requena et al. (2020), we employ the standard definition of "session" from Google Analytics[4], with a total of five different product actions tracked: *detail*, *add*, *purchase*, *remove*, *click*[5]. Shop A and Shop B are

[4]https://support.google.com/analytics/answer/2731565?hl=en

[5]Please note that, as in many previous embedding studies (Caselles-Dupré et al., 2018; Bianchi et al., 2020), action

Shop	Sessions	Products	50/75 *pct*
Shop A	1,970,832	38,486	5, 7
Shop B	3,992,794	102,942	5, 7

Table 2: Descriptive statistics for the training dataset. *pct* shows 50^{th} and 75^{th} percentiles of the session length.

mid-sized digital shops, with revenues between 25 and 100 millions USD/year; however, they differ in many aspects, from traffic, to conversion rate, to catalog structure: Shop A is in the sport apparel category, whereas Shop B is in home improvement. Sessions for training are sampled with undisclosed probability from the period of March-December 2019; testing sessions are a completely disjoint dataset from January 2020. After pre-processing[6], descriptive statistics for the training set for Shop A and Shop B are detailed in Table 2. For fairness of comparison, the exact same datasets are used for both Prod2BERT and *prod2vec*.

Testing on fine-grained, recent data from *multiple* shops is important to support the internal validity (i.e. "is this improvement due to the model or some underlying data quirks?") and the external validity (i.e. "can this method be applied robustly across deployments, e.g. Tagliabue et al. (2020b)"?) of our findings.

5 Experiments

5.1 Experiment #1: Next Event Prediction

Next Event Prediction (NEP) is our first evaluation task, since it is a standard way to evaluate the quality of product representations (Letham et al., 2013; Caselles-Dupré et al., 2018): briefly, NEP consists in predicting the next action the shopper is going to perform given her past actions. Hence, in the case of Prod2BERT, we mask the last item of every session and fit the sequence as input to a pre-trained Prod2BERT model[7]. Provided with the model's output sequence, we take the top K most likely values for the masked token, and perform comparison with the true interaction. As for *prod2vec*, we perform the NEP task by following industry best practices (Bianchi et al., 2020): given a trained *prod2vec*, we take all the before-last items in a session to construct a session vector by average pooling, and use kNN to predict the last item[8]. Following industry standards, *nDCG@K* (Mitra and Craswell, 2018) with $K = 10$ is the chosen metric[9], and all tests ran on $10,000$ testing cases (test set is randomly sampled first, and then shared across Prod2BERT and *prod2vec* to guarantee a fair comparison).

5.1.1 Results

Model	Config	**Shop A**	**Shop B**
Prod2BERT	$e = 10, l = 4,$ $m = 0.25, d = 0$	0.433	0.259
Prod2BERT	$e = 5, l = 4,$ $m = 0.25, d = 1$	**0.458**	**0.282**
Prod2BERT	$e = 10, l = 8,$ $m = 0.25, d = 0$	0.027	0.260
Prod2BERT	$e = 100, l = 4,$ $m = 0.25, d = 0$	0.427	0.255
Prod2BERT	$e = 10, l = 4,$ $m = 0.15, d = 0$	0.416	0.242
prod2vec	$dimension = 48$	<u>0.326</u>	0.214
prod2vec	$dimension = 100$	0.326	<u>0.218</u>

Table 3: *nDCG@10* on NEP task for both shops with Prod2BERT and *prod2vec* (**bold** are best scores for Prod2BERT; <u>underline</u> are best scores for *prod2vec*).

Table 3 reports results on the NEP task by highlighting some key configurations that led to competitive performances. Prod2BERT is significantly superior to *prod2vec*, scoring up to 40% higher than the best *prod2vec* configurations. Since shopping sessions are significantly shorter than sentence lengths in Devlin et al. (2019), we found that changing masking probability from 0.15 (value from standard BERT) to 0.25 consistently improved performance by making the training more effective. As for the number of layers, similar to Lan et al. (2020), we found that adding layers helps only up until a point: with $l = 8$, training Prod2BERT with more layers resulted in a catastrophic drop in model performance for the smaller Shop A; however, the

type is not considered when preparing session for training.

