Contextual BERT: Conditioning the Language Model Using a Global State

Timo I. Denk
Zalando SE
Berlin, Germany
timo.denk@zalando.de

Ana Peleteiro Ramallo
Zalando SE
Berlin, Germany
ana.peleteiro.ramallo@zalando.de

Abstract

BERT is a popular language model whose main pre-training task is to fill in the blank, i.e., predicting a word that was masked out of a sentence, based on the remaining words. In some applications, however, having an additional context can help the model make the right prediction, e.g., by taking the domain or the time of writing into account. This motivates us to advance the BERT architecture by adding a global state for conditioning on a fixed-sized context. We present our two novel approaches and apply them to an industry use-case, where we complete fashion outfits with missing articles, conditioned on a specific customer. An experimental comparison to other methods from the literature shows that our methods improve personalization significantly.

1 Introduction

Since its publication, the BERT model by Devlin et al. (2019) has enjoyed great popularity in the natural language processing (NLP) community. To apply the model to a specific problem, it is commonly pre-trained on large amounts of unlabeled data, and subsequently fine-tuned on a target task. During both stages, the model's only input is a variably-sized sequence of words.

There are use-cases, however, where having an additional context can help the model. Consider a query intent classifier whose sole input is a user's text query. Under the assumption that users from different age groups and professions express the same intent in different ways, the classifier would benefit from having access to that user context in addition to the query. Alternatively, one might consider training multiple models on separate, age group- and profession-specific samples. However, this approach does not scale well, requires more training data, and does not share knowledge between the models.

To the best of our knowledge, there is a shortcoming in effective methods for conditioning BERT on a fixed-sized context. Motivated by this, and inspired by the graph-networks perspective on self-attention models (Battaglia et al., 2018), we advance BERT's architecture by adding a global state that enables conditioning. With our proposed methods [GS] and [GSU], we combine two previously independent streams of work. The first is centered around the idea of explicitly adding a global state to BERT, albeit without using it for conditioning. The second is focused on injecting additional knowledge into the BERT model. By using a global state for conditioning, we enable the application of BERT in a range of use-cases that require the model to make context-based predictions.

We use the outfit completion problem to test the performance of our new methods: The model predicts fashion items to complete an outfit and has to account for both style coherence and personalization. For the latter, we condition on a fixed-sized customer representation containing information such as customer age, style preferences, hair color, and body type. We compare our methods against two others from the literature and observe that ours are able to provide more personalized predictions.

2 Related Work

BERT's Global State In the original BERT paper, Devlin et al. (2019) use a `[CLS]` token which is prepended to the input sequence (e.g., a sentence of natural language). The assumption is that the model

Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs), pages 46–50
Barcelona, Spain (Online), December 13, 2020

aggregates sentence-wide, global knowledge at the position of the [CLS] token. This intuition was confirmed through attention score analysis (Clark et al., 2019), however, the BERT architecture does not have an inductive bias that aids it. Recent work therefore treats the [CLS] token differently. Zaheer et al. (2020) constrain their BERT variant Big Bird to local attention only, with the exception that every position may always attend to [CLS] regardless of its spatial proximity.

Ke et al. (2020) also observe that the [CLS] attention exhibits peculiar patterns. This motivates them to introduce a separate set of weights for attending to and from [CLS]. The authors thereby explicitly encode into the architecture that the sequence's first position has a special role and different modality than the other positions. The result is an increased performance on downstream GLUE tasks.

It is important to note that all related work on BERT's global state does not use the global state for conditioning. Instead, the architectural changes are solely being introduced to improve the performance on non-contextual NLP benchmarks.

Conditioning on a Context To the best of our knowledge, Wu et al. (2018) are the first to provide sentence-wide information to the model to ease the masked language model (MLM) pre-training task. The authors inject the target label (e.g., positive or negative review) of sentiment data by adding it to the [CLS] token embedding. In a similar application, Li et al. (2020) process the context separately and subsequently combine it with the model output to make a sentiment prediction.

