Personalizing Dialogue Agents for Russian: Retrieve and Refine

Pavel Posokhov, Kirill Apanasovich, Anastasia Matveeva, Olesia Makhnytkina, Anton Matveev
ITMO University
Saint Petersburg 197101, Russian Federation
paposokhov, apan.kirill, aamatveeva, makhnytkina, aymatveev@itmo.ru

***Abstract*—Currently, the development of automatic dialogue systems is in demand not only for traditional applications (increasing the level of automation of contact centers) but also for relatively new cases (development of virtual assistants, smart speakers, interactive robots). Adding information about characteristics of a person, which the agent must take into account when generating a response, i.e. personification of conversational agents, helps to increase user loyalty and engagement. This paper presents a study of Retrieve and Refine models for automatic generation of utterances of a personalized Russian-speaking dialogue agent. To train models in Russian, the Toloka Persona Chat Rus dataset is used. Refine models that used an adaptation of the BlenderBot model for the Russian language showed worse performance than for datasets in English. For Retrieve models, a solution based on the BERT encoder model was proposed, which made it possible to obtain the value of the metrics hits@1=0.705 for the model without a person, and hits@1=0.717 for the model with a person.**

I. Introduction

The development of dialogue systems is the fundamental and one of the most urgent tasks in NLP in recent years, given the promising prospects for its application in practice. Conversational systems (or chatbots) are in great demand in industry and everyday life. The chatbot market is projected to grow from $2.6 billion in 2021 to $9.4 billion by 2024 at a compound annual growth rate (CAGR) of 29.7 percent. At the same time, modeling such systems remains labour-intensive, due to the complexity of the tasks they solve, which require detailed consideration.

At the moment, following the problems under consideration, it is common to feature several approaches: task-oriented models, the purpose of interaction with which is strictly defined and aimed at solving a specific problem, ordering tickets, booking a table in a restaurant, etc., usually the dialogue in this case is confined by a single topic, which greatly facilitates the interaction between the chatbot and a person; as well as open-domain models that can function across various topics and the goals of communication in this case can be different, including phatic (used for general purposes of social interaction). The latter are of the greatest interest due to their versatility since fine-tuning open-domain bots is more optimal than developing new models to solve a specific problem. Also, the construction of these types of models is an important step necessary to create a strong AI. Moreover, the problem of open-domain communication, that is, without a limited scope of a dialogue, on free topics, is also relevant, for example, less than 5 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.

However, at the same time, open-domain systems still have drawbacks. Despite the great progress both in the field of natural language processing and in dialogue research in particular, caused by the success of the application of modern deep learning methods to computational linguistics problems, modern dialogue systems are at the initial stage of their development. Human validation of such models indicates several serious problems, including: 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 communication. Most of them are caused by the lack of an image of a person and learning from the cumulative sample of dialogues from various people, in the result of which the model tends to stick to a general, average personality, which can often lead to factual errors, inconsistency or superficiality of the narrative. It is possible to eliminate the described shortcomings by creating personalized dialogue agents trained on datasets of specific people’s conversations, enhanced by personality traits. To solve the problem of developing open-domain dialogue systems, it is common to distinguish two types of architectures [1]: retrieval search models which are based on ranking, choosing the most relevant to the input context answers from the scope of possible answers; refine, generator models that produce a system response token by token, based on the input context and optionally additional data necessary for generation (note: information about the person). The article proposes an approach to developing a personalized dialogue agent in Russian.

There are few studies available on dialogue personification for the Russian language and only a single dataset is commonly available, so to test that the developed approaches can be universally applied, experiments were conducted with both English-language and Russian-language datasets. At present, the most common approach to personification of dialogue assistants is to add information about a person to the content of the dialogue and to combine the vectors of the person and the context. This approach is also used in this work.

II. Related works

Open-domain dialogue systems are often classified into three categories according to the approach to producing replicas of a model and the type of underlying machine learning problem. Generative (refine) systems generate responses sequentially, often in a seq2seq fashion, reflecting the user's message, conversation history, and additional metadata into a sequence of unique responses. Such architectures tend to create flexible, contextualized dialogue responses, although they sometimes lack coherence and factual information.

