Contrastive Learning with Intent-Preserving and Intent-Corrupting Augmentations for Training Dialogue Embeddings. *

Ilya Alekseev
Lomonosov Moscow State University
ilya_alekseev_2016@list.ru

Denis Kuznetsov
Moscow Institute of Physics and Technology
kuznetsov.dp@phystech.edu

Abstract

Text embeddings from pre-trained language models have been proven to be extraordinarily useful for various sentence-level tasks, such as pair classification, similarity estimation, and retrieval. Corresponding models are usually trained on large amounts of clean and diverse data using contrastive loss. Unfortunately, there are no such datasets for the domain of dialogue data. In this work, we describe the process of mining a synthetic dataset of dialogues for contrastive learning with hard negative samples. We apply various augmentation strategies for constructing dialogues with preserved or corrupted sets of intents. We train dialogue embeddings and report its performance on transfer learning tasks: domain classification, intent similarity, dialogue retrieval.

***K*eywords** text embedding · dialogue · augmentation · natural language processing · contrastive learning

1 Introduction

Obtaining embeddings is one of the key tasks in machine learning and popular one in recent years, because vector representation of an object is a convenient mathematical object. If an embedding accurately and comprehensively encodes the semantics of the original data, it opens up the possibility of using it in a wide range of downstream tasks. добавить бы ссылку.

In the field of natural language processing, classical methods for text vectorization such as bag of words [1] and tf-idf [2] have long been discovered. Thanks to deep learning, we have witnessed remarkable word vectorization techniques such as word2vec [3] and GloVe [4], which convey the semantic similarity between words; CoVe [5] and ELMo [6], which encode information about the surrounding context. Recently, powerful encoder models have emerged that produce general-purpose text embeddings [7, 8, 9]. They incorporate so much semantic information about texts that it can be applied to tasks such as classification, clustering, ranking, semantic textual similarity, and more [10]. The success of these models is largely attributed to the use of contrastive learning on massive datasets.

However, the more specific the data structure, the more challenging it is to mine a large dataset. Language models, such as those presented in [11, 12], have demonstrated successful adaptation to the hierarchical and temporal characteristics of dialogues. However, it is important to note that their training primarily involves inter-token and inter-utterance tasks rather than inter-dialogue tasks.

Data is almost always scarce when it comes to building dialogue models. To date, numerous methods for generating synthetic dialogues have been devised [13, 14, 15, 16, 17], but they do not generate dialogues in pairs, as it is conceptually important for training SoTA text embedding. The simplest way to generate pairs is through augmentation [18]. In this work, we will describe a method for constructing a dialogue dataset for contrastive learning using various augmentations that preserve or alter the set of intents in the dialogue. These augmentations can be used for contrastive learning for pretraining powerful dialogue embeddings.

---

* *Citation*: **Authors. Title. Pages.... DOI:000000/11111.**

 ?

2 Problem Formulation

Задача из сырых данных создать выборку.

Dialogue Data. A dialogue is defined as the following list:

$$d = [(u_1, s_1), \dots, (u_n, s_n)],$$

where u_i represents the utterance of a participant in the dialogue at step s_i. We are interested in so-called task-oriented dialogues with two participants: the system and the user. With some approximation, they can be described as dialogues between a customer and a service worker (or a robot). During the dialogue, the customer has various intents that the worker strives to fulfill. These intents can be finding a restaurant and booking a table, calling a taxi, or purchasing a train ticket. We will consider two dialogues similar if they have a similar set of intents.

Кажется это к методу?

Augmentation. By augmentation, we mean the generation of new valid examples by transforming existing ones. Valid examples are those that sufficiently resemble real-world data. Let D be the set of valid objects. Then augmentation is a non-identical mapping $\text{aug}(d)$ that does not take objects outside the set of valid objects: $\text{aug} : D \to D$.

