Personalizing Retrieval-Based Dialogue Agents

Pavel Posokhov, Anastasia Matveeva, Olesia Makhnytkina(✉), Anton Matveev, and Yuri Matveev

ITMO University, Saint Petersburg, Russia
makhnytkina@itmo.ru

Abstract. The development of various kinds of interactive assistants at present is highly in demand. In this field, one critical problem is the personalization of these dialog assistants seeking to increase user loyalty and involvement in a conversation, which may be a competitive advantage for enterprises employing them. This paper presents a study of retrieve models for a personalized dialogue agent. To train models the Persona Chat and Toloka Persona Chat Rus datasets are used. The study found the most effective models among the retrieval models, learning strategies. Also, to solve one of the major limitations of the personalization of dialogue assistants—the lack of large data sets with dialogues containing person characteristics—a text data augmentation method was developed that preserves individual speech patterns and vocabulary.

Keywords: Personalized dialogue systems · Retrieve models · Text augmentation

1 Introduction

Currently, due to the promising prospects for the practical use of dialogue systems in various fields, their development has become one of the most relevant tasks in NLP. Conversational systems, also known as chatbots, are in a high demand in industry and everyday life. According to a study by Business Insider, the chatbot market will grow from $2.6 billion in 2021 to $9.4 billion by 2024 at a compound annual growth rate (CAGR) of 29.7%.

Currently, depending on the tasks being solved, it is common to separate goal-oriented models, the purpose of interaction with which is strictly defined and aimed at solving a specific problem: ordering movie tickets, booking a hotel room, etc., usually the dialogue in this case is constricted to one subject area, which greatly simplifies the interaction between the chatbot and a person and open-domain models which can operate in various subject areas, the goals of communication in this case can be different, including phatic (idle-domestic). The latter are of the greatest interest due to their versatility, since fine-tuning open-domain chatbots is more optimal than developing a new model to solve a specific problem. Also, it is clear that the construction of these types of models

S. R. M. Prasanna et al. (Eds.): SPECOM 2022, LNAI 13721, pp. 554–566, 2022.
https://doi.org/10.1007/978-3-031-20980-2_47

is an important step necessary to create a strong AI. In addition, the task of open-domain communication, that is, without a strict goal-setting of a dialogue, on free topics, is also relevant, for example, less than five percent of Twitter posts are specific questions, while about 80 percent contain statements about a personal emotional state, thoughts or actions, represented by the so-called 'Me'-forms.

Like all systems, open-domain systems have several disadvantages. For example, despite the rapid development of natural language processing, and the presence of numerous dialogue studies in particular, caused by the success of the application of modern deep learning techniques to computational linguistics problems, modern dialogue systems are at the initial stage of their development. Human interactions with such models indicate the existence of numerous problems, such as lack of a coherent personality, lack of explicit long-term memory, a tendency to give vague and meaningless answers, these factors are the main reason for the decrease in the motivation of the second participant (human) to continue the communicative act. Often these problems are caused by the lack of direct information about the person and learning from the aggregate sample of dialogues of various people, which leads to the model adhering to a general, average personality, which can often lead to factual errors, inconsistency or superficiality of the narrative. It is possible to avoid such behavior of the model by creating personalized dialogue agents trained on datasets of people's dialogues, extended by personality characteristics.

Retrieval architectures, generative models, and hybrid models combining the first two types in varied sequences are commonly used for the development of dialogue agents. This article proposes an approach to developing a personalized dialogue agent using retrieval models.

Currently, the field of natural language processing is actively developing in the direction of improving the quality of solving problems via the emergence of more complex and deep architectures of neural networks that require large datasets for training. The use of pre-trained models and additional fine-tuning on target datasets also relies on the amount of data available. To increase the volume, data often are collected from multiple sources. This approach has significant drawbacks for the development of virtual assistants, smart speakers, interactive robots, etc. since the data contains the speech of multiple people, each with its personal characteristics, which leads to a lack of cohesion in the responses, views, judgments, and style of communication. For handling this issue, augmentation of text data preserving individual speech patterns and vocabulary that is unique to the original text is relevant. This article presents a study on the influence of the use of augmentation methods that preserve style and vocabulary distinctive for a person on the performance of models for the automatic generation of replicas of a personalized dialogue agent.

For producing high-quality models, it is critical to have large datasets. It is possible to increase the volume using successful dialogues between a bot and a user [10]. However, at the stage of training neural networks, this approach is not applicable and the use of data augmentation methods is suggested.