Search models (retrieval) are based on ranking search results for answers according to a certain distance measure that reflects the relevance of the statements in the database to the input data, which is also represented by the dialogue history and additional metadata. Search engines are limited by the availability of the possible responses, and sometimes the responses show a weak correlation with the context of the dialogue. However, despite this, at this moment they demonstrate the best performance due to greater consistency at the semantic, syntactic, and pragmatic levels.

Hybrid models combine the architectures described above. The interaction between the architectures is possible in two routes: the responses obtained via a generative model are ranked by a retrieval model for picking the best one, or a generative model is used to refine the responses obtained by ranking in the context of the dialogue in question.

Retrieval models have a clear advantage over generative ones since the former are evaluated by simple and efficient metrics, such as top-k, which reflects the probability of finding the correct answer in the first k ranks, R-precision k, mrr, hits, etc. Also, since automated evaluation metrics might not adequately reflect the performance of a dialogue system, it is critical to test the system with a human expert. As noted in several studies, for example [2], ranking models are superior to generative models in this regard. Let us review some modern generative models.

One of the first research works that can be considered a starting point for building personified conversational agents is Information Retrieval (IR) system [3]. This model employs an algorithm based on TF-IDF as a BoW tokenizer. It searches for the closest values by measuring the cosine distance between the context and candidate vectors. Information about a person enhances the model via concatenation with the BoW context.

A slightly different approach to personification is demonstrated in [4]. The Starspace model contains a single layer of embeddings for the context and candidates, the parameters of which are optimized during training. In training, margin ranking is utilized as an error function and k-negative sampling is used to not only minimize the distance to the target replicas but also to maximize it for other statements except the correct ones. The context vector of this model is the concatenation of the history and the person.

The Ranking Profile Memory Network model is similar to Starspace with the same tokenization algorithms and approach to training, but instead of simple concatenation, a "memory" model is used to interact with a person, which employs the attention mechanism. Thus the new context vector is the original vector weighted by a softmax function applied to the measure of similarity between the original context vectors and each feature of the person.

The Key-Value Profile Memory Network [5] model is also similar to Starspace, however, it extends the attention mechanism used in the Memory Network with key-value pairs where the keys represent the history of the dialogue and the values represent the meaning of the following statements. This method allows the model to remember past dialogues that directly affect the current prediction. For the datasets with which the models of personalized agents are trained, we consider personachat [2] and toloka rupersonachat [https://toloka.ai/ru/datasets]. Table I presents a comparative analysis of the reviewed architectures for developing personalized dialogue agents.

The best results were shown by the KV Profile Memory model. We will consider this and IR models as the baseline for further research.

The main modern approaches to training generative models for personalized dialogue agents are shown in Fig. 1.

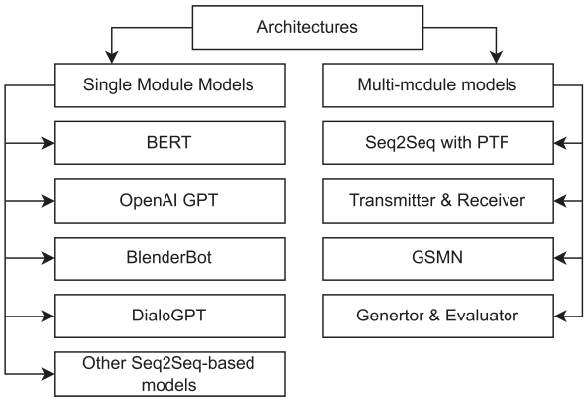

Fig. 1. Classification or neural network architectures

At the stage of text generation, word vectors are fed to the input of a neural network that generates a response. In some studies, the models consist of several modules, and processing information about a person is delegated to another module; the results are then transmitted to the input of the decoder of the first module where they are combined with the results of processing the history of the dialogue. Among the modern approaches to training generative models for personalized dialogue agents are:

1) Seq2Seq [2], [6], [7] 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

TABLE I. PERFORMANCE OF BASELINE MODELS

Method	No Persona		Original Persona		Revised Persona	
	ppl	h@1	ppl	h@1	ppl	h@1
Generative Models						
Seq2Seq	38.08	0.092	40.53	0.084	40.65	0.082
Profile Memory	38.08	0.092	**34.54**	0.125	**38.21**	0.108
Ranking Models						
IR baseline	-	0.214	-	0.410	-	0.207
Starspace	-	0.318	-	0.491	-	0.322
Profile Memory	-	0.318	-	0.509	-	**0.354**
KV Profile Memory	-	0.349	-	**0.511**	-	0.351

the hidden state of the encoder block and the context, which is represented by the previous hidden state of the decoder block, as input.

2) Seq2Seq with Personality Trait Fusion (PTF) module [8] - in this method, the processing of information about a person is performed by a separate encoder module. After that, the result is combined with the processed dialogue in the decoder of the seq2seq model.
3) BERT [9]–[11] is a pre-trained neural network developed by Google. OpenAI GPT [12] is a pre-trained model developed by the OpenAI company of the same name.
4) Two-module Transmitter and Receiver (TR) system [12] is a model where the Transmitter module, based on a pre-trained OpenAI GPT model, generates text, and the Receiver measures the closeness between created experiences and personas. It consists of two encoders initialized with BERT.
5) Blenderbot [13] 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.
6) DialoGPT [14] is a dialogue model from Microsoft, pre-trained on 147 million comments from the Reddit platform. The model is based on another GPT-2 model from OpenAI.
7) Generative Split Memory Network (GSMN) [14] is a model featuring a delegation of information processing about a person into a separate module represented by two encoder models. The text generating module is implemented with the DialoGPT model.
   Two modules Generator and Evaluator (GE) [15] is a Generator module built on top of the seq2seq model. The Evaluator, in turn, consists of the Naturalness and Consistency submodules. The aim of the former is to distinguish generated text from created by a human. The latter is an NLI classifier that makes sure the dialogue is consistent.

For experiments with the described models, we consider personachat [2] and ConvAI2 [16] datasets. For generative models, Perplexity and BLEU (bilingual evaluation understudy) are considered as quality metrics. Tables II and III and shows comparative characteristics of the generative models.

TABLE II. PERFORMANCE OF VARIOUS TEXT GENERATION MODELS USING THE CONVAI2 DATASET

Paper	Architecture	PPL	BLEU-1
Hiroaki Sugiyama, 2021 [6]	Seq2Seq	16.28	-
Qian Liu, 2020 [12]	TR	15.12	-
BlenderBot90M, 2020 [13]	BlenderBot	11.36	-
BlenderBot2.7B, 2020 [13]	BlenderBot	8.74	-
BlenderBot9.4B, 2020 [13]	BlenderBot	**8.36**	-

TABLE III. PERFORMANCE OF VARIOUS TEXT GENERATION MODELS USING THE PERSONACHAT DATASET

Paper	Architecture	PPL	BLEU-1
Siqi Bao, 2020 [10]	BERT	-	0.41
Saizheng Zhang, 2018 [2]	Seq2Seq	40.53	-
Qian Liu, 2020 [12]	TR	15.12	-/-
Zhaojiang Lin, 2020 [7]	Seq2Seq	41.64	**0.74**
Hyunwoo Kim, 2020 [11]	BERT	11.7	0.27
Haoyu Song, 2021 [15]	GE	29.99	-
Andrea Madotto, 2020 [9]	DialoGPT	**11.08**	0.16
Yuwei Wu, 2021 [14]	GSMN	33.51	0.7

For generative models, BlenderBot shows the best performance and we will consider this model the baseline. Research of personalized conversational agents is actively developing, creating new models for both the retrieval approach and the refine approach. A significant part of the research is conducted for datasets in English and significantly less for datasets in Russian; it appears, creation of new models and adaptation of existing ones for Russian-language personalized dialogue agents is a high-priority issue.