Typically, this generation is implemented by making small changes to a valid training object. For example, image augmentation might involve slight rotations or blurring. Text augmentation is especially challenging because validity implies adherence to language rules, meaningfulness and a certain style. In the case of dialogues, it is necessary to maintain the structure and role differentiation.

Embedding. Embedding is a mapping of D to a vector space:

$$e : D \to E \subseteq \mathbb{R}^d.$$

For an object d, its embedding $e(d)$ should convey some semantics of d. This is reflected in the fact that $e(d)$ may contain lexical information or latent features useful for classification and other downstream tasks. It is especially valuable if using embeddings $e(a), e(b)$ allows for assessing the semantic similarity of objects a, b.

3 Related Works

→ Ближе к introduction (section 2⟷3).

`this section is incomplete, this is still a draft`

3.1 Text Embedding

One of the breakthroughs in text embedding came from the need for solving the task of semantic textual similarity [19], i.e. comparing texts with each other. Previously existing methods were either weak (averaging word embeddings) or required enormous amount of computation (BERT cross-encoder). Comparing texts with bi-encoder is performed via averaging hidden states from the last transformer block of language model or taking hidden state of special token `[CLS]`. This setting opens up a possibility for clustering, retrieval, pair classification etc. within text domain [10].

3.2 Contrastive Learning

To make the encoder produce rich vector representations, it is necessary to train it with tasks where semantic features are engaged. A popular one is contrastive learning:

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(x, y))}{\sum_z \exp(\text{sim}(x, z))}.$$

Here, $x = e_\theta(a), y = e_\theta(b)$ are embeddings of a pair of semantically similar objects and $z = e_\theta(c)$ is semantically distant from x, sim is a metric similarity function (e.g. cosine). This task trains model parameters θ to make embeddings convey semantic similarity as cosine similarity.

This objective can be viewed as a special case of self-supervised learning, that long been discovered in the field of computer vision [20, 21, 22, 23]. This unsupervised approaches introduced a new view of transfer learning, allowing to advance from learning models to learning representations, which are more convenient to use for downstream tasks.

One of the most well-known methods for negative sampling was introduced by word2vec [3] and it is to make random sampling from the dataset. Present-day methods with random negative sampling (e.g. OpenAI embeddings [24]) implement this idea as in-batch negative sampling and rely on large batch. Another idea is to use hard negative samples [25, 26], which are the closest to positive pair sample among all negative samples.

The most efficient way of mining positive pairs is to use supervision. For example, datasets on NLI (natural language inference), QA (question answering), fact verification, paraphrases contain pairs of semantically similar texts and are appropriate for training text embedding [9]. This data is extremely clean, but scarce. Another way is to scrap data from web pages, such as QA forums (Quora), social media (Reddit), etc. This data is large, but can be noisy. When there are no any facility like supervised or scrapped pairs, augmentation is the last variant. Two augmented views of the same object must preserve semantics, that are essential for this object, but provide sufficiently hard task for a model of discriminating one positive sample from the rest of negative ones.

Present-day SoTA text embeddings such as BGE [7], GTE [9], E5 [8] follow one pipeline of training. First, a retrieval-oriented language model is trained [27, 28], that is a BERT-like model that can efficiently aggregate global information about input sequence. Second, a general-purpose fine-tuning is performed on a large weakly supervised text pairs with large batch contrastive learning. Third, task-specific fine-tuning is performed on supervised text pairs with contrastive learning with hard negative samples.

3.3 Similar Approaches

AugSBERT [29] cannot be viewed as the similar approach to ours. Its augmentation uses an already presented dataset of text pairs, which is absent in our case.

Text embedding. CERT [30] utilizes back-translation for mining positive pairs. ConSERT [31] utilizes various augmentations at the token-level. Doc2vecC [32] utilizes thesaurus-based augmentations and back-translation.

Dialogue embedding. Dial2vec [33] modifies transformer architecture by adding cross-attention between utterances of different participants. DialogueCSE [34] adds cross-attention between utterances of different stages of a dialogue.