[6]We only keep sessions that have between 3 and 20 product interactions, to eliminate unreasonably short sessions and ensure computation efficiency.

[7]Note that this is similar to the word prediction task for cloze sentences in the NLP literature (Petroni et al., 2019).

[8]Previous work using LSTM in NEP (Tagliabue et al., 2020b) showed some improvements over kNN; however, the differences cannot explain the gap we have found between *prod2vec* and Prod2BERT. Hence, kNN is chosen here for consistency with the relevant literature.

[9]We also tracked *HR@10*, but given insights were similar, we omitted it for brevity in what follows.

Model	Time A-B	Cost A-B
prod2vec	4-20	0.006-0.033$
Prod2BERT	240-1200	48.96-244.8$

Table 4: Time (minutes) and cost (USD) for training one model instance, per shop: *prod2vec* is trained on a *c4.large* instance, Prod2BERT is trained (10 epochs) on a *Tesla V100 16GB* GPU from *p3.8xlarge* instance.

same model trained on the bigger Shop B obtained a small boost. Finally, duplicating training data has been shown to bring consistent improvements: while keeping all other hyperparameters constant, using duplicated data results in an up to 9% increase in *nDCG@10*, not to mention that after only 5 training epochs the model outperforms other configurations trained for 10 epochs or more.

While encouraging, the performance gap between Prod2BERT and *prod2vec* is consistent with Transformers performance on sequential tasks (Sun et al., 2019). However, as argued in Section 1.1, product representations are used as input to many downstream systems, making it essential to evaluate how the learned embeddings generalize outside of the pure sequential setting. Our second experiment is therefore designed to test how well contextual representations transfer to other eCommerce tasks, helping us to assess the accuracy/cost trade-off when difference in training resources between the two models is significant: as reported by Table 4, the difference (in USD) between *prod2vec* and Prod2BERT is several order of magnitudes.[10]

5.2 Experiment #2: Intent Prediction

A crucial element in the success of Transformer-based language model is the possibility of adapting the representation learned through pre-training to new tasks: for example, the original Devlin et al. (2019) fine-tuned the pre-trained model on 11 downstream NLP tasks. However, the practical significance of these results is still unclear: on one hand, Li et al. (2020); Reimers and Gurevych (2019) observed that sometimes BERT contextual embeddings can underperform a simple GloVe (Pennington et al., 2014) model; on the other, Mosbach et al. (2020) highlights catastrophic forgetting, vanishing gradients and data variance as important factors in practical failures. Hence, given the range of downstream applications and the active debate on transferability in NLP, we investigate how Prod2BERT representations perform when used in the *intent prediction* task.

Intent prediction is the task of guessing whether a shopping session will eventually end in the user adding items to the cart (signaling purchasing intention). Since small increases in conversion can be translated into massive revenue boosting, this task is both a crucial problem in the industry and an active area of research (Toth et al., 2017; Requena et al., 2020). To implement the intent prediction task, we randomly sample from our dataset $20,000$ sessions ending with an add-to-cart actions and $20,000$ sessions without add-to-cart, and split the resulting dataset for training, validation and test. Hence, given the list of previous products that a user has interacted with, the goal of the intent model is to predict whether an add-to-cart event will happen or not. We experimented with several adaptation techniques inspired by the most recent NLP literature (Peters et al., 2019; Li et al., 2020):

1. *Feature extraction (static)*: we extract the contextual representations from a target hidden layer of pre-trained Prod2BERT, and through average pooling, feed them as input to a multi-layer perceptron (MLP) classifier to generate the binary prediction. In addition to alternating between the first hidden layer (*enc_0*) to the last hidden layer (*enc_3*), we also tried concatenation (*concat*), i.e. combining embeddings of all hidden layers via concatenation before average pooling.

2. *Feature extraction (learned)*: we implement a linear weighted combination of all hidden layers (*wal*), with learnable parameters, as input features to the MLP model (Peters et al., 2019).

3. *Fine-tuning*: we take the pre-trained model up until the last hidden layer and add the MLP classifier on top for intent prediction (*fine-tune*). During training, both Prod2BERT and task-specific parameters are trainable.