III. DATASETS & METHODS

For building a non-task-oriented dialogue system it is critical to have a clear definition of the machine learning problem being solved. Specifically, a training collection is a selection of dialogues in natural language $X : X_1, X_2, \ldots, X_n$ where each replica Xi corresponds to an answer Yi belonging to the set of answers $Y : Y_1, Y_2, \ldots, Y_n$, in addition, the input data X can be extended with metadata $M : M_1, M_2, \ldots, M_n$ containing information about a person and other additional information. Thus, the dialogue agent model is represented by the function

$DA(X_i, M_i) = Y_i$. At present, the most efficient method for training such models is supervised fine-tuning of a pre-trained unsupervised representation of the input data.

A. Datasets

Both personalized datasets containing dialogues and those not including data about a person have to be considered when developing a non-task-oriented dialogue system.

1) Ubuntu Dialogue Corpus is a corpus of Ubuntu technical support chat dialogues in English, which does not contain data characterizing persons. This dataset contains 930,000 dialogues, more than 7 million lines, and more than 269 million words packaged in CSV files; its dictionary includes 100 million unique words. Each dialogue has at least 3 phrases, 8 on average. In addition, this corpus is weakly structured due to the use of a microblogging system of communication between participants, which, combined with other factors, makes it suitable for training both question-answer and dialogue systems.
2) DSTC7 (Dialog System Technology Challenge 7) is a dataset of dialogues between two participants obtained from various sources, in particular from the Ubuntu chat and the consultation corpus collected by the University of Michigan. This dataset also does not contain data characterizing persons. This dataset includes 815 conversations, with an average of 18 messages per conversation and 9 words per message. The dataset is packaged as a JSON file.
3) PERSONA-CHAT is an English-language corpus of dialogues between two participants, reproducing artificial personas modeled based on 3 to 5 sentences with a description (e.g. “I like to ski”, “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.
4) 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.
5) ConvAI2 is a dataset from the challenge of the same name, which aimed at finding approaches to creating high-quality conversational agents. The dataset is based on the Personachat dataset. The dataset contains 131438 utterances, 17878 dialogues, and 1155 different personas in the training set. The model validation set has 7801 utterances, 1000 dialogues, and 100 personas. The subset for testing has 6634 statements, 1015 dialogues, and 100 persons. To increase variety, the persons from the training set were modified by paraphrasing, for example, the sentence ”I just got my nails done” could be rewritten as ”I love to pamper myself on a regular basis”.

A summary of all datasets is present in Table IV.

B. Methods

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

1) Retrieval models: Studies [17] show that utilizing pre-trained Bidirectional Encoder Representations from Transformers (BERT) type transformer models as an embedding component for NLP models greatly increases their performance in solving a wide range of problems, including solving ranking problems. BERT is the encoding part of the transformer architecture, it uses a self-attention mechanism and multi-head attention to represent words, with the positional encoding of tokens it allows to obtain contextual representations 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.

Our study will consider architectures that use BERT base models - Bi-Encoder, Cross-Encoder, and Poly-Encoder [18].

Bi-Encoder architecture is represented by a pair of independent BERT base models initialized with the same parameters before training. Models receive context and candidate vectors as inputs, encoded using the WordPiece tokenizer, and process them independently. Dotprod is used for measuring errors. Negative sampling can also be utilized during training by partially masking the distance values for distractors.

Cross-Encoder architecture for ranking 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 reduced by weighted summation through a linear layer to obtain a scalar value that can be interpreted as the similarity of the candidate vector and the context. This approach makes it possible to use the internal attention of the model to encode both vectors, which significantly increases the efficiency of their representation but noticeably increases the inference speed and memory resources consumed.