Our approach is distinct from these above:

- our augmentations do not take objects outside the set of valid objects (do not violate grammars and meaningfulness), because our augmentations are thoroughly designed with context-aware language models in contrast to ConSERT that simply shuffles and deletes tokens leaving only lexical info in touch
- our augmentations form a full-fledged set of transformations that construct non-obvious examples, not just paraphrasing or token-level replacements
- our approach doesn't modify BERT architecture

4 Method

In this section, we describe our method. It can be divided into two stages. At the first stage, we develop and make augmentations for dialogue data. In result, the dataset consists of samples with following schema: original dialogue, set of augmentations with preserved intents, set with corrupted intents. At the second stage, we apply the contrastive learning.

4.1 Augmentations

Token Insertion. One of the simple yet effective ways to augment text is to lengthen it by inserting extra tokens. For this purpose, we added a special token `<mask>` to random places in the dialogues and used transformer models trained on the MLM task [35] to fill these masks. Insertion is rejected if the token proposed by the model is only a fragment of a word [36, 37] or if the prediction probability is below a manually set threshold. To take the dialogue context into account during token insertion, multiple consecutive responses were fed into the mask-filling model at once as a compromise between feeding single utterances and entire dialogue.

Token Replacement. This method is identical to the previous one, except that instead of adding the `<mask>` token, some tokens in the original dialogue are replaced. In this case, the mask-filling model is fed with single utterances to make replacements more diverse and random.

Back Translation. Translation from the original language to another language and then back to the original language. Neural machine translation models were used for this purpose [38].

Shuffling Utterances. Previous augmentations modify the dialogue within a single utterance, since they are methods applicable to arbitrary text data. It seems essential to learn how to change the order of utterances in a dialogue to create new valid dialogue. For this purpose, we propose using a model that measures the similarity between utterances within a dialogue. Using these similarities, it is possible to cluster utterances within each dialogue. Experiments showed that these clusters represent significant individual stages of the dialogue that can be shuffled with each other.

Shortening Dialogue. Individual clusters of utterances within the dialogue can be discarded, resulting in a pruned dialogue with fewer utterances.

Lengthening Dialogue. The special model was trained to arrange a list of given utterances. This transformer takes the text embeddings of each utterance as input sequence, without specifying information about their order in the original dialogue. It outputs ranks that can be used to "sort" the utterances to restore the original order. If some external responses are added to the original dialogue, this model generates a new, longer dialogue.

The augmentations mentioned above can be configured to either preserve or alter intent. Specifically, token replacement can be viewed as intent-corrupting augmentation, because all the keywords such as "restaurant", "taxi" etc. tend to be replaced. Pruning dialogue may remove some intents, but the result is still much similar to original dialogue, since its intents are fully encompassed by the original dialogue's intents. Shuffling utterances doesn't change any intents, it only changes their order. The rest of augmentations preserves intents because they either perform paraphrasing (back translation), or add new information (token insertion, lengthening dialogue).

To expand the set of augmentations even further, we use a composition of several ones. For more details on augmentation implementations, please refer to Appendix B.

4.2 Dialogue Encoder Architecture

As a model for embedding, we try BERT [35], RoBERTa [39], RetroMAE [27] without any modifications. The input is `[CLS] ut1 [SEP] ut2 [SEP] ut3`, and the output is the hidden state of `[CLS]` token from the last layer.

Also, we experiment with a slightly advanced dialogue language model, HSSA [11]. It uses BERT as a backbone and modifies its attention mechanism to capture the hierarchical structure of dialogue and reach the computational trade-off between feeding a transformer separate utterances and feeding an entire dialogue.

4.3 Pre-train Task

In our case, a, b are two intent-preserving augmentations of the same dialogue, and the remaining dialogues from the training batch are used as negative examples. This task simulates the second stage of training general-purpose text embeddings, which uses a large batch and weakly supervised text pairs. In addition to this, in order to reduce required batch size, we utilize intent-corrupting augmentations as hard negative samples. Notably, the technique with two augmentations and dual encoder resembles BYOL [40], well-known method for self-supervised learning `(maybe move this sentence to Related Works)`.