As for our baseline, i.e. *prod2vec*, we implement the intent prediction task by encoding each product within a session with its *prod2vec* embeddings, and

[10] Costs are from official AWS pricing, with 0.10 USD/h for the *c4.large* (https://aws.amazon.com/it/ec2/pricing/on-demand/), and 12,24 USD/h for the *p3.8xlarge* (https://aws.amazon.com/it/ec2/instance-types/p3/). While obviously cost optimizations are possible, the "naive" pricing is a good proxy to appreciate the difference between the two methods.

Model	*Method*	Shop	Accuracy
Prod2BERT	*enc_0*	*Shop B*	**0.567**
Prod2BERT	*enc_3*	*Shop B*	0.547
Prod2BERT	*concat*	*Shop B*	0.553
Prod2BERT	*wal*	*Shop B*	0.543
Prod2BERT	*fine-tune*	*Shop B*	0.560
prod2vec	-	*Shop B*	0.558
Prod2BERT	*enc_0*	*Shop A*	0.593
prod2vec	-	*Shop A*	**0.602**

Table 5: Accuracy scores in the intent prediction task (best scores for each shop in **bold**).

feeding them to a LSTM network (so that it can learn sequential information) followed by a binary classifier to obtain the final prediction.

5.2.1 Results

From our experiments, Table 5 highlights the most interesting results obtained from adapting to the new task the best-performing Prod2BERT and *prod2vec* models from NEP. As a first consideration, the shallowest layer of Prod2BERT for feature extraction outperforms all other layers, and even beats concatenation and weighted average strategies[11]. Second, the quality of contextual representations of Prod2BERT is highly dependent on the amount of data used in the pre-training phase. Comparing Table 3 with Table 5, even though the model delivers strong results in the NEP task on Shop A, its performance on the intent prediction task is weak, as it remains inferior to *prod2vec* across all settings. In other words, the limited amount of traffic from Shop A is not enough to let Prod2BERT form high-quality product representations; however, the model can still effectively perform well on the NEP task, especially since the nature of NEP is closely aligned with the pre-training task. Third, fine-tuning instability is encountered and has a severe impact on model performance. Since the amount of data available for intent prediction is not nearly as important as the data utilized for pre-training Prod2BERT, overfitting proved to be a challenging aspect throughout our fine-tuning experiments. Fourth, by comparing the results of our best method against the model learnt with *prod2vec* embeddings, we observed *prod2vec* embeddings can only provide limited values for intent estimation and the LSTM-based model stops to improve very quickly; in contrast, the features provided by Prod2BERT embeddings seem to encode more valuable information, allowing the model to be trained for longer epochs and eventually reaching a higher accuracy score. As a more general consideration – reinforced by a qualitative visual assessment of clusters in the resulting vector space –, the performance gap is *very small*, especially considering that long training and extensive optimizations are needed to take advantage of the contextual embeddings.

[11] This is consistent with Peters et al. (2019), which states that inner layers of a pre-trained BERT encode more transferable features.

6 Conclusion and Future Work

Inspired by the success of Transformer-based models in NLP, *this* work explores contextualized product representations as trained through a BERT-inspired neural network, *Prod2BERT*. By thoroughly benchmarking Prod2BERT against *prod2vec* in a multi-shop setting, we were able to uncover important insights on the relationship between hyperparameters, adaptation strategies and eCommerce performances on one side, and we could quantify for the first time quality gains across different deployment scenarios, on the other. If we were to sum up our findings for interested practitioners, these are our highlights:

1. Generally speaking, our experimental setting proved that pre-training Prod2BERT with Mask Language Modeling can be applied successfully to sequential prediction problems in eCommerce. These results provide independent confirmation for the findings in Sun et al. (2019), where BERT was used for in-session recommendations over academic datasets. However, the tighter gap on downstream tasks suggests that Transformers' ability to model long-range dependencies may be more important than pure representational quality in the NEP task, as also confirmed by human inspection of the product spaces (see Appendix A for comparative t-SNE plots).

2. Our investigation on adapting pre-trained contextual embeddings for downstream tasks featured several strategies in feature extraction and fine-tuning. Our analysis showed that feature-based adaptation leads to the peak performance, as compared to its fine-tuning counterpart.