Xia et al. (2020) condition on richer information, namely an *intent*, which can be thought of as a task descriptor given to the model. The intent is represented in text form, is variably sized, and prepended to the sequence. This is very similar to a wide range of GPT (Radford et al., 2019) applications.

Chen et al. (2019) condition on a customer's variably-sized click history using a Transformer (Vaswani et al., 2017). The most similar to our work are Wu et al. (2020) who personalize by concatenating every position in the input sequence with a user embedding – method [C] from Section 3. Their approach, however, lacks an architectural bias that makes the model treat the user embedding as global information.

BERT as a Graph Neural Network (GNN) Battaglia et al. (2018) introduce a framework that unites several lines of research on GNNs. In the Appendix, the authors show that – within their framework – the Transformer architecture is a type of GNN; Joshi (2020) supports this finding. In both cases the observation is that a sentence can be seen as a graph, where words correspond to nodes and the computation of an attention score is the assignment of a weight to an edge between two words.

In the GNN framework, a global state is accessible from every *transfer function* and can be individually updated from layer to layer.[1] Neither Transformer nor BERT, however, have a global state in that sense. Inspired by this observation, we introduce a global state and use it for conditioning. We explain our two novel methods in the following section, alongside with two that are derived from the literature.

3 Conditioning BERT With a Global State

Let $\boldsymbol{w}$ denote a sequence of n words $w_i \in \mathbb{V}$ from a fixed-sized vocabulary $\mathbb{V}$. Further, let $\boldsymbol{w}_{-i}$ be the sequence without the ith word. Recall that a vanilla BERT model (Devlin et al., 2019) can predict the probability $\Pr(M = w_i \mid \boldsymbol{w}_{-i})$ that a word w_i is masked-out in a sequence ($M = w_i$ being the masking event), conditioned on the other words in the sequence. Next, we introduce four methods to additionally condition BERT on a context vector $\boldsymbol{c} \in \mathbb{R}^{d_{\text{context}}}$, which allows it to predict $\Pr(M = w_i \mid \boldsymbol{w}_{-i}, \boldsymbol{c})$.

Concat [C] Similar to Wu et al. (2020), we concatenate the context vector with every position in the input sequence. Let $\boldsymbol{x}_i \in \mathbb{R}^{d_{\text{model}}}$ denote the embedding of word w_i at position i. The resulting input matrix is $\boldsymbol{I}_{[\text{C}]} := \boldsymbol{W}\begin{bmatrix}\boldsymbol{x}_1 & \dots & \boldsymbol{x}_n \\ \boldsymbol{c} & \dots & \boldsymbol{c}\end{bmatrix}$. $\boldsymbol{W} \in \mathbb{R}^{d_{\text{model}} \times (d_{\text{model}} + d_{\text{context}})}$ is a trainable weight matrix that reduces the input dimensionality.

New Position [NP] This method adds a new position to the input sequence at which the context is stored. It is comparable to how Wu et al. (2018) add label information. Instead of feeding the word sequence $\begin{bmatrix}\boldsymbol{x}_1 & \dots & \boldsymbol{x}_n\end{bmatrix}$ into the model, we prepend the transformed context $\boldsymbol{W}\boldsymbol{c}$ to the sequence, where

[1] For details on the GNN definition of a global state we refer the reader to Section 3.2 in Battaglia et al. (2018).

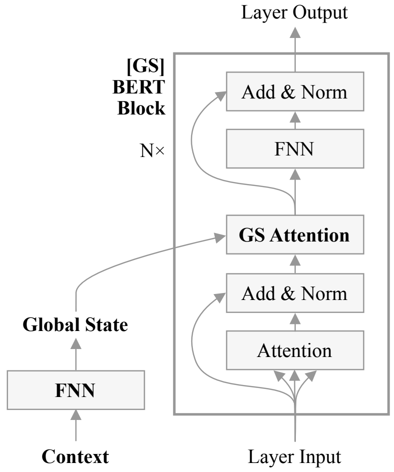

Figure 1: A BERT block with global state. Boxes have learned parameters; our additions are bold.

