How Far is Too Far? Studying the Effects of Domain Discrepancy on Masked Language Models

Subhradeep Kayal, Alexander Rakhlin, Ali Dashti, Serguei Stepaniants
Amazon
{dkayal, rakhlina, dashtia, sergueis}@amazon.com

Abstract

Pre-trained masked language models, such as BERT, perform strongly on a wide variety of NLP tasks and have become ubiquitous in recent years. The typical way to use such models is to fine-tune them on downstream data. In this work, we aim to study how the difference in domains between the pre-trained model and the task effects its final performance. We first devise a simple mechanism to quantify the domain difference (using a cloze task) and use it to partition our dataset. Using these partitions of varying domain discrepancy, we focus on answering key questions around the impact of discrepancy on final performance, robustness to out-of-domain test-time examples and effect of domain-adaptive pre-training. We base our experiments on a large-scale openly available e-commerce dataset, and our findings suggest that in spite of pre-training the performance of BERT degrades on datasets with high domain discrepancy, especially in low resource cases. This effect is somewhat mitigated by continued pre-training for domain adaptation. Furthermore, the domain-gap also makes BERT sensitive to out-of-domain examples during inference, even in high resource tasks, and it is prudent to use as diverse a dataset as possible during fine-tuning to make it robust to domain shift.

Keywords: BERT, Pre-training, Domain Adaptation

1. Introduction and Prior Work

Starting with *BERT* (Devlin et al., 2019), the strong performance of language models have been attributed mainly to being *pre-trained* on substantially large uncurated unlabeled datasets, wherein they learn contextual representations in an unsupervised manner. These pre-trained language models or *PLMs* may be *fine-tuned* in a second training step on a labeled dataset to orient them towards particular downstream tasks.

In preceding years, BERT has been adapted to new and niche domains ranging from scientific (Beltagy et al., 2019) to biomedical text (Gu et al., 2021), with varying degrees of relative gain in performance. Recently, Aragón et al. (2023) proposed a two-stage domain adaptation of BERT to detect mental disorders based on social media posts, whose marked uplift over the original suggests the need for adaptation, especially for narrow domains. Apart from these, several works have explored inexpensive forms of domain adaptation, the common theme being the injection of missing vocabulary tokens into the modeling pipeline (Sachidananda et al., 2021; Poerner et al., 2020). Finally, there also exists work measuring the difference between domains of training and testing data using various divergence metrics (Ramesh Kashyap et al., 2021).

In our paper, we aim to provide empirical evidence of the limits of pre-trained masked language models, specifically BERT, when it comes to efficiency and applicability in a new domain. We quantify this *domain-gap* using a simple mechanism of inspecting reconstructed masked tokens from sentences of an out-of-domain dataset. We then partition the dataset from slightly to highly out-of-domain, and perform experiments on these partitions showing how the domain-gap affects fine-tuning performance and robustness, as well as whether domain-adaptive pre-training is able to mitigate it.

2. Methodology

2.1. BERT in brief

BERT (*Bidirectional Encoder Representations from Transformers*) combines the transformer architecture introduced by Vaswani et al. (2017) with the concept of pre-training and fine-tuning for maximum performance (Howard and Ruder, 2018). BERT is encoder only and bidirectional, with the self-attention mechanism having access to the information from the entire sequence, making it very suitable for classification and structured prediction tasks. BERT was trained using two objectives: a *masked language model* objective, where it learns to predict randomly masked tokens based on their context, and a next-sequence prediction objective, where the model is tasked to predict whether a sequence B would naturally follow the previous sequence A.

Why BERT? Numerous modifications have been made to BERT since its inception, such as cross-layer parameter sharing to reduce size (Lan et al., 2020), distillation (Sanh et al., 2019) or introducing additional optimization criteria (such