4.4 Evaluation

We evaluate dialogue embeddings in transfer learning setting. Specifically, our evaluation methods utilize embeddings as features, without modifying the encoder model itself. Inspired by [33] we employ these evaluation methods: 1) domain classification, 2) dialogue retrieval, 3) intent similarity. Evaluation is performed on MultiWOZ 2.2 dataset [41].

Domain classification. The goal is to predict in which domain a dialogue is taking place. In dataset there are 7 domains: attraction, bus, hospital, hotel, restaurant, taxi, train. Each dialogue can take place in multiple domains at once. The method is to train a linear classifier upon dialogue embedding, this is the so-called linear probe evaluation, that is used in many works. F1-macro is used to measure quality.

Dialogue retrieval. For each dialogue from validation split, the goal is to retrieve dialogues from train split with at least one domain in common. Ranking score is calculated as cosine similarity between query and answer embeddings. Mean average precision at 100 is used to measure quality.

Intent similarity. We sample pairs of dialogues from train and validation splits of dataset. A linear classifier is trained on the former pairs and evaluated on the latter pairs using Pearson correlation between predicted similarity scores and gold ones. The gold scores are obtained using DGAC clustering [42]. Clusters represent intents of dialogue participants. Hence, each dialogue can be associated with a set of intents. We define gold intent similarity of two dialogues as a dice similarity between their sets of intents.

5 Experiments

Dataset. In all experiments, we used the same dataset of dialogues. This dataset is a combination of several task-oriented dialogue datasets from DialogStudio [43]. All the dialogues were filtered based on their length, resulting in a dataset comprising 501K dialogues. For more details on the dataset used, please refer to Appendix A.

Baseline Solution. As our baseline solution, we employ BERT and BERT-RetroMAE without the pretraining method proposed by us. The results are presented in Table 1.

	Service Clf	Categorization	Relatedness	Retrieval
BERT	0.49			
BERT-RetroMAE				
BERT + Contr loss	0.68			

Table 1: Dialogue embedding evaluation results.

Так как это создание выборки. Добавь про статистики?
Какието инварианты?

Acknowledgments

This was was supported in part by......

References

[1] Wisam A. Qader, Musa M. Ameen, and Bilal I. Ahmed. An overview of bag of words;importance, implementation, applications, and challenges. In *2019 International Engineering Conference (IEC)*, pages 200–204, 2019.

[2] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60:493–502, 2021.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[4] Jeffrey Pennington, Richard Socher, and Christopher Manning. 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, October 2014. Association for Computational Linguistics.

[5] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors, 2018.

[6] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[7] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. C-pack: Packaged resources to advance general chinese embedding, 2023.

[8] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022.

[9] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning, 2023.

[10] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark, 2023.

[11] Zhenyu Zhang, Lei Shen, Yuming Zhao, Meng Chen, and Xiaodong He. Dialog-post: Multi-level self-supervised objectives and hierarchical model for dialogue post-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10134–10148, Toronto, Canada, July 2023. Association for Computational Linguistics.

[12] Yiyang Li, Hai Zhao, and Zhuosheng Zhang. Back to the future: Bidirectional information decoupling network for multi-turn dialogue modeling, 2022.

[13] Sungdong Kim, Minsuk Chang, and Sang-Woo Lee. Neuralwoz: Learning to collect task-oriented dialogue via model-based simulation, 2021.

[14] Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. Simulated chats for building dialog systems: Learning to generate conversations from instructions, 2021.

[15] Dazhen Wan, Zheng Zhang, Qi Zhu, Lizi Liao, and Minlie Huang. A unified dialogue user simulator for few-shot data augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3788–3799, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[16] Chujie Zheng, Sahand Sabour, Jiaxin Wen, Zheng Zhang, and Minlie Huang. Augesc: Dialogue augmentation with large language models for emotional support conversation, 2023.