Poly-Encoder architecture employs a pair of BERT embedders to represent contexts and candidates similar to the Bi-Encoder model. However, to calculate the similarity of the candidate and context vectors, the latter passes through an attention block that includes m context representations 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 makes it possible to obtain representations dependent on both context and candidates, similar

TABLE IV. A SUMMARY OF ALL DATASETS

Dataset	Language	Number of dialogues	Number of persons
Ubuntu Dialogue Corpus	EN	930 000	-
DSTC7	EN	815	-
PERSONA-CHAT	EN	9 139	1 155
Toloka Persona Chat Rus	RU	10 013	1 505
ConvAI2	EN	19 893	1 355

to the Cross-Encoder architecture, and improve the model performance.

To encode a joint representation of a context and a person, we concatenate them since it is the most efficient way to obtain a sentence representation [17]. A cls token is capable of embedding two sentences separated by a sep token. Moreover, it allows applying attention across contexts and persons, which empowers them to be more extensively encoded in each other's context.

2) Refine models: Adaptation of the model is performed via a Russian-language tokenizer and pre-training of the multilingual BERT model with the conversational Russian dataset deeppavlov-conv-bert. Text preprocessing only includes tokenization based on the Byte-Pair Encoding (BPE) method [19]. As the basis for our model, we consider the BlenderBot model, which is a standard Seq2Seq Transformer architecture to generate responses rather than retrieve them from a defined set of responses. The model exists in three configurations: with 90M parameters (8 encoder layers, 8 decoder layers, embedding dimensionality – 512 tokens, attention heads - 16), 2.7B parameters (2 encoder layers, 24 decoder layers, embedding dimensionality – 2560 tokens, attention heads - 32), and 9.4B parameters (4 encoder layers, 32 decoder layers, embedding dimensionality – 4096 tokens, attention heads - 32). This model will be adapted for Russian-language datasets.

C. Metrics

1) Retrieval models: The following metrics are used to evaluate the effectiveness of the models:

$R@k$ - 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 dialogue system, $k = 1$), then $R@1 = acc(topk)$. The sensitivity of the model can be analyzed by varying the number of ranks.

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.

In terms of clarity R@k and mrr are the best metrics for ranking models but in dialogue systems implementations it must be considered that this metric transforms into nonlinear form. The reason for that fact is the strict differentiation of semanticaly and stylisticaly similar candidates, that is not typical for human speech. This means that the metric will increase slower the higher it gets. While human perception of model quality will increase linearly.

2) Refine models: The following metrics are used to evaluate the effectiveness of the models:

Perplexity (ppl) is a metric for assessing the quality of language models. This metric is calculated as the inverse probability of the test set, normalized by the number of words in the test set. The general formula is shown in 1. The lower the value of this metric, the higher the quality of the model.

$$PP(W) = \frac{1}{P(w_1, w_2, \ldots, w_N)^{\frac{1}{N}}} = \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots, w_N)}} \quad (1)$$

BLEU (bilingual evaluation understudy) is a metric for evaluating the quality of models, with values from 0 to 1. It is calculated as the proportion of word uni- or bigrams that matched the original.

IV. RESULTS

1) Retrieval models: Experiments for retrieval models were conducted both for datasets with a persona and without a persona. Experiments for training models without a persona revealed the best approach to taking into account the context of the dialogue. In the final revision of the article [18] during training, the parameters of the context model and candidates were trained independently of each other. However, experiments show that synchronous training of both models, although it can reduce accuracy (making it reasonable to run only one BERT embedding block), greatly increases the inference rate of the model, as well as the training speed, since the parameters are optimized at 2 batches per iteration, and will reduce the effect of overfitting on small training collections due to the reduction in the number of trained weights. The results of training the original BERT encoder models are presented in Table V.

Experiments on Ubuntu and DSTC 7 datasets conducted to determine the most optimal use of BERT as a retrieval model demonstrated that using a cross encoder is inefficient, so Siamese models were used instead.

Further research was aimed at determining the best similarity function for utterances. Dotprod, ossim, Euclidian, and Manhattan metrics were reviewed.

