Large-Scale Transfer Learning for Natural Language Generation

Sergey Golovanov [*1], Rauf Kurbanov [*1], Sergey Nikolenko [*12],
Kyryl Truskovskyi [*1], Alexander Tselousov [*], and Thomas Wolf [*3]
[1]Neuromation OU, Liivalaia tn 45, 10145 Tallinn, Estonia
[2]Steklov Mathematical Institute at St. Petersburg,
nab. r. Fontanki 27, St. Petersburg 191023, Russia
[3]Huggingface Inc., 81 Prospect St. Brooklyn, New York 11201, USA
sergey.golovanov@neuromation.io, rauf.kurbanov@neuromation.io,
snikolenko@neuromation.io, kyryl@neuromation.io,
al.tselousov@gmail.com, thomas@huggingface.co
[*]All authors contributed equally, names in alphabetical order.

Abstract

Large-scale pretrained language models define state of the art in natural language processing, achieving outstanding performance on a variety of tasks. We study how these architectures can be applied and adapted for natural language generation, comparing a number of architectural and training schemes. We focus in particular on open-domain dialog as a typical high entropy generation task, presenting and comparing different architectures for adapting pretrained models with state of the art results.

1 Introduction

Over the past few years, the field of natural language processing (NLP) has witnessed the emergence of transfer learning methods which have significantly improved the state of the art (Dai and Le, 2015; Peters et al., 2018; Howard and Ruder, 2018; Radford et al., 2018; Devlin et al., 2018). These methods depart from classical supervised machine learning where a predictive model for a given task is trained in isolation on a single dataset. Here, a model is pretrained on large text corpora and then fine-tuned on the target task. Such models are usually evaluated on natural language understanding (NLU) tasks such as text classification or question answering (Wang et al.; Rajpurkar et al., 2016), but natural language generation (NLG) tasks such as summarization, dialog, or machine translation remain relatively underexplored. At first glance, large-scale pretrained models appear to be a natural fit for NLG since their pretraining objectives are often derived from language modeling. However, interesting questions and problems still arise.

We consider a text-only NLG task where the generation of an output sequence of symbols $\mathbf{y} = (y_1, \ldots, y_m)$ is conditioned on a context $\mathbf{X} = (\mathbf{x}^1, \ldots, \mathbf{x}^K)$ composed of one or several sequences of symbols $\mathbf{x}^k = (x_1^k, \ldots, x_n^k)$. Several *types* of contexts may warrant different treatment in the model. E.g., in case of dialog generation they may include: (i) *facts* from a knowledge base, (ii) *dialog history*, and (iii) *the sequence of already generated output tokens* $(y_1, \ldots, y_{m-1})$. Thus, there arises a general question of how to adapt a *single-input* pretrained model to a *multi-input* downstream generation task.

In this work, we study two general schemes to adapt a pretrained language model to an NLG task. In the *single-input* setting, contexts are concatenated to create a sequence prefix from which the output is decoded as a continuation by the pretrained language model following Radford et al. (2018, 2019). The model can be used *as is* or with a small number of special token embeddings added to the vocabulary to identify the contexts. In the *multi-input* setting, the pretrained model is duplicated to form an encoder-decoder structure where the encoder processes contexts while the decoder generates the output.

2 Related work

Unsupervised pretraining for transfer learning has a long history in natural language processing, and a common thread has been to reduce the amount of task-specific architecture added on top of pretrained modules. Most early methods (Mikolov et al., 2013; Pennington et al., 2014) focused on learning word representations using shallow models, with complex recurrent or convolutional networks later added on top for specific tasks. With

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6053–6058
Florence, Italy, July 28 - August 2, 2019.

Persona for Speaker 1 (P1)
I like to ski
My wife does not like me anymore
I have went to Mexico 4 times this year
I hate Mexican food
I like to eat cheetos
P1: Hi
P2: Hello! How are you today?
P1: I am good thank you, how are you.
P2: Great, thanks! My children and I were just about to watch Game of Thrones.
P1: Nice! How old are your children?
P2: I have four that range in age from 10 to 21. You?
P1: I do not have children at the moment.
P2: That just means you get to keep all the popcorn for yourself.
P1: And Cheetos at the moment!
P2: Good choice. Do you watch Game of Thrones?
P1: No, I do not have much time for TV.
P2: I usually spend my time painting: but, I love the show.

Table 1: Sample dialogue from *PersonaChat* with persona facts for Speaker 1 (P1). Speaker 2 (P2) also has a random persona (not shown).