[17] Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models, 2021.

[18] Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. Data augmentation for conversational ai. *arXiv preprint arXiv:2309.04739*, 2023.

[19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[20] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2017.

[21] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning, 2017.

[22] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination, 2018.

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[24] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training, 2022.

[25] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval, 2020.

[26] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding, 2022.

[27] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder, 2022.

[28] Luyu Gao and Jamie Callan. Condenser: a pre-training architecture for dense retrieval. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[29] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks, 2021.

[30] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding, 2020.

[31] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online, August 2021. Association for Computational Linguistics.

[32] Dongsheng Luo, Wei Cheng, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Bo Zong, Yanchi Liu, Zhengzhang Chen, Dongjin Song, Haifeng Chen, and Xiang Zhang. Unsupervised document embedding via contrastive augmentation, 2021.

[33] Che Liu, Rui Wang, Junfeng Jiang, Yongbin Li, and Fei Huang. Dial2vec: Self-guided contrastive learning of unsupervised dialogue embeddings, 2022.

[34] Che Liu, Rui Wang, Jinghua Liu, Jian Sun, Fei Huang, and Luo Si. Dialoguecse: Dialogue-based contrastive learning of sentence embeddings, 2021.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[36] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[37] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[38] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

[39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[40] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[41] Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. Multiwoz 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines, 2020.

[42] Mark Nagovitsin and Denis Kuznetsov. Dgac: Dialogue graph auto construction based on data with a regular structure. In Boris Kryzhanovsky, Witali Dunin-Barkowski, Vladimir Redko, and Yury Tiumentsev, editors, *Advances in Neural Computation, Machine Learning, and Cognitive Research VI*, pages 508–529, Cham, 2023. Springer International Publishing.

[43] Jianguo Zhang, Kun Qian, Zhiwei Liu, Shelby Heinecke, Rui Meng, Ye Liu, Zhou Yu, , Huan Wang, Silvio Savarese, and Caiming Xiong. Dialogstudio: Towards richest and most diverse unified dataset collection for conversational ai, 2023.

[44] Zhihan Zhou, Dejiao Zhang, Wei Xiao, Nicholas Dingwall, Xiaofei Ma, Andrew O. Arnold, and Bing Xiang. Learning dialogue representations from consecutive utterances. In *NAACL 2022*, 2022.

[45] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[46] Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. Convert: Efficient and accurate conversational representations from transformers, 2020.

A Dialogue Dataset

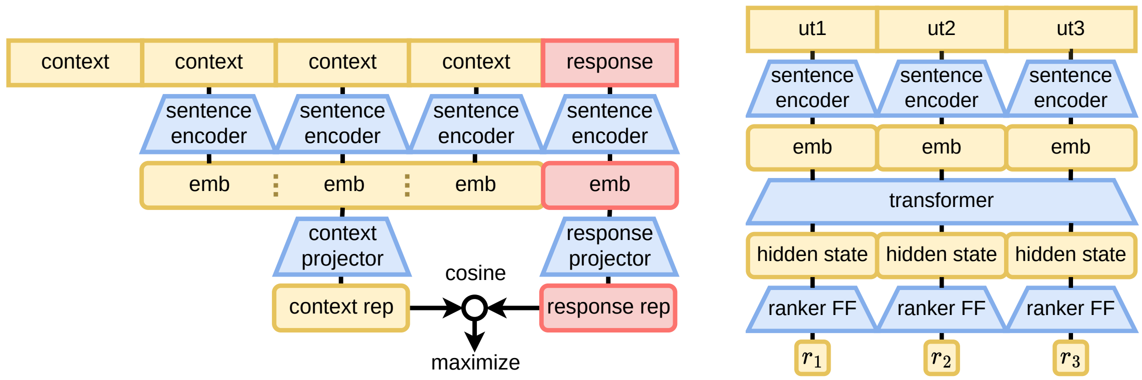
Large dataset is important for contrastive learning. For training our models, we took a merge of some task-oriented datasets from DialogStudio collection [43]. All of them are listed in the table 2.