2 Related Work

For building open-domain dialog systems, it is common to distinguish two types of architectures [18,20]: retrieval search models which are based on the principle of ranking: choosing the most relevant answers to the input context from the selection of possible answers, and refine models which are generator models that produce a system response token by token, based on the input context and, optionally, additional data necessary for generation, and also hybrid ensembles of those models, based on various strategies for their interaction. Non-goal-oriented dialogue systems have several features, including:

1. Models tend to produce generic, low-content answers. This problem is more characteristic of generative models, however, with a sufficient variety of candidates in the data for retrieval models, this problem also occurs. The main reason for that is the lack of extensive extralinguistic knowledge of the model, which is why it produces answers containing as little factual information as possible, thus reducing the likelihood of making a mistake.
2. The conversational agents are not consistent in their responses which is reflected in contradictory statements following one another (for example, to the question "What do you like?" the model may answer "winter", but the next answer to "What do you not like?" might be "cold and snow"). The main reason for that is the lack of an explicit logical apparatus and reliance on a consistent personality of the agent, which, without additional personification, is represented by an average set of all personalities in the training sample.
3. Dialogue models are not fully capable of grasping the context, primarily due to the lack of extralinguistic knowledge and the inability to personalize communication.

One of the main approaches to solving these problems is the personalization and personification of dialogue agents. Personalization is changing or modifying the responses of the model according to the information about another participant of the dialogue, passed to the model as an additional metadata vector. Some researchers find this method unethical in certain cases, however, personalization is more applicable to goal-oriented dialogue systems and is not included in this study. Personification involves full-fledged modeling of responses by the model in the context of the information about the persona of the sender. Person metadata can include various facts about the person (e.g., gender, age, hobbies, etc.) enabling a direct or indirect communication on behalf of the described person.

Recent studies of retrieval models [5] show that the use of pre-trained BERT type transformer models as an embedding component of NLP models significantly increases their efficiency in solving a wide range of problems, including solving ranking problems as encoders, which was also confirmed in the first phase of our study. Bidirectional Encoder Representations from Transformers (BERT) is the coding part of the transformer architecture, it uses the self-attention mechanism and multi-head attention to represent words, positional coding of tokens, which allows to achieve the effect of a contextual representation of words. The effectiveness of this approach is largely attributed to pre-training BERT on a

large dataset with auto-labeling (MLM—masked word prediction, next sentence prediction, etc.), with the possibility of further fine-tuning on the target dataset to improve the representation of the lexical meaning of words in the context of the problem under consideration.

1. Bi-Encoder [12] architecture is represented by a pair of independent BERT base models, which are initialized with the same parameters before the training starts. Models receive context and candidate vectors as inputs, encoded using the WordPiece tokenizer, and process them independently. Dotprod is used to calculate an error. Negative sampling, where the distance value for distractors can be partially masked, can also be used during training.
2. Cross-Encoder [12]—the architecture for ranking tasks which employs one instance of BERT, the input of which is a concatenated vector of context and candidates separated by a special token. The resulting vector is then compacted by weighted summation through a linear layer to obtain a scalar value that can be interpreted as the similarity between the candidate vector and the context. This approach allows the internal attention of the model to encode both vectors, which significantly increases the efficiency of their representation, though significantly increases the operating time and memory resources consumed.
3. Poly-Encoder [12]—the architecture that utilizes a pair of BERT embedders to represent contexts and candidates, similar to the Bi-Encoder model, but for the calculation of the similarity of the candidate and context vectors, the latter passes through an attention block that has a collection of representations of the context initialized randomly and optimized during training, where the candidate vector is the query. Then the distance between the context and the candidates is calculated by multiplying their vectors. This approach allows to obtain contextual representations that are dependent both on the context and on the candidates, similar to how it happens in the Cross-Encoder architecture and improves the performance of the model.
4. Co-Encoder [30] similar to the Poly-Encoder architecture has two independent context and candidate representation blocks, but instead of the standard attention mechanism, it processes several stages of co-attention. This approach also produces extended views of the context. Additionally, the diffusion of information in this case extends to the representation of candidates, and the use of co-attention allows for incorporation of additional metadata vectors which are also used to expand the views.