LREC-COLING 2024, pages 8192–8199

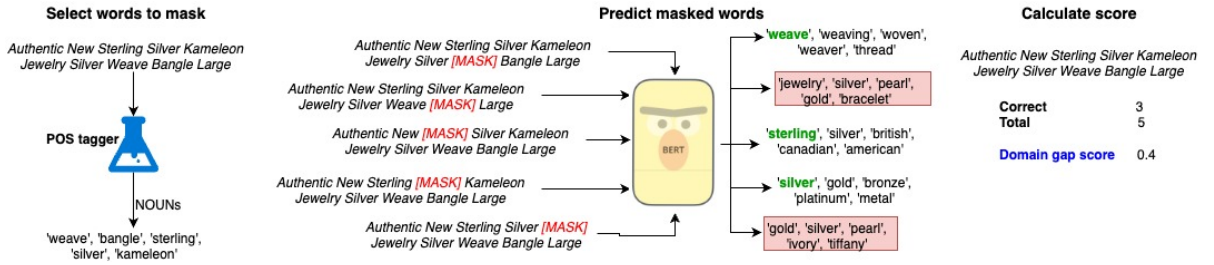

Figure 1: Overview of steps to calculate the *domain-gap score*. Words in green were masked and successfully predicted in the top-5 predictions, while red blocks indicate that the respective masked word (in this example, *bangle* and *kameleon*, could not be predicted.

as predicting the right order for shuffled words (Wang et al., 2020)).

Even so, in this work we focus on BERT because: (1) all of the aforementioned variants of BERT have been pre-trained on the same dataset, namely the English Wikipedia and the Bookcorpus dataset (Zhu et al., 2015). Therefore, it is our assumption that they will not behave significantly differently than BERT, in the context of their operating characteristics towards out-of-domain data. (2) BERT remains one of the most used workhorses for natural-language processing applications in English for classification and structured prediction, as evidenced by number of downloads from the model hub of the open-source community *Huggingface*[1234] at the time of writing this manuscript.

2.2. Masked-word Prediction as a Mechanism to Quantify domain-gap

We use the cloze test (Taylor, 1953) in order to quantify BERT's domain-gap with respect to a new data domain. A cloze test is an exercise in which one or more words in a piece of text is/are masked and the ask is to fill-in these masked parts of the text, and has also been termed as masked language modeling (Devlin et al., 2019) in recent NLP literature. The process we employ to calculate the *domain-gap score* is illustrated in Figure 1 and described next.

Step 1 - Choosing words to mask: In prior works for pre-training models, the masking of words is done randomly with a selected occlusion probability. However, this approach maybe prone to masking words related to general grammar (e.g., verbs, articles, adjectives, prepositions, etc.) rather than domain-specific vocabulary (e.g., nouns). Thus, we subject every piece of text in our dataset to a part-of-speech tagger to extract words which are tagged as *NOUN*. We construct sentences with masked words for every original sentence and every NOUN extracted from it.

Step 2 - Masked-word prediction: We use BERT to predict the top-5 most suitable tokens for every masked noun. Since an original sentence may have multiple nouns masked, there maybe multiple sets of top-5 predictions per original sentence.

Step 3 - Calculating per-sentence domain-gap scores and partitioning a dataset: Now, for every original sentence, we have masked nouns and predicted token sets. We calculate the number of correct predictions by simply checking whether the masked noun is a part of the corresponding top-5 predictions, with an exact match. Then the per-sentence domain-gap score is calculated as $1-(\#correct/\#masked)$. This score represents the inverse of the accuracy of the model in predicting a missing word, given the context, and thus can be used as a proxy for how out-of-domain a sentence is. We sort on the domain-gap score (which varies from 0-1, where 1 signifies the maximum domain-gap) and partition the dataset in question into four parts: *high domain-gap (DG=H)* (score 0.75-1), *medium (DG=M)* (score 0.5-0.75), *low (DG=L)* (score 0.25-0.5) and *none (DG=N)* (score 0-0.25).

Why not directly calculate the probability of a sentence? As an illustrative example, let us take a language model capturing bi-gram relationships. Given a sentence and this model, it is possible to calculate directly the probability of a sentence "occuring" with respect to the domain knowledge of the model. For instance, given a sentence *"he is going to school"*, the joint probability can be calculated using the chain rule: $P(he) * P(is|he) * P(going|is) * P(to|going) * P(school|to)$. This can, of course, be generalized to an *N-gram* case. However, as BERT (and the likes) are bidirectional in nature, the conditional token probabilities would take into account proceeding and pre-

[1] https://huggingface.co/models?pipeline_tag=question-answering&language=en&sort=downloads
[2] https://huggingface.co/models?pipeline_tag=text-classification&language=en&sort=downloads
[3] https://huggingface.co/models?pipeline_tag=fill-mask&language=en&sort=downloads
[4] https://huggingface.co/models?pipeline_tag=token-classification&language=en&sort=downloads

ceeding tokens, and the calculation of sentence probabilities as above would not be sensible.