3. Dataset size *does* indeed matter: as evident from the performance difference in Table 5, Prod2BERT shows bigger gains with the largest amount of training data available. Considering the amount of resources needed to train and optimize Prod2BERT (Section 5.1.1), the gains of contextualized embedding may not be worth the investment for shops outside the top 5k in the Alexa ranking[12]; on the other hand, our results demonstrate that with careful optimization, shops with a large user base and significant resources may achieve superior results with Prod2BERT.

While our findings are encouraging, there are still many interesting questions to tackle when pushing Prod2BERT further. In particular, our results require a more detailed discussion with respect to the success of BERT for textual representations, with a focus on the differences between words and products: for example, an important aspect of BERT is the tokenizer, that splits words into subwords; this component is absent in our setting because there exists no straightforward concept of "sub-product" – while far from conclusive, it should be noted that our preliminary experiments using categories as "morphemes" that attach to product identifiers did not produce significant improvements. We leave the answer to these questions – as well as the possibility of adapting Prod2BERT to even more tasks – to the next iteration of this project.

As a parting note, we would like to emphasize that Prod2BERT has been so far the largest and (economically) more significant experiment run by *Coveo*: while we *do* believe that the methodology and findings here presented have significant practical value for the community, we also recognize that, for example, not all possible ablation studies were performed in the present work. As Bianchi and Hovy (2021) describe, replicating and comparing some models is rapidly becoming prohibitive in term of costs for both companies and universities. Even if the debate on the social impact of large-scale models often feels very complex (Thompson et al., 2020; Bender et al., 2021) – and, sometimes, removed from our day-to-day duties – Prod2BERT gave us a glimpse of what unequal access to resources may mean in more meaningful contexts. While we (as in "humanity we") try to find a solution, we (as in "authors we") may find temporary solace knowing that good ol' *prod2vec* is still pretty competitive.

[12] See https://www.alexa.com/topsites.

7 Ethical Considerations

User data has been collected by *Coveo* in the process of providing business services: data is collected and processed in an anonymized fashion, in compliance with existing legislation. In particular, the target dataset uses only anonymous uuids to label events and, as such, it does not contain any information that can be linked to physical entities.

References

Samar Al-Saqqa and Arafat Awajan. 2019. The use of word2vec model in sentiment analysis: A survey. In *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control*, pages 39–43.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Federico Bianchi, Ciro Greco, and Jacopo Tagliabue. 2021a. Language in a (search) box: Grounding language learning in real-world human-machine interaction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4409–4415, Online. Association for Computational Linguistics.

Federico Bianchi and Dirk Hovy. 2021. On the gap between adoption and understanding in nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics.

Federico Bianchi, Jacopo Tagliabue, and Bingqing Yu. 2021b. Query2Prod2Vec: Grounded word embeddings for eCommerce. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 154–162, Online. Association for Computational Linguistics.

Federico Bianchi, Jacopo Tagliabue, Bingqing Yu, Luca Bigon, and Ciro Greco. 2020. Fantastic embeddings and how to align them: Zero-shot inference in a multi-shop scenario. In *Proceedings of the SIGIR 2020 eCom workshop, July 2020, Virtual Event, published at http://ceur-ws.org (to appear)*.

Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. 2018. Word2vec applied to recommendation: hyperparameters matter. In *Proceedings*

of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018, pages 352–356. ACM.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ruben Eschauzier. 2020. ProdBERT: Shopping basket completion using bidirectional encoder representations from transformers. In *Bachelor's Thesis*.

Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 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, Sydney, NSW, Australia, August 10-13, 2015*, pages 1809–1818. ACM.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864. ACM.

Matthieu Harbich, Gaël Bernard, P. Berkes, B. Garbinato, and P. Andritsos. 2017. Discovering customer journey maps using a mixture of markov models. In *SIMPDA*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Sara Latifi, Noemi Mauro, and D. Jannach. 2020. Session-aware recommendation: A surprising quest for the state-of-the-art. *ArXiv*, abs/2011.03424.

Benjamin Letham, Cynthia Rudin, and David Madigan. 2013. Sequential event prediction. *Machine learning*, 93(2-3):357–380.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*, abs/1907.11692.

Yifei Ma, Balakrishnan (Murali) Narayanaswamy, Haibin Lin, and Hao Ding. 2020. Temporal-contextual recommendation in real-time. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2291–2299. ACM.