Generative architectures of conversational agents generate responses token by token using language modeling. Among the modern approaches to training generative models of personalized dialogue agents are:

1. GPT [7] is a Transformer-based architecture and training procedure for natural language processing tasks. Training follows a two-stage procedure. First, a language modeling objective is used on the unlabeled data to learn the initial parameters of a neural network model. These parameters are adapted to a target task using the corresponding supervised objective.

2. Blenderbot [21] is a model created by the Facebook AI development team. It is built according to the standard architecture of the Seq2Seq transformer model. It is created for user interaction but can also be used for many other text generation tasks.
3. Seq2Seq [16,24,28] is a tandem of two recurrent neural networks: encoder and decoder. These models can consist of several encoder and decoder blocks and a variable number of parameters. Also, such models employ an attention mechanism that solves the issue that the influence of previous block states on the current one decreases exponentially with the distance between words. The layer of this mechanism is often implemented by a single-layer neural network that receives the hidden state of the encoder block and the context, which is represented by the previous hidden state of the decoder block, as input.

A tremendous advantage of retrieval models over generative ones is the high volume of relevant content in the answers, allowing the dialogue to appear more meaningful and realistic. Moreover, retrieval models have an objective advantage over generative ones since the former are evaluated by simple and effective metrics such as top-k, which reflects the probability of finding the correct answer in the first k ranks, R-precision k (equivalent to the value of recall for the k-th position), mrr (the reciprocal of the rank of the target response), etc. However, since automated evaluation metrics may not adequately reflect the quality of the dialogue system, it is critical to evaluate the performance of the model by a person; as the study of dialogue systems shows, retrieval models are superior to generative ones in this respect as well. Taking all of that into account, in this study, we focus on retrieval models.

Also for producing high-quality models it is critical to have large datasets. It is possible to increase the volume using successful dialogues between a bot and a user [10], however, at the stage of training neural networks this approach is not applicable and the use of data augmentation methods is suggested.

A set of the basic augmentation techniques is presented in the EDA algorithm [25], which consists of four operations: synonym replacement, random insertion, random permutation, and random deletion. When augmenting textual data, it is important to preserve the meaning of the text; various dictionaries, for example, WordNet [8,25] or pre-trained language models such as BERT [26], GPT2 [15], Word2Vec [19], Glove [8], etc. are commonly used to replace words with synonyms.

Work [19] considers several augmentation techniques similar to those in [8]: interpolation method, extrapolation method, adding random noise. The difference was the use of a pre-trained Word2Vec language model, unlike Glove in the previous work. The experiments were conducted for the problem of extracting types of contract elements from a text [3].

Various techniques for adding noise to word vector representations obtained using the Word2Vec model were reviewed in [29]: Gaussian noise, Bernoulli noise, Adversarial Noise, etc. Studies of adding noise to vector representations of words [8,19,29], in general, showed favorable results, however, the use of such

augmentation methods involves embeddings of non-existing words, which can potentially lead to a mismatch in message class labels.

Work [14] proposes a contextual augmentation approach based on the assumption that sentences are natural even if words in sentences are replaced by other words with paradigmatic relations.

An alternative to generation of paraphrases is reverse translation. Reference [6] has studied the quality of reverse translation using deep neural networks, showing positive results.

A study in [2] reviews the GECA (Good-Enough Compositional Data Augmentation) method, which is based on the idea that if two entities appear in a common environment, then any additional environment where one of the entities appears independently is also valid for another entity.

Another non-trivial augmentation technique is presented in [9] where the augmentation is performed via text shuffling using a neural network.

Unfortunately, none of the methods above preserve the style and the vocabulary unique to the original message which is one of the most important issues for various applications such as the development of dialog assistants [17] since the personalization of dialog assistants is key to user loyalty.

The method using the generative model LAMBADA [1] showed particular efficiency in text data augmentation. One of the main traits of this method is a phase for filtering augmented data using the BERT classifier. Also, this method is noteworthy for its supposed ability to preserve the speech characteristic of a person. A significant drawback of this method is the high demand for computing resources. Additionally, data from intermediate stages can not be immediately discarded due to the specifics of the algorithm, which leads to an increased usage of permanent (physical) memory.

When modifying text data, transformations can lead to distortions in the text making it grammatically or semantically incorrect or stylistically distinct from the original text and it demands for techniques that can augment text preserving the style, vocabulary while maintaining syntactic integrity.

There are also more advanced augmentation methods that have the ability to preserve the styles and vocabulary of a text, for example, Paraphrases generator based on syntax trees transformation [4]. This method involves modifying a text by transforming the syntax tree based on syntactic grammars. Text augmentation via syntax trees with the generation of new data based on syntactic templates was also considered in [23,27].