2.3. Domain Adaptation by Continued Pre-training

In order to adapt BERT to the domain of our choice, we employ the process suggested by Howard and Ruder (2018), and Wolf et al. (2022), which is to continue training BERT on the new corpus of data in a self-supervised fashion using the masked language modeling loss. Instead of masking words at random, we mask the extracted nouns from each sentence, as described in Step 1 of Section 2.2, to bias the learning towards important words in the target domain. We update all of the weights of the model without freezing any layer.

3. Dataset and Task

Data: For our experiments, we use a dataset of Amazon product metadata (Ni et al., 2019) which has the titles, descriptions and categories for 15.5 million products from the e-commerce website amazon.com. The dataset is minimally cleaned such that we: (1) rid the titles and descriptions of stray HTML tags, (2) concatenate the title and description into one string column, (3) ignore short entries less than 30 characters long, and (4) ignore all categories which have less than 0.1% of the total items attributed them. Furthermore, after the extraction of noun words to mask (as described in the Step 1 of Section 2.2), we also ignore entries where there are no extracted nouns. After these cleanup steps, we are left with 13 million datapoints.

Task Selection: We choose the prediction of product category, given the title and description, as our downstream task for all our experiments.

4. Experiments and Results

We aim to answer three main questions in this paper; *RQ1*: How does the domain-gap between BERT and the downstream data impact the performance on the downstream task?, *RQ2*: How robust is a fine-tuned model to out-of-domain data during test-time? and *RQ3*: Does domain adaptation help with the robustness of BERT?

4.1. Setup

Implementation details: We use an open-source POS tagger from *SpaCy*[5] for NOUN extraction, and BERT from Huggingface[6].

Infrastructure details: All experiments were performed on a *g5.4xlarge* instance[7] having an NVIDIA A10G GPU with 24 GB of memory and 16 virtual CPUs with 64 GB of RAM.

Hyperparameter details: The tokenization is done to a *max length* of 250 with truncation or padding (as suitable). For fine-tuning BERT, 30% of the training data is held-out for validation and the model trained for 20 *epochs* to minimize the *cross-entropy loss*, such that the best model is saved based on the validation *macro F1-score*. The *batch-size* is 64, and the emphAdamW optimizer (Loshchilov and Hutter, 2019) is used with an initial *learning rate* of 10^{-5}. For continued pre-training, we convert the training dataset into equal *chunks* of size 128; other hyperparameters are identical to the ones for fine-tuning, except the batch-size which is larger and set to 128.

Data details: Recall that we partition the Amazon product dataset into four parts (see Section 2.2). For our experiments, we construct equal partitions by randomly subsampling 500,000 samples from each part, thus controlling for imbalance.

4.2. Results

RQ1: Effect of domain-gap on downstream tasks We fine-tune and test BERT on data *within* each partition, splitting into 50-50 for training-testing, repeating experiments for 3 different splits and reporting average results. For ablation, we fine-tune the model on 1%, 5%, 10%, 25%, 50% and 100% of the training split, while testing on the full test split.

We observe (in Table 1) the classification performance benefits significantly from the addition of data. For example, in the case of the lowest domain-gap (DG=N), the F1-score increases by 40% relative to the original value (from 0.46 to 0.644) as the size of the data increases 10-times (from 1% to 10%), and again by 25% for the next 10-fold increase. The relative performance gains slow as more data is added. As for the domain-gap, it has a significant adverse effect on the downstream performance of the model, especially in low-resource cases. For the extreme low-resource case where the model was trained only on 1% available data, the performance is 29.3% worse (relative) in case where the gap is highest (DG=H), versus by only 2% where it is lowest (DG=N). Even for more realistic low-resource cases (10% or 25%), the perfomance gap is significant for the high-domain-gap dataset (17% and 14% respectively).

In summary, we learn that although addition of data is helpful, it cannot fully counter the effects of the gap between a pretrained model and the domain of the downstream task, and performance degradation may be expected in such a case.