Name	# dialogues	# utterances	# tokens
AirDialogue	321K	4086K	49.4M
SimJointGEN	100K	1584K	22.5M
MS-DC			
MetaLWOZ			
MULTIWOZ2_2			
SGD			
KETOD			
FRAMES			
Disambiguation			
ABCD			
AirDialogue			
BiTOD			
Taskmaster1			
Total			
Filtered	501K	6320K	83.7M

Table 2: All the datasets are taken from DialogStudio collection [43]. BERT tokenizer is used to count # tokens column.

B Auxiliary Models

As mentioned in Section 4.1, we trained special models to perform dialogue-level augmentations.



(a) for measuring similarity between utterances

(b) for lengthening dialogue

Figure 1: Auxiliary models for performing augmentations.

B.1 Pairwise Model

We utilized a model for measuring the similarity between dialog utterances. In its basic form, it can be implemented as follows: take sentence embeddings of the utterances and compare the cosine similarities between them. Training such a model on sequential utterances using contrastive learning yields commendable utterance embeddings [44].

However, this approach has a significant drawback; it does not consider the context from several preceding utterances. In a dialogue, it is crucial to compare not just pairs of utterances but pairs of context-response. Therefore, we employed the following model (Fig. 1a):

1. First, embeddings for all context utterances $c = [u_1, \dots, u_k]$ and the response r are obtained using a pretrained sentence encoder.
2. The embeddings of the context are concatenated and passed to a projector that outputs a vector representing the context.
3. The response embedding is fed into a second encoder, resulting in a vector representing the response.
4. The cosine similarity between the obtained vectors is computed as a measure of the context and response similarity.

`aws-ai/dse-bert-large` model from hugging face [45] was used as the sentence encoder. The model was trained with a contrastive loss using in-batch negative sampling, with the following parameters: batch size is 128, temperature is 0.05, context size is 3, projection size is 256. Only 3 last layers of sentence encoder were fine-tuned in order to decrease computational cost. Trained model reaches 0.955 retrieval accuracy@5.

Batches were formed from "context-response" pairs from the entire dialog dataset, where negative examples were not samples from the same dialog but entirely random examples from the dataset. This allows batching of arbitrary sizes, not limited to the dialog size, making the pre-training task more challenging.

The resulting model closely resembles the ConveRT model [46] for obtaining utterance embeddings. The drawbacks of the latter model are, firstly, that it is proprietary, and secondly, its architecture is highly specific and does not utilize the familiar BERT-like backbone.

As a result, the obtained model is capable of recognizing individual stages in dialogues (Fig. 2). This behavior is achieved due to two factors. Firstly, when a dialogue furthers to a new topic, the similarity between two consecutive utterances drops substantially. This is clearly visible, for example, in dialogues in which ordering a taxi replaces booking a table in a restaurant. Secondly, within one topic, there are also small drops in similarity in cases where there is a transition from one question to another. For example, the question "how many people should I book for?" replaces the question "which restaurant do you prefer?". Moreover, since these questions relate to one topic, they remain close to themselves and distant to questions on other topics. Therefore, clusters are obtained.

(!provide the dialogue and change pic to vector instead of raster!)