3 Methods

3.1 Models

In the study, we consider retrieval and refine models approaches for creation of personalized dialogue agents.

Retrieval Models. Numerous studies show that it is possible to improve the performance of nlp models in solving a wide range of tasks, including ranking tasks, by using pre-trained BERT-type transformer models as an embedding component. Bidirectional Encoder Representations from Transformers (BERT) is the encoding part of the transformer architecture. A self-attention mechanism, multi-head attention, and positional coding of tokens are employed for representing words in BERT, which allows obtaining of contextual representation of words. Pre-training BERT on a large dataset with auto-labeling (MLM—masked word prediction, next sentence prediction, etc.) is one of the main reasons for the effectiveness of the method. As part of the study, we performed fine-tuning of BERT models on target datasets to improve the representation of the lexical meaning of words within the scope of the problem under consideration.

Within the scope of this study, we consider the following architectures based on BERT base models—Simple Bi-Encoder, Bi-Encoder with Coattention [11]. Text preprocessing includes only tokenization based on the Wordpiece [22] method. The following metrics are used to evaluate the performance of the models:

(1) $R@k$—an interpretation of the recall for the ranking problem. The number of relevant responses from the k highest ranks divided by the total number of relevant responses. In the traditional form, the value k must match the number of relevant responses, thus $R@k = acc(topk)/k$, however, the number of highest ranks considered can be changed independently. Computing $R@k$ requires knowledge of all documents relevant to the query (in the case of a dialog system, $k = 1$), then $R@1 = acc(topk)$. The sensitivity of the model can be analyzed by varying the number of ranks.
(2) MRR or inverse rank, calculated by the formula $MRR = 1/r$, where r is the rank of the correct answer. This is a statistical measure for evaluating models that return responses sorted by probability of correctness. Unlike $R@1$, MRR can only be applied in the case of a single correct answer, and the metric itself is multiplicatively inverse.

3.2 Augmentation

In this work, we present a new augmentation method that preserves the distinctive characteristics of a person's speech (see Fig. 1). The idea for this augmentation method derives from the adapted data augmentation scheme shown in [4]. The proposed method also includes the stage of extracting syntax trees. The stage of transformation and generation of paraphrased data in this method is executed in a single process.

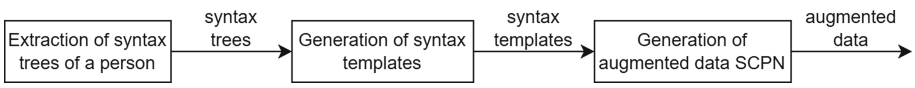

Fig. 1. Schematic diagram of the augmentation process.

Syntax tree extraction—in this stage, syntax trees are extracted for each replica using Stanford Core Nlp for the English language and SyntaxNet for the Russian language producing syntax trees of all replicas used by each person.

```
(ROOT
  (SQ
    (VP (VB have)
      (NP
        (NP (DT any) (NNS pets))
        (PP (IN at)
          (NP (NN home)))))
    (?)))
```

The example of a syntax tree is produced from the original message "Do you have any pets at home?". Creation of syntactic templates—the syntactic trees obtained at the previous stage are transformed to a uniform format by removing all the words of the original sentence from them and leaving only syntactic structural units. Further, duplicates and similar ones are removed from the set of obtained syntactic structures, and n most frequently used ones are selected from the remaining ones. An example of a template is represented by formula (1).

$$(ROOT(SQ(VP(VB)(NP(NP(DT)(NNS))(PP(IN)(NP(NN)))))(?))) \quad (1)$$

where $ROOT$ is the root of the sentence; SQ—inverted yes/no question, or main clause of a wh-question, following the wh-phrase in $SBARQ$; VP—verb phrase; VB—verb, base form; NP—noun phrase; DT—determiner; NNS—noun, common, plural; PP—prepositional phrase; IN—preposition or subordinating conjunction; To these sets of templates obtained for each person we add common for OpenAttacker EOS line endings. Then, together with the original replica of the person, they are sent to the Syntactically Controlled Paraphrase Network (SCPN—encoder-decoder model for syntactically controlled generation of paraphrases) from the OpenAttaker framework [13] to generate augmented data. This way, each replica can be transformed in n different ways. Since only syntactic structures characteristic of a person are used for data augmentation, the syntactic features of a person's speech are preserved. Vocabulary and style are preserved since they remain almost unchanged as augmentation is based on transformation of the syntax tree. If any of the parts of speech is not present in the original replica, SCPN adds the necessary parts of speech (conjunctions, prepositions, particles) to maintain the syntactic coherence of the augmented replicas. If the remaining parts of speech are missing, they are added using a word generating LSTM.