[5] https://spacy.io/usage/linguistic-features#pos-tagging

[6] https://huggingface.co/bert-base-uncased

[7] https://aws.amazon.com/ec2/instance-types/g5/

%train / dataset	1	5	10	25	50	100
DG=N	**0.460**	**0.639**	**0.644**	**0.749**	**0.763**	**0.807**
DG=L	0.450 (↓-2.17%)	0.626 (↓-2.03%)	0.634 (↓-1.55%)	0.736 (↓-1.73%)	0.750 (↓-1.70%)	0.802 (↓-0.62%)
DG=M	0.387 (↓-15.86%)	0.585 (↓-8.45%)	0.592 (↓-8.07%)	0.702 (↓-6.27%)	0.722 (↓-5.37%)	0.772 (↓-4.33%)
DG=H	0.325 (↓-29.34%)	0.518 (↓-18.93%)	0.533 (↓-17.23%)	0.643 (↓-14.15%)	0.679 (↓-11%)	0.742 (↓-8.05%)

Table 1: Effect of domain-gap on downstream performance. In this table we show the effect of fine-tuning BERT on the individual partitions of data for different training data sizes. The reported figures are F1-scores and the percentages in brackets quantify relative performance drop, from the maximum score in **bold**, as the domain-gap grows (interpret column-wise).

%train	Low-resource (25% data)				High-resource (100% data)			
train on / test on	DG=N	DG=L	DG=M	DG=H	DG=N	DG=L	DG=M	DG=H
DG=N	**0.749**	0.684 (↓-8.67%)	0.625 (↓-16.55%)	0.537 (↓-28.30%)	**0.807**	0.761 (↓-5.70%)	0.729468 (↓-9.60%)	0.649 (↓-19.57%)
DG=L	0.684 (↓-7.06%)	**0.736**	0.654 (↓-11.14%)	0.564 (↓-23.36%)	0.777 (↓-3.11%)	**0.802**	0.757 (↓-5.61%)	0.678 (↓-15.46%)
DG=M	0.677 (↓-3.56%)	0.701 (↓-0.14%)	**0.702**	0.608 (↓-13.39%)	0.762 (↓-1.29%)	0.766 (↓-0.77%)	**0.772**	0.711 (↓-7.90%)
DG=H	0.631 (↓-1.86%)	0.636 (↓-1.08%)	0.632 (↓-1.71%)	**0.643**	0.731 (↓-1.48%)	0.735 (↓-0.94%)	0.738 (↓-0.54%)	**0.742**

Table 2: Robustness to out-of-domain test-time data. Here, we test the fine-tuned model on other partitions than that it was trained on. Numbers in **bold** represent maximum performance when the train/test domains are the same and the percentages quantify relative performance drop from this maximum (interpret row-wise).

%train of DG=N	Low-resource (25% data)				High-resource (100% data)			
Test on	DG=N	DG=L	DG=M	DG=H	DG=N	DG=L	DG=M	DG=H
product-BERT	**0.774** (↑**+3.33%**)	0.709 (↓-8.39%) (↑**+3.65%**)	0.648 (↓-16.27%) (↑**+3.68%**)	0.563 (↓-27.26%) (↑**+4.84%**)	**0.812** (↑**+0.62%**)	0.767 (↓-5.54%) (↑**+0.78%**)	0.735 (↓-9.48%) (↑**+0.82%**)	0.659 (↓-18.84%) (↑**+1.54%**)

Table 3: Effect of domain-adaptive pre-training. We report the performance of *product-BERT*, which was subjected to domain adaptation and then fine-tuned on DG=N. This row can be compared to row 1 in Table 2. The first set of percentages with ↓ signify the performance drop from the maximum (in **bold**), while the second set with ↑ represent the performance gain over vanilla BERT.

RQ2: Robustness to out-of-domain data at test-time In this case, we fine-tune the model on one partition and test on another, to illustrate the out-of-domain robustness. Recall that for RQ1, we made a 50-50 train-test split in every partition. We use the same splits for RQ2, which makes the F1-scores comparable across the two research questions. In RQ2, we focus on a low-resource case (use 25% of the training data) and a high-resource case (100%).

We observe (in Table 2) that the robustness of the model benefits slightly from the addition of more data: a model trained on DG=N and tested on DG=H shows a performance deficit of 19.5% relative to its original efficiency, when 4-fold more data is added, versus 28.3% in the low-resource case. BERT does not benefit significantly from being fine-tuned on similar data, even when the amount of data increases manyfold, and is sensitive to test-time domain-gap.

Secondly, we observe that the model benefits from being fine-tuned on diverse data; the more diverse the data, the more robust it is to out-of-domain examples during testing. For example, considering the high-resource setting, a model fine-tuned with similar data is expected to have degradation from 5.7% (relative) to 19.5% depending on how out-of-domain a dataset is during test-time. The magni-

tude of this degradation keeps decreasing as the model is fine-tuned on more diverse data. BERT trained on DG=H shows little degradation (maximum of 1.5%) when tested across all other partitions.