TABLE V. RESULTS OF TRAINING MODELS WITH SEPARATE AND SIAMESE ENCODERS

Model	R@1 1 model	R@1 2 models	MRR 1 model	MRR 2 models
Ubuntu				
Bi-Encoder	0.760	0.806	0.844	0.880
Poly-Encoder 16	0.766	0.812	0.851	0.883
Poly-Encoder 64	**0.767**	**0.813**	**0.854**	**0.884**
Poly-Encoder 360	0.754	0.809	0.842	0.881
Cross-Encoder	0.742	-	0.833	-
DSTC 7				
Bi-Encoder	0.437	0.518	0.538	0.551
Poly-Encoder 16	0.447	0.527	0.550	**0.559**
Poly-Encoder 64	0.438	0.528	0.546	0.556
Poly-Encoder 360	**0.453**	**0.538**	0.545	0.557
Cross-Encoder	0.502	-	**0.599**	-

The similarity function, dotprod, is calculated as a product of the context and candidate matrices $dotprod = X_i \times Y_i$. Maximization of this parameter leads to the model choosing the most semantically similar texts, since each axis of the output vector can be interpreted as a numerical representation of the value of one or more aggregate semes present in the utterance, and the product of parallel, and therefore similar vectors, produces the highest result. The cosine similarity function is calculated in a similar way

$$cossim = \frac{X_i \times Y_i}{||X_i|| \times ||Y_i||},$$

but is normalized by the lengths of the matched vectors, which negates the increase in the function value only due to a simple increase in the value of individual features of the vectors. There are also more various distance functions, such as Euclidean

$$Ed^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2$$

or Manhattan

$$Md = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n|.$$

If the latter are used as an error function, models need to use their inverse values.

Table V shows the results of the comparison, demonstrating that the dot product of vectors is indeed the most efficient, but given that the results of some functions deteriorate when using Siamese encoders, we can assume that it is dotprod that allows us to use one representation model as efficiently as two. The average results of training models on the Ubuntu dataset with various similarity functions obtained in the scope of this study are presented in Table VI.

Employing BERT representations, as expected, offers a significant increase in the accuracy of the model for a personalized dialogue system, even without additional information about a person. However, it is worth noting that concatenation of context and person is not the optimal method for aggregating input data. Comparison of the baseline personachat models and the BERT encoder models modified in the scope of this study, using the metrics calculated for the personachat and toloka rupersonachat datasets, is shown in Table VII. For building a personified conversational agent, the input context vector is enhanced by concatenating with the person vector. The described architectures were implemented with a pre-trained Russian-language BERT model (rubert).

The proposed approach for training retrieval models showed results that are significantly superior to Base-line models.

2) Refine models: .

To test the applicability of the BlenderBot model and compare its performance with other models, experiments were conducted on the PersonaChat dataset. Results of running the BlenderBot model on the Persona Chat dataset is shown in Table VIII

The results obtained are quite close to those obtained on other datasets. Therefore, this model was further tested with the Russian-language dataset Toloka Persona Chat Rus. Table IX shows the performance of the model with approximately 90 million parameters. This time, various tokenization methods were also used: SentencePiece (SP) [20], Byte-Pair Encoding (BPE) [19], WordPiece (WP) [21].

The adaptation of the BlenderBot model, which shows good results for the English language, showed significantly worse performance for the Russian language. As the main approach for creating personalized dialogue assistants, we will consider retrieval models that have surpassed the results obtained on both Russian and English datasets. The construction of hybrid models can also produce positive results, and such experiments are scheduled for further research.

V. CONCLUSION

At the moment, there are two main types of dialogue systems: generative and ranking; there are also different types of ensembles based on them. The latter show the best performance, both in automatic and expert testing and have simpler and more informative metrics. Due to these factors, in this paper, we review the retrieval approach. From the experiments, we draw the following conclusions:

1) BERT representations in conversational systems provide a significant increase in accuracy in comparison to the basic models of conversational agents, such as IR and Starspace. On average, the bi and poly encoders implemented in the scope of the study outperform the base models by 1.5 times for