4 Experiments

4.1 Datasets

1. PERSONA-CHAT is an English-language corpus of dialogues between two participants, reproducing artificial personas modeled based on 3–5 sentences

with a description (e.g. “I like to sk, “I am an artist”, “I eat sardines for breakfast daily”). This dataset consists of 8939 completed conversations and 955 persons as a training set, 1000 dialogues and 100 persons for validation, and 968 dialogues and 100 persons for testing. To prevent word overlapping, information about persons after the collection of dialogues was reworked, using paraphrasing, generalization, and concretization.

2. Toloka Persona Chat Rus is a dataset compiled at the Laboratory of Neural Systems and Deep Learning at the Moscow Institute of Physics and Technology by each participant in the study modeling a certain specified person in dialogues. This dataset is packaged in two files: profile.tsv containing lines with characteristics of 1505 different persons, represented by 5 sentences such as “I draw”, “I live abroad”, or “I have a snake”; dialogues.tsv containing 10,013 dialogues in Russian between study participants. Russian.

4.2 Retrieval Models Results

In this study, we chose the most optimal configurations of ranking: Simple Bi-Encoder and CoBERT. In addition, for the modules, we employed a modified training method which involves the preliminary training of Siamese architectures, when the encoders of context and candidates are trained synchronously and then are separated and trained separately. One key condition for such learning method is to prevent a complete optimization of the weights at the pre-learning stage because, having reached the clear minima of the error function, the Siamese encoders, when separated, will find themselves in local minima, the escape from which will require unreasonably large values of the learning step, which negates the benefit of pre-learning. A comparison of the selected methods applied to the English Persona Chat dataset is presented in Table 1.

Table 1. Performance of retrieval models Persona Chat.

Strategy	Model	Persona	Valid loss	Valid acc	Valid r1	Valid r5	Valid r10	Valid MRR
15/0	Cobert	Mean	0.898	0.535	0.535	0.714	0.971	0.653
	Cobert	Concat	0.892	0.512	0.512	0.680	0.968	0.634
	Simple	Mean	0.911	0.509	0.509	0.686	0.963	0.629
	Simple	Concat	0.919	0.522	0.521	0.694	0.952	0.638
0/15	Cobert	Mean	1.220	0.259	0.260	0.484	0.929	0.425
	Cobert	Concat	0.932	0.257	0.257	0.467	0.870	0.412
	Simple	Mean	1.298	0.253	0.254	0.474	0.919	0.418
	Simple	Concat	0.921	0.352	0.352	0.490	0.949	0.489
5/10	cobert	mean	0.874	0.539	0.539	0.716	0.972	0.656
	Cobert	Concat	0.890	0.533	0.533	0.712	0.970	0.653
	Simple	Mean	0.900	0.519	0.519	0.696	0.964	0.638
	Simple	Concat	0.995	0.519	0.519	0.699	0.962	0.642

According to the results of the comparison, the best result was achieved with the training with five epochs trained together and ten separately. With this approach to training, the most effective was the CoBERT model.

The performance results when using the Russian-language Toloka Persona Chat Rus dataset are presented in Table 2.

Table 2. Performance of retrieval models Toloka Persona Chat Rus.

Strategy	Model	Persona	Valid loss	Valid acc	Valid r1	Valid r5	Valid r10	Valid r10
15/0	Cobert	Mean	0.9	0.53	0.53	0.71	0.97	0.65
	Cobert	Concat	0.89	0.51	0.51	0.68	0.97	0.63
	Simple	Mean	0.91	0.51	0.51	0.69	0.96	0.63
	Simple	Concat	0.92	0.52	0.52	0.69	0.95	0.64
0/15	Cobert	Mean	1.22	0.26	0.26	0.48	0.93	0.43
	Cobert	Concat	0.93	0.26	0.26	0.47	0.87	0.41
	Simple	Mean	1.3	0.25	0.25	0.47	0.92	0.42
	Simple	Concat	0.92	0.35	0.35	0.49	0.95	0.49
5/10	Cobert	Mean	0.87	0.55	0.54	0.72	0.97	0.66
	Cobert	Concat	0.89	0.53	0.53	0.71	0.97	0.65
	Simple	Mean	0.9	0.52	0.52	0.7	0.96	0.44
	Simple	Concat	0.99	0.52	0.52	0.7	0.96	0.44

With Russian-language data, the best performance was achieved by the CoBERT and the 5/10 training approach.