RQ3: Effect of continued pre-training with out-of-domain data To answer the final RQ, we continue training BERT using the masked language modeling loss, as described in Section 2.3. The steps are as follows: (1) continue (pre-)training BERT on the data corresponding to partitions DG=L, DG=M and DG=H; we call this model *product-BERT* (2) fine-tune product-BERT on DG=N and test on all partitions, following the protocol described previously.

We observe by comparing the results in Table 3 to those of the first row in Table 2 that pre-training has significant benefits in the low-resource case (as much as 4.8% relative improvement), but less so when there is enough data for fine-tuning for the high-resource case. Interestingly, even though the DG=N partition was not a part of the pre-training data, we still observe a lift of 3.3% in the case where the model was tested on it. However, challenges around robustness still remain.

5. Conclusions and Future Work

We investigate aspects related to the domain adaptation of PLMs; our model is BERT and the domain, e-commerce product metadata. Employing a cloze task we create a simple scoring mechanism, which we call *domain-gap score*, to quantify how out-of-domain a sentence is. Using this score, we create partitions in the dataset corresponding to their domain-gap and use them for experimentation, the findings of which are stated next.

Firstly, we find that the domain-gap has a notable adverse effect on downstream model performance, particularly in low-resource scenarios. Secondly, with respect to the model's robustness to out-of-domain data, we learn that it is better to fine-tune with data as diverse as possible from the model's original domain. Finally, we observe that pre-training provides substantial benefits in low-resource scenarios, but less so in high-resource settings. In summary, our work underscores the importance of carefully considering domain-gaps and diversity of training data in pre-training or fine-tuning BERT-like models for downstream tasks.

As for future work, two potential avenues maybe suggested to further explore the effects domain-gap: (1) replicating our experiments with datasets from other domains (e.g., culinary data) and (2) testing with downstream tasks other than classification, such as entity recognition or question answering from text.

6. Bibliographical References

Mario Aragón, Adrian Pastor Lopez Monroy, Luis Gonzalez, David E. Losada, and Manuel Montes. 2023. DisorBERT: A double domain adaptation model for detecting signs of mental disorders in social media. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15305–15318, Toronto, Canada. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

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.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR*

2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. Inexpensive domain adaptation of pretrained language models: Case studies on biomedical NER and covid-19 QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1482–1490, Online. Association for Computational Linguistics.

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2021. Domain divergences: A survey and empirical analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1830–1849, Online. Association for Computational Linguistics.

Vin Sachidananda, Jason Kessler, and Yi-An Lai. 2021. Efficient domain adaptation of language models via adaptive tokenization. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 155–165, Virtual. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Wilson L. Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. Structbert: Incorporating language structures into pre-training for deep language understanding. In *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2022. Fine-tuning a masked language model.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

Appendices

Appendix A: Data Distributions

The Amazon product metadata dataset consists of metadata and reviews for 15.5 million products. Products are attributed to categories such as *Clothing, Shoes & Jewelry* or *Books*. The distribution of these categories are shown in Figure 2.

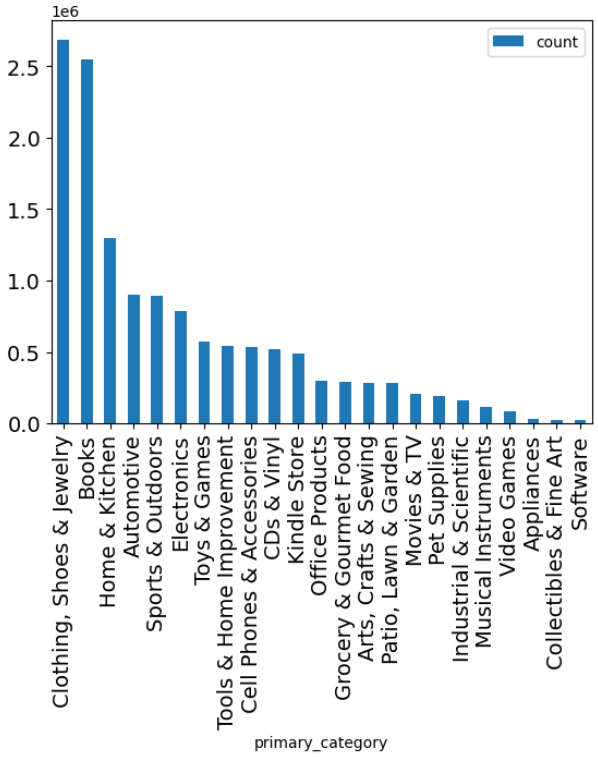

Figure 2: Distribution of product categories in the Amazon dataset

Appendix B: Distribution of Domain-Gap Scores

In Section 2.2, we outline the simple process we follow to utilize the cloze task to create a sentence level score (*domain-gap score*) to assess how out-of-domain with respect to BERT's original training data. We plot the histogram of these scores in Figure 3. We can see that majority of the dataset is in-domain. Using these scores, we fabricate four sub-datasets signifying different domain levels of domain-gap. The distribution of these four parts are shown in Table 4.