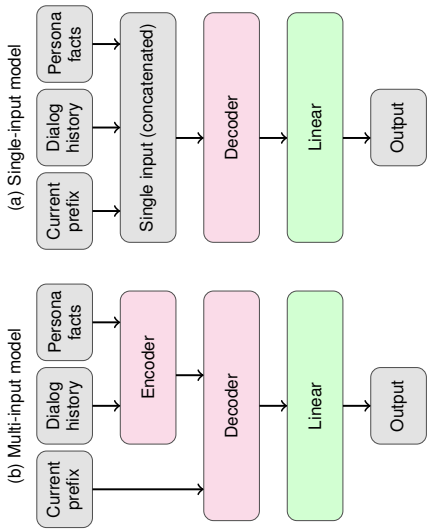

Figure 1: General model architectures: (a) single-input model; (b) multi-input model.

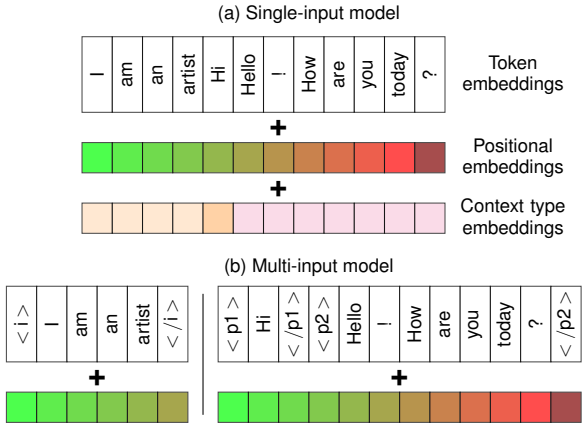

Figure 2: Token embeddings: (a) single-input model with CTE; (b) multi-input model with start/end tokens.

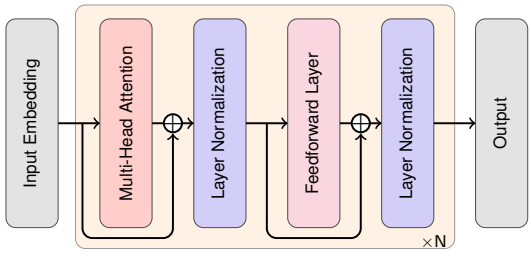

Figure 3: OpenAI GPT

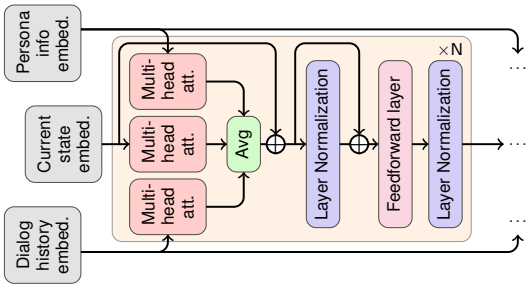

Figure 4: Multi-input Transformer-based architecture.

increased computing capacities, it has now become feasible to pretrain deep neural language models. Dai and Le (2015); Ramachandran et al. (2016) proposed unsupervised pretraining of a language model for transfer learning and to initialize encoder and decoder in a *seq2seq* model for machine translation tasks. Works in zero-shot machine translation used large corpora of monolingual data to improve performances for low-resource languages (Johnson et al., 2017; Wada and Iwata, 2018; Lample and Conneau, 2019). Most of the work transfering large-scale language models from and for monolingual NLG tasks focus on classification and natural language understanding (Kiros et al., 2015; Jozefowicz et al., 2016). Recently, Radford et al. (2019) studied large-scale language models for various generation tasks in the zero-shot setting focusing on summarization and translation and Wolf et al. (2019) presented early work on chit-chat.

3 Problem setting and dataset

NLG tasks can be divided into high entropy (story generation, chit-chat dialog) and low entropy (summarization, machine translation) tasks. We focus on the high entropy task of chit-chat dialog to study the use and effect of various types of contexts: facts, history and previous tokens.

Table 1 shows a typical dialog from *PersonaChat* (Zhang et al., 2018b), one of the largest multi-turn open-domain dialog dataset available. *PersonaChat* consists of crowdsourced conversations between real human beings who were asked to chit-chat. Each participant was given a set of 4-5 profile sentences that define his/her *persona*

for the conversation and asked to chitchat naturally and try to get to know each other. The dataset contains 162,064 utterances over 10,907 dialogs with 1,155 possible personas and 7 speaker turns per dialogue on average. Although it is one of the largest multi-turn dialogue datasets, *PersonaChat* is still too small to train a large-scale model; state of the art models trained directly on *PersonaChat* are very prone to overfitting (Dinan et al., 2019), hence the motivation for the present work.

4 Single- and multi-input adaptation

While we expect many more large-scale pretrained language models to become publicly available soon (Radford et al., 2019), our work is based on the only large-scale pretrained language model that was available at the time of this study, the OpenAI GPT (Radford et al., 2018). We refer to this publication for the details of the model, which is a 12-layer decoder-only Transformer (Vaswani et al., 2017) with masked multi-head attention. The model uses a bytepair encoding (BPE) vocabulary (Sennrich et al., 2015) with 40,000 merges and learned positional embeddings for sequences with at most 512 positions.