4.3 Augmentation Results

Examples of data augmented with syntactic paraphrasing are presented in Table 3.

Table 3. Examples of data augmented with syntactic paraphrasing.

Source text	**Dataset**	**Result**
No I am not found a new girl at a wedding last week	PERSONA-CHAT	i didn't find a girl at the wedding.
Люблю животных, просто обо- жаю, как и свою работу). Я фан-Rustастику люблю	Toloka Persona Chat Rus	Люблю животных, просто люблю свою работу и фан- тастику.

Training of retrieval models (simple Bi-encoder, CoBERT) was performed with three different setups for data augmentation. The option $aug - prob = 0.0$ corresponds to training without augmentations. With $aug - prob = 1.0$, each statement is replaced by augmentation. The option $augprob = 0.5$ where half of the statements are replaced by augmentation. Results of the experiments are presented in Table 4.

Table 4. Performance of retrievel models with augmentation.

Augment Prob	Language	Model	Valid acc	Valid r1	Valid r5	Valid r10	Valid MRR
0.0	En	Cobert	0.455	0.456	0.652	0.945	0.585
	En	Simple	0.405	0.406	0.6860	0.943	0.537
1.0	En	Cobert	0.234	0.234	0.428	0.788	0.340
	En	Simple	0.290	0.290	0.522	0.874	0.407
0.5	En	Cobert	0.463	0.463	0.655	0.942	0.593
	En	Simple	0.425	0.425	0.702	0.950	0.554
0.0	Ru	Cobert	0.516	0.517	0.694	0.965	0.636
	Ru	Simple	0.520	0.520	0.699	0.959	0.440
1.0	Ru	Cobert	0.234	0.234	0.429	0.765	0.338
	Ru	Simple	0.283	0.284	0.507	0.857	0.401
0.5	Ru	Cobert	0.645	0.645	0.795	0.967	0.739
	Ru	Simple	0.629	0.629	0.785	0.967	0.728

5 Discussion and Conclusion

Within the scope of this study, we analysed the modern types and architectures of dialogue systems, among which we identified the most efficient type, namely, retrieval models. Among them, the best performance metrics were achieved by the CoBERT architecture when training five epochs together and ten others separately. Also in this paper, we propose a text data augmentation method that preserves individual speech patterns and vocabulary. We find that data augmentation with the presented method produces an increase in performance at values of $aug - prob < 1.0$ because in this case there is a chance that the original message remains unchanged. We observe a performance increase for different models up to 12%.

Acknowledgments. The research was financially supported the Russian Science Foundations (project 22-11-00128).

References

1. Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., Zwerdling, N.: Not enough data? deep learning to the rescue! (2019). http://arxiv.org/abs/1911.03118
2. Andreas, J.: Good-enough compositional data augmentation (2019). http://arxiv.org/abs/1904.09545
3. Chalkidis, I., Androutsopoulos, I., Michos, A.: Extracting contract elements. In: Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law, pp. 19–28. ICAIL '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3086512.3086515
4. Coulombe, C.: Text data augmentation made simple by leveraging NLP cloud apis (2018). http://arxiv.org/abs/1812.04718