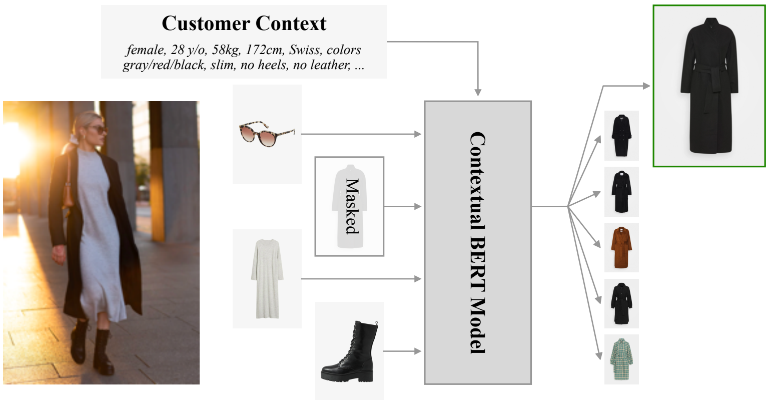
Customer Context
female, 28 y/o, 58kg, 172cm, Swiss, colors gray/red/black, slim, no heels, no leather, ...
Masked
Contextual BERT Model

Figure 2: The fill-in-the-blank task on fashion outfits. Given a set of articles (left-hand side) and customer context, the model makes several predictions (right-hand side) for a masked out item (here: the coat). The predictions are personalized, because the model is utilizing the customer context.

$\boldsymbol{W} \in \mathbb{R}^{d_\text{model} \times d_\text{context}}$ is a trainable weight matrix. The resulting input is $\boldsymbol{I}_\text{[NP]} := \begin{bmatrix}\boldsymbol{W}\boldsymbol{c} & \boldsymbol{x}_1 & \dots & \boldsymbol{x}_n\end{bmatrix}$. The model's attention masks are adjusted, such that every position can attend to the new first position.

Global State [GS] Our method is inspired by the GNN perspective on BERT. Its implementation is similar to the way the Transformer (Vaswani et al., 2017) decoder attends to the encoder. [GS] treats the context as a read-only global state from which the internal representations can be updated. In order to adjust the architecture of BERT accordingly, we insert a *global state attention layer* between the intra-sequence attention and the (originally) subsequent feed-forward neural network (FNN). Figure 1 shows how the inserted elements fit into the vanilla BERT block.

More formally, let $\boldsymbol{X}^{(l)} \in \mathbb{R}^{n \times d_\text{model}}$ be the output of the lth BERT block (of which there are N); let $\boldsymbol{X}^{(0)} := \begin{bmatrix}\boldsymbol{x}_1 & \dots & \boldsymbol{x}_n\end{bmatrix}$ be the model input; and let $\tilde{\boldsymbol{c}} := \text{FNN}(\boldsymbol{c})$ be the global state derived from the context vector using a non-linear transformation $\text{FNN}(\boldsymbol{x}) := \boldsymbol{W}_2 \max(0, \boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2$. With our modification, $\boldsymbol{X}^{(l)}$ is defined by first performing the normal intra-sequence attention as in BERT $\boldsymbol{A} := \text{Attention}\left(\boldsymbol{W}^\text{Q}\boldsymbol{X}^{(l-1)}, \boldsymbol{W}^\text{K}\boldsymbol{X}^{(l-1)}, \boldsymbol{W}^\text{V}\boldsymbol{X}^{(l-1)}\right)$; multi-head attention can be used here as well. Then, also unchanged, $\hat{\boldsymbol{A}} := \text{LayerNorm}\left(\text{Dropout}(\boldsymbol{A}) + \boldsymbol{X}^{(l-1)}\right)$. The internal representation is then updated once more by reading[2] from the global state $\tilde{\boldsymbol{c}}$ with