Laurens van der Maaten and Geoffrey Hinton. 2008. Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Bhaskar Mitra and Nick Craswell. 2018. An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval*, 13(1):1–126.

Marius Mosbach, Maksym Andriushchenko, and D. Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *ArXiv*, abs/2006.04884.

Patrick Ng. 2017. dna2vec: Consistent vector representations of variable-length k-mers. *ArXiv*, abs/1701.06279.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6338–6347.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [MASK]? making sense of language-specific BERT models. *arXiv preprint arXiv:2003.02912*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Ann Pichestapong. 2019. Website personalization: Improving conversion with personalized shopping experiences.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Borja Requena, Giovanni Cassani, Jacopo Tagliabue, Ciro Greco, and Lucas Lacasa. 2020. Shopper intent prediction from clickstream e-commerce data with minimal browsing information. *Scientific Reports*, 2020:16983.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Statista Research Department. 2020. Global retail e-commerce sales 2014-2023.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1441–1450. ACM.

Jacopo Tagliabue, Ciro Greco, Jean-Francis Roy, Bingqing Yu, Patrick John Chia, Federico Bianchi, and Giovanni Cassani. 2021. Sigir 2021 e-commerce workshop data challenge.

Jacopo Tagliabue and Bingqing Yu. 2020. Shopping in the multiverse: A counterfactual approach to in-session attribution. *ArXiv*, abs/2007.10087.

Jacopo Tagliabue, Bingqing Yu, and Marie Beaulieu. 2020a. How to grow a (product) tree: Personalized category suggestions for eCommerce type-ahead. In *Proceedings of The 3rd Workshop on e-Commerce and NLP*, pages 7–18, Seattle, WA, USA. Association for Computational Linguistics.

Jacopo Tagliabue, Bingqing Yu, and Federico Bianchi. 2020b. The embeddings that came in from the cold: Improving vectors for new and rare products with content-based inference. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, pages 577–578. ACM.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Techcrunch. 2019a. Algolia finds $110m from accel and salesforce.

Techcrunch. 2019b. Coveo raises 227m at 1b valuation.

Techcrunch. 2019c. Lucidworks raises $100m to expand in ai finds.

Techcrunch. 2021. Bloomreach raises $150m on $900m valuation and acquires exponea.

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and G. Manso. 2020. The computational limits of deep learning. *ArXiv*, abs/2007.05558.

Arthur Toth, L. Tan, G. Fabbrizio, and Ankur Datta. 2017. Predicting shopping behavior with mixture of rnns. In *eCOM@SIGIR*.

Manos Tsagkias, Tracy Holloway King, Surya Kallumadi, Vanessa Murdock, and Maarten de Rijke. 2020. Challenges and research opportunities in ecommerce search and recommendations. In *SIGIR Forum*, volume 54.

U.S. Department of Commerce. 2020. U.s. census bureau news.

Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016a. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 225–232. ACM.

Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016b. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 225–232. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

David Vilares, Michalina Strzyz, Anders Søgaard, and Carlos G'omez-Rodr'iguez. 2020. Parsing as pretraining. *ArXiv*, abs/2002.01685.

Bo Yan, Krzysztof Janowicz, Gengchen Mai, and Song Gao. 2017. From itdl to place2vec: Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '17, New York, NY, USA. Association for Computing Machinery.

Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1469–1478. ACM.

Han Zhang, Songlin Wang, Kang Zhang, Zhiling Tang, Yunjiang Jiang, Yun Xiao, Weipeng Yan, and Wenyun Yang. 2020. Towards personalized and semantic retrieval: An end-to-end solution for e-commerce search via embedding learning. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2407–2416. ACM.

Xiaoting Zhao, Raphael Louca, Diane Hu, and Liangjie Hong. 2020. The difference between a click and a cart-add: Learning interaction-specific embeddings. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 454–460, New York, NY, USA. Association for Computing Machinery.

Zhen Zuo, L. Wang, Michinari Momma, W. Wang, Yikai Ni, Jianfeng Lin, and Y. Sun. 2020. A flexible large-scale similar product identification system in e-commerce.