5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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, vol.1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
6. Edunov, S., Ott, M., Auli, M., Grangier, D.: Understanding back-translation at scale. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 489–500. Association for Computational Linguistics, Brussels, Belgium (2018). https://doi.org/10.18653/v1/D18-1045, https://aclanthology.org/D18-1045
7. Floridi, L., Chiriatti, M.: GPT-3: its nature, scope, limits, and consequences. Mind. Mach. **30**(4), 681–694 (2020). https://doi.org/10.1007/s11023-020-09548-1
8. Giridhara, P.K.B., Mishra, C., Venkataramana, R.K.M., Bukhari, S.S., Dengel, A.R.: A study of various text augmentation techniques for relation classification in free text. In: ICPRAM (2019)
9. Guo, H., Mao, Y., Zhang, R.: Augmenting data with mixup for sentence classification: an empirical study (2019). arXiv:abs/1905.08941
10. Hancock, B., Bordes, A., Mazare, P.E., Weston, J.: Learning from dialogue after deployment: feed yourself, chatbot! pp. 3667–3684 (2019). https://doi.org/10.18653/v1/P19-1358
11. Humeau, S., Shuster, K., Lachaux, M., Weston, J.: Real-time inference in multi-sentence tasks with deep pretrained transformers (2019). http://arxiv.org/abs/1905.01969
12. Humeau, S., Shuster, K., Lachaux, M.A., Weston, J.: Poly-encoders: architectures and pre-training strategies for fast and accurate multi-sentence scoring. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=SkxgnnNFvH
13. Iyyer, M., Wieting, J., Gimpel, K., Zettlemoyer, L.: Adversarial example generation with syntactically controlled paraphrase networks. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long Papers), pp. 1875–1885. Association for Computational Linguistics, New Orleans, Louisiana (2018). https://doi.org/10.18653/v1/N18-1170, https://aclanthology.org/N18-1170
14. Kobayashi, S.: Contextual augmentation: data augmentation by words with paradigmatic relations (2018). arXiv:abs/1805.06201
15. Kumar, V., Choudhary, A., Cho, E.: Data augmentation using pre-trained transformer models (2020). arXiv:abs/2003.02245
16. Lin, Z., Liu, Z., Winata, G.I., Cahyawijaya, S., Madotto, A., Bang, Y., Ishii, E., Fung, P.: XPersona: evaluating multilingual personalized chatbot. In: Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI. pp. 102–112. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.nlp4convai-1.10, https://aclanthology.org/2021.nlp4convai-1.10
17. Matveev, A., Makhnytkina, O., Matveev, Y., Svischev, A., Korobova, P., Rybin, A., Akulov, A.: Virtual dialogue assistant for remote exams. Mathematics **9**(18) (2021). https://doi.org/10.3390/math9182229, https://www.mdpi.com/2227-7390/9/18/2229
18. Ni, J., Young, T., Pandelea, V., Xue, F., Adiga, V., Cambria, E.: Recent advances in deep learning-based dialogue systems (2021)

19. Papadaki, M., Chalkidis, I., Michos, A.: Data augmentation techniques for legal text analytics (2017)
20. Posokhov, P., Apanasovich, K., Matveeva, A., Makhnytkina, O., Matveev, A.: Personalizing dialogue agents for Russian: retrieve and refine, vol. 2022, pp. 245–252 (2022). https://doi.org/10.23919/FRUCT54823.2022.9770895
21. Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E.M., Boureau, Y.L., Weston, J.: Recipes for building an open-domain chatbot. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 300–325. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.eacl-main.24, https://aclanthology.org/2021.eacl-main.24
22. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. CoRR abs/1508.07909 (2015), http://arxiv.org/abs/1508.07909
23. Shen, T., Lei, T., Barzilay, R., Jaakkola, T.S.: Style transfer from non-parallel text by cross-alignment (2017). arXiv:abs/1705.09655
24. Sugiyama, H., Mizukami, M., Arimoto, T., Narimatsu, H., Chiba, Y., Nakajima, H., Meguro, T.: Empirical analysis of training strategies of transformer-based Japanese chit-chat systems (2021). arXiv:abs/2109.05217
25. Wei, J.W., Zou, K.: EDA: easy data augmentation techniques for boosting performance on text classification tasks (2019). arXiv:abs/1901.11196
26. Wu, X., Xia, Y., Zhu, J., Wu, L., Xie, S., Fan, Y., Qin, T.: Mixseq: a simple data augmentation method for neural machine translation, pp. 192–197 (2021). https://doi.org/10.18653/v1/2021.iwslt-1.23
27. Yang, Z., Hu, Z., Dyer, C., Xing, E.P., Berg-Kirkpatrick, T.: Unsupervised text style transfer using language models as discriminators (2018). arXiv:abs/1805.11749
28. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: I have a dog, do you have pets too? In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 2204–2213. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-1205, https://aclanthology.org/P18-1205
29. Zhang, Z., Zweigenbaum, P.: Gneg: graph-based negative sampling for word2vec (2018). https://doi.org/10.18653/v1/P18-2090
30. Zhong, P., Sun, Y., Liu, Y., Zhang, C., Wang, H., Nie, Z., Miao, C.: Endowing empathetic dialogue systems with personas (2020). arXiv:abs/2004.12316