$$\boldsymbol{B} := \text{Attention}\Big(\boldsymbol{V}^\text{Q}\hat{\boldsymbol{A}}, \boldsymbol{V}^\text{K}\left[\tilde{\boldsymbol{c}}\right], \boldsymbol{V}^\text{V}\left[\tilde{\boldsymbol{c}}\right]\Big) = \boldsymbol{V}^\text{V}\left[\tilde{\boldsymbol{c}}\right], \tag{1}$$

and computing $\hat{\boldsymbol{B}} := \text{LayerNorm}\Big(\text{Dropout}(\boldsymbol{B}) + \hat{\boldsymbol{A}}\Big)$. Lastly, the BERT layer output is computed as $\boldsymbol{X}^{(l)} := \text{LayerNorm}\Big(\text{FNN}\Big(\hat{\boldsymbol{B}}\Big) + \hat{\boldsymbol{B}}\Big)$. The definitions of $\text{Attention}(\cdot)$, $\text{LayerNorm}(\cdot)$, and $\text{Dropout}(\cdot)$ are identical to Vaswani et al. (2017). The weight matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ are not shared between layers. The layer indices $^{(l)}$ of $\boldsymbol{A}$, $\boldsymbol{B}$, and the weight matrices are omitted to aid readability.

Global State With Update [GSU] Note that in the [GS] method the global state $\tilde{\boldsymbol{c}}$, which is being added to the internal representation in Equation 1, is the same for all blocks. With [GSU], a global state transfer function $\tilde{\boldsymbol{c}}^{(l+1)} := \text{FNN}\left(\tilde{\boldsymbol{c}}^{(l)}\right)$ updates the global state, with its initial value being derived from the context: $\tilde{\boldsymbol{c}}^{(1)} := \text{FNN}(\boldsymbol{c})$. The weights of FNN are not shared between layers. Equation 1 is updated to read from the global state belonging to the lth layer:

$$\boldsymbol{B}^{(l)}_\text{[GSU]} := \boldsymbol{V}^\text{V}\left[\tilde{\boldsymbol{c}}^{(l)}\right]. \tag{2}$$

[2] Note that the global state *sequence* $\left[\tilde{\boldsymbol{c}}\right]$ does not admit for attention in the sense of selecting a weighted average of multiple vectors, because it consists only of a single vector. Instead, Equation 1 is reduced to the possibility for the lth layer to transform the context and update its internal state based on it.

Method	Cross-entropy loss	Recall@1	Recall@5	Recall@250	Parameters
[None]	5.2636 ± 0.0038	$8.53\% \pm 0.06$	$21.44\% \pm 0.04$	$87.14\% \pm 0.09$	546 432
[C]	4.9428 ± 0.0047	$10.26\% \pm 0.16$	$25.75\% \pm 0.09$	$90.76\% \pm 0.10$	673 664
[NP]	4.9260 ± 0.0078	$10.53\% \pm 0.06$	$26.27\% \pm 0.25$	$90.80\% \pm 0.11$	640 768
[GS]	4.8542 ± 0.0084	$11.19\% \pm 0.06$	$27.56\% \pm 0.28$	$91.35\% \pm 0.13$	723 328
[GSU]	4.7459 ± 0.0043	$12.21\% \pm 0.17$	$29.40\% \pm 0.06$	$92.19\% \pm 0.08$	921 856

Table 1: Comparison of the different conditioning methods on validation data. [None] provides no customer context to the model. Values are means across three training runs reported with standard error.

4 Empirical Evaluation and Discussion

We evaluate the performance of our proposed methods on a real-world industry problem: personalized fashion outfit completion (see Figure 2) for Europe's largest fashion platform. Our proprietary dataset consists of 380k outfits, created by professional stylists for individual customers. When styling a customer, i.e., putting together an outfit, the stylist has access to all customer features that we later use to condition the model. Therefore, customer data and outfit are statistically dependent.