We now detail the various adaptation schemes we used to adapt this model to the task of open-domain dialogue. More specifically, in our target task the inputs to the model are: (i) a set of personality sentences, (ii) a dialog history involving two speakers, and (iii) the history of previously generated tokens for auto-regressive generation.

In the first adaptation setting, which we call the *single-input* model, the pretrained language model is used as is to generate an output sequence $\mathbf{y} = (y_1, \ldots, y_m)$ without any architectural modifications. Contexts are concatenated to create a sequence prefix from which the output is then decoded as a continuation. In this direction, several ways to construct prefixes from heterogeneous contexts can be investigated: (i) concatenating contexts with natural separators to make the test data distribution close to the training data (Radford et al., 2019) (in our case we added double quotes to each utterance to mimic dialog punctuation); (ii) concatenating contexts with additional spatial-separator tokens (fine-tuned on the target task) to build an input sequence (Radford et al., 2018); (iii) concatenating contexts and supplementing the input sequence with a parallel sequence of context-type embeddings (CTE) to be added to the token and positional embeddings (Devlin et al., 2018). Each CTE shows the context type for its input token as shown on Fig. 2a: $\mathbf{w}^{\text{info}}_{\text{CTE}}$ for persona info, $\mathbf{w}^{p1}_{\text{CTE}}$ for dialog history coming from person 1, and $\mathbf{w}^{p2}_{\text{CTE}}$ for person 2. These vectors are also fine-tuned on the target task.

In the second adaptation scheme, the *multi-input* model, the pretrained language model is duplicated in an encoder-decoder architecture (Fig. 1b). Similar to the *single-input* model, natural separators, spatial-separator tokens or context-type embeddings can be added for each persona fact and dialog utterance, surrounding the corresponding text with these tokens as preprocessing, as shown on Fig. 2b. Persona information and dialogue history are successively processed in the encoder (Fig. 4) to obtain two respective sequences of vector representations to be used as input to the decoder model. The multi-head attention layers of the decoder are modified to process the three inputs as follows (see Fig. 4). We copy the multi-headed attention layer of the decoder three times—for the embeddings of the current state, persona facts, and dialog history—averaging the results (Zhang et al., 2018a). The weights in both encoder and decoder are initialized from the pretrained model.

Using both encoder and decoder allows to separate the contexts (dialogue history and persona information) and alleviate the maximal length constraint of 512 tokens. Weight sharing between encoder and decoder reduces the total number of model parameters and allows for multi-task learning. On the other hand, untying the decoder and encoder lets the attention heads and architectures specialize for each task.

5 Results

We have performed a series of quantitative evaluation on the test subset of the *PersonaChat* dataset as well as a few quantitative evaluations.

Following the recommendations of the End-to-End conversation Modeling Task at DSTC-7 Workshop (Michel Galley and Gao), we evaluated the models on the following set of metrics: METEOR (Lavie and Agarwal, 2007), NIST-4, BLEU (Papineni et al., 2002) as well as diversity metrics: Entropy-4, Distinct-2, and the average length of the generated utterances. Table 2 illustrates the results for three typical models: the *single-input* model in the zero-shot set-

Model	METEOR	NIST-4	BLEU	Entropy-4	Distinct-2	Average Length
Single-input (zero-shot)	0.07727	1.264	2.5362	9.454	0.1759	9.671
Single-input (additional embeddings)	0.07641	1.222	2.5615	9.234	0.1614	9.43
Multi-input	**0.07878**	**1.278**	**2.7745**	9.211	0.1546	9.298

Table 2: Selected evaluation results and statistics.