Dataset	DG=H	DG=M	DG=L	DG=N
Size	736259	2086257	3691766	6562251

Table 4: Size of the dataset partitions according to the domain-gap scores, from high (i.e., signifying that BERT is highly misaligned to the data partition) to none.

Appendix C: Examples of masked word reconstruction

Tables 5 and 6 provide examples of how BERT fills masked nouns. The first table shows cases where BERT succeed (low domain-gap) and the second where it fails (high domain-gap).

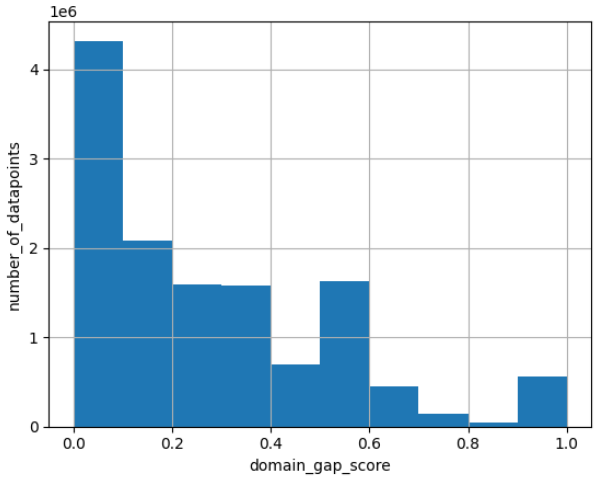

Figure 3: Distribution of domain-gap scores for BERT

Description	**Masked words**	**Top-5 predictions per masked word**
P&B I don't Sweat I Sparkle Men's T-shirt	men sweat shirt	men, women, man, ##men, girls sweat, ##g, ##j, ##f, . shirt, shirts, blouse, suit, tie
Anne Klein Beacon Gold-Tone Circular Collar Necklace	necklace collar gold tone	necklace, ., -, pendant, earrings collar, neck, necklace, chain, pendant gold, silver, golden, bronze, copper tone, tones, toned, color, style
Orange and Blue Footie Sock with Green Trim	orange sock	orange, red, yellow, green, white sock, socks, shoe, boot, shoes
Invicta Men's 1798 Russian Diver Collection Multi-Function Watch	men multi diver watch collection	men, women, woman, man, ##men multi, single, dual, multiple, tri diver, divers, diving, dive, sailor ., ;, !, watch, | collection, collections, collecting, collector, collectors
U.S. Polo Assn. Little Boys' Short Sleeve Solid with Contrast Rib and Collar	contrast sleeve rib collar boys	front, low, back, lower, contrast sleeve, sleeves, jacket, skirt, shirt rib, ribs, bar, flap, bone collar, jacket, sleeve, sleeves, heel boys, girls, lads, kids, guys

Table 5: Examples of descriptions from the *Clothing, Shoes & Jewelry* with *domain-gap score* as 0 where BERT correctly predicted all the masked tokens.

Description	**Masked words**	**Top-5 predictions per masked word**
Michael Antonio Women's Thames CHN Sandal	thames sandal	team, volleyball, club, basketball, hockey -, ., victoria, gold, london
Augusta Sportswear mens Medalist Jacket	medalist	sports, cycling, sport, wear, .
Kendra Scott Abalone Iridescent Drusy Deva Earrings	iridescent drusy earrings	and, silver, a, pink, blue diamond, crystal, silver, pearl, and ., &, ring, !, |
Roaman's Osana Sandal by Comfortview (Cognac,8 M)	sandal osana	sand, honey, white, coffee, gold indian, rainbow, tiger, golden, white
Stuart Weitzman Womens Closed Stuart Weitzman Womens Closed	womens	college, library, hospital, university, bank

Table 6: Examples of descriptions from the *Clothing, Shoes & Jewelry*with *domain-gap score* as 1 where BERT incorrectly predicted all the masked tokens.