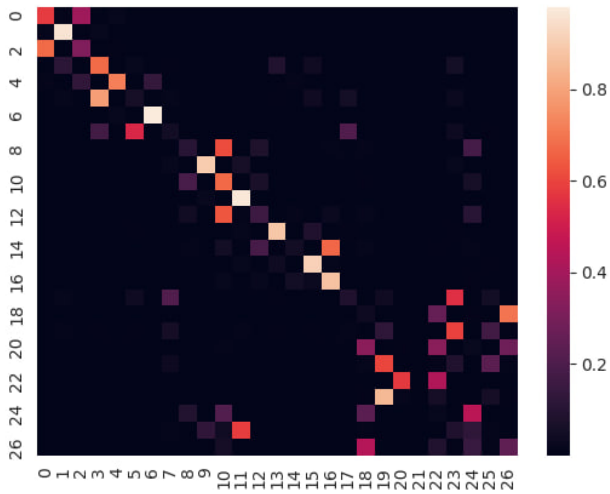


Figure 2: All similarities between contexts and responses within a dialogue. It is easy to see, that consecutive utterances form clusters.

B.2 Listwise Model

We trained the special model to merge utterances of two different dialogues (Fig. 1b). It is a transformer over text embeddings of the utterances. Sentence encoder is `sentence-transformers/all-mpnet-base-v2` from hugging face. We used 4-layer transformer with 4 attention heads and hidden dimension twice smaller than sentence encoder's one, i.e. 384. Only 3 last layers of sentence encoder were fine-tuned in order to decrease computational costs.

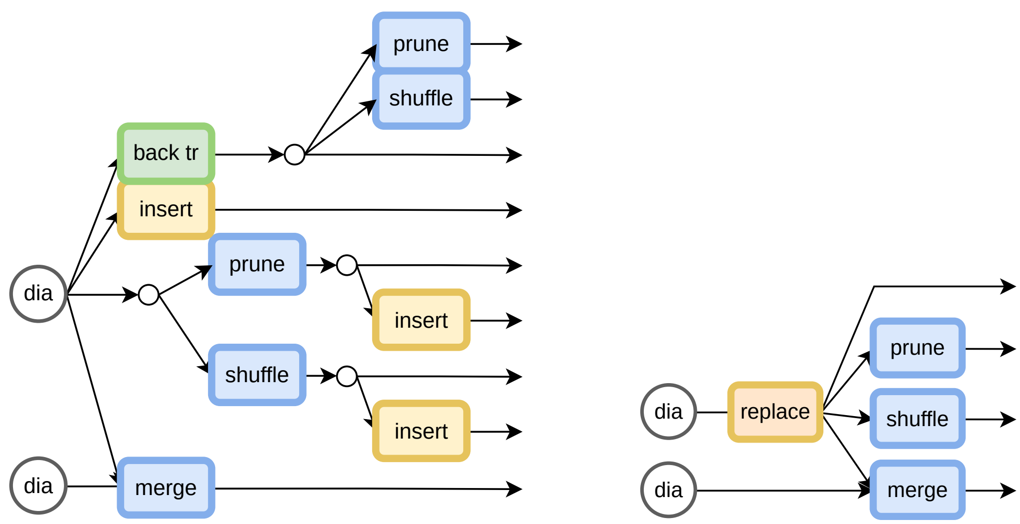
Output ranks are transformed with softmax function. Then KL-divergence between output and target probabilities are minimized. Target probabilities are defined as softmax over true ranks of utterances, i.e. $-i$ for i-th utterance in dialogue.

Resulting model trains to "sort" given utterances. Thanks to the attention mechanism of transformers, this can be viewed as asking the children at physical education class to look at each other and line up by height.

To measure the sorting quality, we need to utilize appropriate metric. All traditional ranking metrics such as nDCG are designed to compare with gold ranks, not just sorting quality. So during validation of our model, we were converting the ranks to a permutation over the original sequence of n elements. Then, we calculated the number of transpositions. It is easy to implement and can normalized by maximum possible number of transpositions $n(n-1)/2$, resulting in $[0, 1]$-ranged metric. Our trained model reaches 0.96 value.

C Composition of Augmentations

In order to maximize diversity of training data, we use not only 5 basic augmentations described in section 4.1, but also 4 extra compositions of augmentations. All the resulting pipelines are defined in Fig. 3.



(a) intent-preserving

(b) intent-corrupting

Figure 3: Compositions of augmentations.