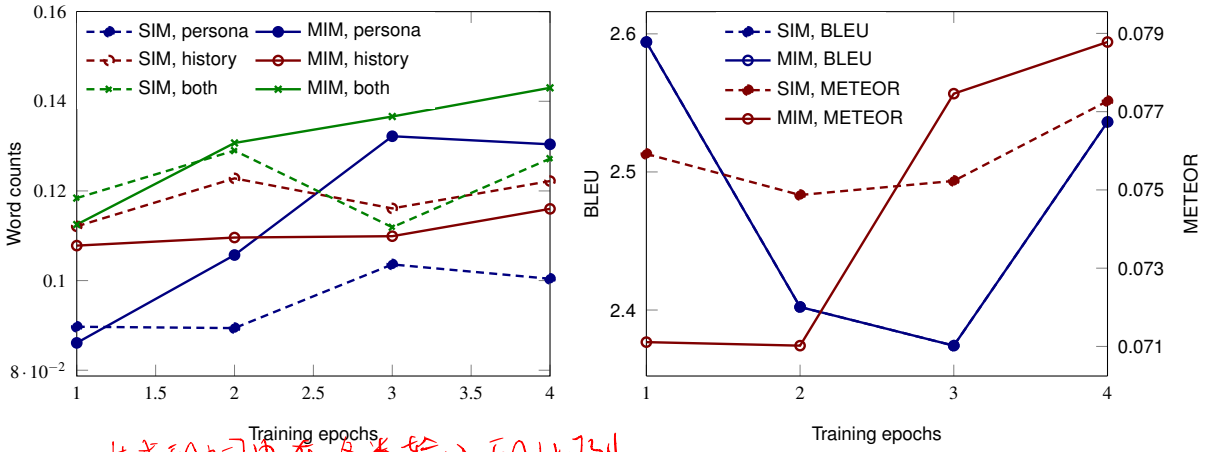

Figure 5: Results for single- (SIM) and multi-input (MIM) models; left: word statistics; right: evaluation metrics.

ting (no modification) and with additional embeddings fine-tuned on the target task, and the *multi-input* model in which the encoder and decoder are not shared, which is thus a high-capacity model in comparison to the previous two models. We can see that both approaches reach comparable performances on the automatic metrics with the *multi-input* model performing better on METEOR, NIST-4 and BLEU.

We investigated in greater detail the evolution of the *single-input* and *multi-input* models during training to understand the origin of their differences. To this aim, we tagged the words generated by each model according to four categories: (i) content words that were mentioned in the persona facts, (ii) content words that were mentioned in the dialog history, (iii) content words that were mentioned in both, and (iv) all other generated words. Fig. 5 shows the statistics of these types of words along a representative training run obtained using `compare-mt` (Neubig et al., 2019).

An interesting observation is that *single-input* and *multi-input* models adopt differing behaviors which can be related to an intrinsic difference between two contextual inputs: dialog history and personality facts. While dialog history is very related to sequentiality, personality facts are not sequential in essence: they are not ordered, a well-trained model should be invariant to the ordering of the facts. Moreover, a personality fact can be relevant anywhere in a dialog. On the contrary, dialog history is sequential; it cannot be reordered freely without changing the meaning of the dialog and the relevance of a particular utterance of the dialog history is strongly dependent on its location in the dialog: older history becomes less relevant.

This difference in nature can be related to differences in the models. *Single-input* adaptation is closer to a bare language-model and the comparison with *multi-input* model shows that the former tends to stay closer to the dialog history and consistently uses more words from the history than *multi-input* model. On the other hand, splitting encoder and decoder makes persona facts available to the *multi-input* model in a non-sequential manner and we can see that the *multi-input* model use more and more persona facts as the training evolves, out-performing the *single-input* model when it comes to reusing words from persona facts. We also note that the *multi-input* model, with its unshared encoder and decoder, may be able to specialize his sub-modules.

6 Conclusion

In this work, we have presented various ways in which large-scale pretrained language models can be adapted to natural language generation tasks, comparing *single-input* and *multi-input* solutions. This comparison sheds some light on the characteristic features of different types of contextual inputs, and our results indicate that the various archi-

tectures we presented have different inductive bias with regards to the type of input context. Further work on these inductive biases could help understand how a pretrained transfer learning model can be adapted in the most optimal fashion to a given target task.

References

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised Sequence Learning. *arXiv:1511.01432 [cs]*. ArXiv: 1511.01432.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2019. The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Vigas, Martin Wattenberg, and Greg Corrado. 2017. Googles multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the Limits of Language Modeling. *arXiv:1602.02410 [cs]*. ArXiv: 1602.02410.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. *arXiv:1506.06726 [cs]*. ArXiv: 1506.06726.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. *arXiv:1901.07291 [cs]*. ArXiv: 1901.07291.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiang Gao Bill Dolan Michel Galley, Chris Brockett and Jianfeng Gao. End-to-end conversation modeling: Dstc7 task 2 description. *In DSTC7 workshop (forthcoming)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. pages 3111–3119.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL) Demo Track*, Minneapolis, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. page 24.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv:1606.05250 [cs]*. ArXiv: 1606.05250.

Prajit Ramachandran, Peter J. Liu, and Quoc V. Le. 2016. Unsupervised Pretraining for Sequence to Sequence Learning. *arXiv:1611.02683 [cs]*. ArXiv: 1611.02683.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Takashi Wada and Tomoharu Iwata. 2018. Unsupervised cross-lingual word embedding by multilingual neural language models. *arXiv preprint arXiv:1809.02306*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. page 14.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. *arXiv:1901.08149 [cs]*. ArXiv: 1901.08149.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018a. Accelerating neural transformer via an average attention network. *CoRR*, abs/1805.00631.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018b. Personalizing dialogue agents: I have a dog, do you have pets too? *CoRR*, abs/1801.07243.