A Visualization of Session Embeddings

Figures 3 to 6 represent browsing sessions projected in two-dimensions with t-SNE (van der Maaten and Hinton, 2008): for each browsing session, we retrieve the corresponding type (e.g. shoes, pants, etc.) of each product in the session, and use majority voting to assign the most frequent product type to the session. Hence, the dots are color-coded by product type and each dot represents a

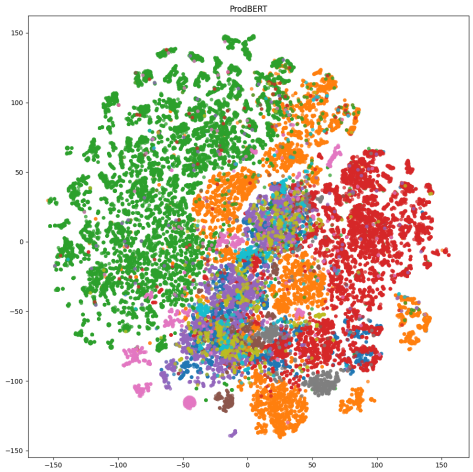

Figure 3: T-SNE plot of browsing session vector space from Shop A and built with the first hidden layer of pre-trained Prod2BERT.

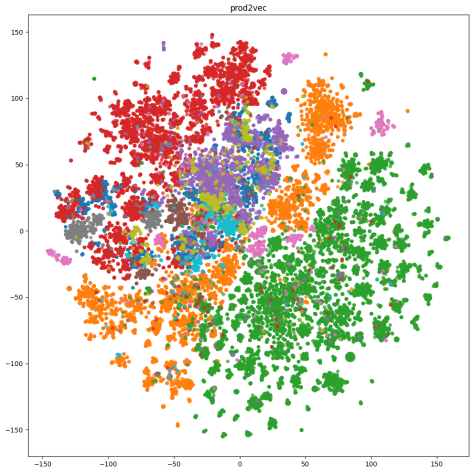

Figure 4: T-SNE plot of browsing session vector space from Shop A and built with *prod2vec* embeddings.

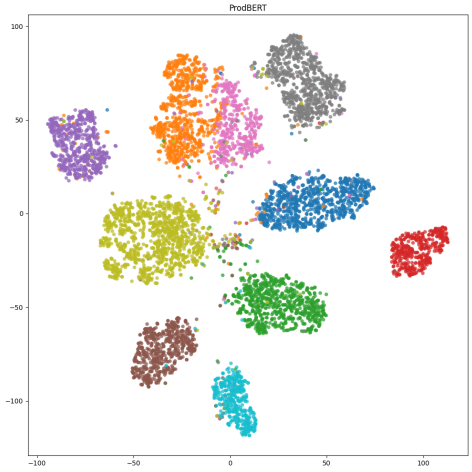

Figure 5: T-SNE plot of browsing session vector space from Shop B and built with the first hidden layer of pre-trained Prod2BERT.

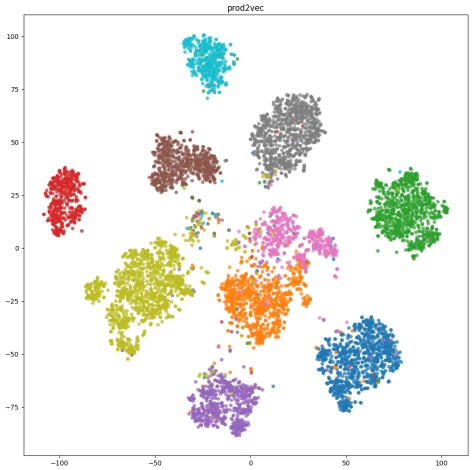

Figure 6: T-SNE plot of browsing session vector space from Shop B and built with *prod2vec* embeddings.

unique session from our logs. It is easy to notice that, first, both contextual and non-contextual embeddings built with a smaller amount of data, i.e. Figures 3 and 4 from Shop A, have a less clear separation between clusters; moreover, the quality of Prod2BERT seems even lower than *prod2vec*, as there exists a larger central area where all types are heavily overlapping. Second, comparing Figure 5 with Figure 6, both Prod2BERT and *prod2vec* improve, which confirms Prod2BERT, given enough pre-training data, is able to deliver better separations in terms of product types and more meaningful representations.