TABLE VI. RESULTS OF TRAINING MODELS WITH VARIOUS SIMILARITY FUNCTIONS

2*Model	Dotprod		ossim		Euclidian		Manhattan	
	R@1	**MRR**	**R@1**	**MRR**	**R@1**	**MRR**	**R@1**	**MRR**
1 model								
Bi-Encoder	0.760	0.844	0.309	0.501	0.202	0.491	0.205	**0.483**
Poly-Encoder 16	0.766	0.851	0.309	0.522	0.210	0.482	**0.213**	0.482
Poly-Encoder 64	**0.767**	**0.854**	**0.312**	**0.525**	0.209	0.425	0.210	0.426
Poly-Encoder 360	0.754	0.842	0.311	0.502	0.212	0.420	0.212	0.425
2 models								
Bi-Encoder	0.806	0.880	0.311	0.505	0.310	**0.531**	**0.299**	0.312
Poly-Encoder 16	0.812	0.883	**0.314**	0.529	**0.321**	0.521	0.287	0.518
Poly-Encoder 64	**0.813**	**0.884**	0.309	**0.530**	0.312	0.522	0.289	**0.520**
Poly-Encoder 360	0.809	0.881	0.310	0.490	0.311	0.518	0.283	0.512

TABLE VII. RESULTS OF TRAINING THE BASELINE AND BERT ENCODER DIALOGUE MODELS

2*Model	No Persona		Persona	
	hits@1 personachat	**hits@1 toloka**	**hits@1 personachat**	**hits@1 toloka**
Base-line models				
IR	0.214	0.332	0.410	0.401
KV Profile Memory	0.349	-	0.511	-
Our models				
Bi-Encoder	**0.631**	0.683	**0.634**	0.701
Poly-Encoder 16	0.612	**0.705**	0.621	**0.717**
Poly-Encoder 64	0.625	0.670	0.628	0.716
Poly-Encoder 360	0.615	0.688	0.625	0.711

TABLE VIII. RESULTS OF THE BLENDERBOT MODEL ON THE PERSONA CHAT DATASET

Model	**PPL**	**BLEU-1**
BlenderBot90M	12.59	0.149
BlenderBot2.7B	**10.66**	**0.162**

TABLE IX. RESULTS OF THE BLENDERBOT MODEL ON THE TOLOKA PERSONA CHAT RUS DATASET

Model	**Tokenization**	**PPL**	**BLEU-1**
BlenderBot90M	BPE	92.07	0.036
BlenderBot90M	WP	83.98	0.039
BlenderBot190M	SP	**47.48**	0.054
BlenderBot2.7B	BPE	88.2	0.044
BlenderBot2.7B	WP	86.38	0.050
BlenderBot2.7B	SP	48.28	**0.063**

predicting answers without using information about a person and by 1.2 times when using a person.

2) When testing hypotheses about the possibility of using similarity measures of context vectors and candidates other than dotprod, experimental results show that cosine similarity, Euclidean, and Manhattan distances are less effective for training retrieval models. In addition, it was revealed that dotprod function allows the use of Siamese encoders.

3) The personification of bi, poly, and cross encoders is possible by concatenation of the person and context vector, although it does not provide such a significant increase in performance that can be observed in the basic models.

This fact can be explained by its nonlinearity due to small number of utterances in dialogue with 10-20% of utterances being phatic expressions, that are stylistically and semanticaly similar to each other

One of the limitations to the research in this field is the lack of large data sets with dialogues containing characteristics for each of the persons. In further research, we plan to develop new approaches to augmentation that take into account the style of speech and vocabulary of a person. The research will continue to improve the performance of the refine models and the tandem use of retrieval and refine models. Also human validation of models with and without persona is planned for future research.