The customer features are individually embedded using trainable, randomly initialized embedding spaces. The per-feature embedding vectors are subsequently concatenated, yielding a context vector of $d_{\text{context}} = 736$. Features include the customer's age, gender, country, preferred brands/colors/styles, no-go types, clothing sizes, price preferences, and the occasion for which the outfit is needed. The second model input, namely the outfit itself, is constructed from learned embeddings for every individual fashion article, with $d_{\text{model}} = 128$. We stack $N = 4$ BERT blocks with multi-head attention (eight heads). For a masked-out item, the model predicts a probability distribution over $|\mathbb{V}| = 30\,000$ articles.

While not being an NLP dataset, our data resembles many of the important traits of a textual corpus: the vocabulary size is comparable to the one of word-piece vocabularies commonly used with BERT models. Fashion outfits are similar to sentences in that some articles appear often together (match style-wise) and others do not. Different is the typical sequence length which ranges from four to eight fashion articles, with an average length of exactly five. In contrast to sentences, outfits do not have an inherent order. To account for that we remove the positional encoding from BERT so it treats its input as a set.

Table 1 shows the results of evaluating the four different methods. We compare cross-entropy and recall@rank (r@r for short) on a randomly selected validation dataset consisting of 17k outfits that are held-out during training. The r@r is defined as the percentage of cases in which the masked-out item is among the top-r most probable items, according to the model. Model parameters are counted without embedding spaces for customer features and the last dense layer.

The empirical evaluation reveals the effectiveness of using a context for making predictions. The model's ability to replicate the stylist behavior better, i.e., achieve a higher r@r, improves substantially with the addition of a context. On r@1 we see a relative improvement of $+43\%$ by using the [GSU] method for conditioning over using no customer context at all ([None]) and $+16\%$ compared to [NP] (the best method without global state).

A comparison of the four different conditioning methods shows [GSU] to be most effective, followed by [GS], [NP], and [C]. The methods [C] and [NP] do not have any bias towards treating the context vector specially. They attend to other positions in the sequence the same way they attend to the context. The superiority of [GS] and [GSU] can presumably be explained by their explicit architectural ability to retrieve information from the global state and therefore effectively utilize the context for their prediction.

We acknowledge the differences between our outfits dataset and typical NLP benchmarks. Nonetheless we hypothesize that the effectiveness of our method translates to NLP. In particular when applied to use-cases in which the modality of context and sequence differ, e.g., for contexts comprised of numerical or categorical meta data about the text. That is because the model's freedom to read from the context separately allows it to process the different modalities of context and input sequence adequately.

5 Conclusions and Future Work

With Contextual BERT, we presented novel ways of conditioning the BERT model. The strong performance on a real-world use-case provides evidence for the superiority of using a global state to inject context into the Transformer-based architecture. Our proposal enables the effective conditioning of BERT, potentially leading to improvements in a range of applications where contextual information is relevant.

A promising idea for follow-up work is to allow for information to flow from the sequence to the global state. Further, it would be desirable to establish a contextual NLP benchmark for the research community to compete on. This benchmark would task competitors with contextualized NLP problems, e.g., social media platform-dependent text generation or named entity recognition for multiple domains.

References

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261.

Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. POG: personalized outfit generation for fashion recommendation at alibaba ifashion. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2662–2670. ACM.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Chaitanya Joshi. 2020. Transformers are graph neural networks. *The Gradient*.

Guolin Ke, Di He, and Tie-Yan Liu. 2020. Rethinking positional encoding in language pre-training. *ArXiv*, abs/2006.15595.

Xinlong Li, Xingyu Fu, Guangluan Xu, Yang Yang, Jiuniu Wang, Li Jin, Qing Liu, and Tianyuan Xiang. 2020. Enhancing BERT representation with context-aware embedding for aspect-based sentiment analysis. *IEEE Access*, 8:46868–46876.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

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 Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2018. Conditional BERT contextual augmentation. *CoRR*, abs/1812.06705.

Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential recommendation via personalized transformer.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip S. Yu. 2020. CG-BERT: conditional text generation with BERT for generalized few-shot intent detection. *CoRR*, abs/2004.01881.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.