REFERENCES

[1] J. Ni, T. Young, V. Pandelea, F. Xue, V. Adiga, and E. Cambria, "Recent advances in deep learning based dialogue systems: A systematic survey," *CoRR*, vol. abs/2105.04387, 2021. [Online]. Available: https://arxiv.org/abs/2105.04387

[2] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, "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 (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2204–2213. [Online]. Available: https://aclanthology.org/P18-1205

[3] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, "A neural network approach to context-sensitive generation of conversational responses," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 196–205. [Online]. Available: https://aclanthology.org/N15-1020

[4] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "Starspace: Embed all the things!" *CoRR*, vol. abs/1709.03856, 2017. [Online]. Available: http://arxiv.org/abs/1709.03856

[5] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1400–1409. [Online]. Available: https://aclanthology.org/D16-1147

[6] H. Sugiyama, M. Mizukami, T. Arimoto, H. Narimatsu, Y. Chiba, H. Nakajima, and T. Meguro, "Empirical analysis of training strategies of transformer-based japanese chit-chat systems," *CoRR*, vol. abs/2109.05217, 2021. [Online]. Available: https://arxiv.org/abs/2109.05217

[7] Z. Lin, Z. Liu, G. I. Winata, S. Cahyawijaya, A. Madotto, Y. Bang, E. Ishii, and P. Fung, "XPersona: Evaluating multilingual personalized chatbot," in *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*. Online: Association for Computational Linguistics, Nov. 2021, pp. 102–112. [Online]. Available: https://aclanthology.org/2021.nlp4convai-1.10

[8] Y. Zheng, G. Chen, M. Huang, S. Liu, and X. Zhu, "Personalized dialogue generation with diversified traits," *CoRR*, vol. abs/1901.09672, 2019. [Online]. Available: http://arxiv.org/abs/1901.09672

[9] A. Madotto, Z. Lin, Y. Bang, and P. Fung, "The adapter-bot: All-in-one controllable conversational model," *CoRR*, vol. abs/2008.12579, 2020. [Online]. Available: https://arxiv.org/abs/2008.12579

[10] S. Bao, H. He, F. Wang, H. Wu, and H. Wang, "PLATO: Pre-trained dialogue generation model with discrete latent variable," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 85–96. [Online]. Available: https://aclanthology.org/2020.acl-main.9

[11] H. Kim, B. Kim, and G. Kim, "Will I sound like me? improving persona consistency in dialogues through pragmatic self-consciousness," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 904–916. [Online]. Available: https://aclanthology.org/2020.emnlp-main.65

[12] Q. Liu, Y. Chen, B. Chen, J.-G. Lou, Z. Chen, B. Zhou, and D. Zhang, "You impress me: Dialogue generation via mutual

[15] H. Song, W.-N. Zhang, J. Hu, and T. Liu, "Generating persona consistent dialogues by exploiting natural language inference," *Proceedings of the AAAI Conference on Artificial Intelligence*, persona perception," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 1417–1427. [Online]. Available: https://aclanthology.org/2020.acl-main.131

[13] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y.-L. Boureau, and J. Weston, "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*. Online: Association for Computational Linguistics, Apr. 2021, pp. 300–325. [Online]. Available: https://aclanthology.org/2021.eacl-main.24

[14] Y. Wu, X. Ma, and D. Yang, "Personalized response generation via generative split memory network," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 1956–1970. [Online]. Available: https://aclanthology.org/2021.naacl-main.157
vol. 34, no. 05, pp. 8878–8885, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6417

[16] E. Dinan, V. Logacheva, V. Malykh, A. H. Miller, K. Shuster, J. Urbanek, D. Kiela, A. D. Szlam, I. Serban, R. Lowe, S. Prabhumoye, A. W. Black, A. I. Rudnicky, J. Williams, J. Pineau, M. S. Burtsev, and J. Weston, "The second conversational intelligence challenge (convai2)," *ArXiv*, vol. abs/1902.00098, 2019.

[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[18] S. Humeau, K. Shuster, M. Lachaux, and J. Weston, "Real-time inference in multi-sentence tasks with deep pretrained transformers," *CoRR*, vol. abs/1905.01969, 2019. [Online]. Available: http://arxiv.org/abs/1905.01969

[19] R. Sennrich, B. Haddow, and A. 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)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://aclanthology.org/P16-1162

[20] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012

[21] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou, "Fast WordPiece tokenization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2089–2103. [Online]. Available: https://aclanthology.org/2021.emnlp-main.160