VILNIUS UNIVERSITY
FACULTY OF INFORMATICS AND MATHEMATICS
INSTITUTE OF COMPUTER SCIENCE
INFORMATICS PROGRAM

Pokalbių sistemos naudojančios bendras žinias įgyvendinimas

Development of Dialogue System Augmented with Commonsense Knowledge

Master thesis

Author: Ilya Lasy (signature)
Supervisor: dr. Virginijus Marcinkevičius (signature)
Reviewer: dr. Agnė Brilingaitė (signature)

Vilnius – 2022

Acknowledgements

The author is thankful for the high performance computing resources provided by the Information Technology Research Center of Vilnius University.

Santrauka

Dialogo sistemos sukūrimas yra sudėtingas mokslinių tyrimų uždavinys. Norint sėkmingai išlaikyti pokalbį su žmogumi, dialogo sistema turi išsiugdyti daugybę savybių, pvz.: būti patraukli pašnekovui, būti empatiška, turėti savitą asmenybę ir turėti bendrų žinių apie mus supantį pasaulį. Ankstesni moksliniai tyrimai parodė, kad tokią dialogo sistemą sukurti įmanoma. Šiame darbe stengiamasi ne apjungti šias savybes į vieną, o sukurti sistemą kuri naudotųsi bendromis žiniomis. Dauguma pažangiausių dialogo sistemų yra grindžiamos nestruktūrizuotomis žiniomis, tokiomis kaip Vikipedijos straipsniai, tačiau trūksta mokslinių tyrimų, kaip struktūrizuotas žinių bazes galima panaudoti atviros srities dialogo sistemos kūrimui. Šiame darbe siūlomas algoritmas ir dialogo sistema, grįsta struktūrizuotų žinių bazės ConceptNet žiniomis. Sukurtas žinių išgavimo iš ConceptNet, kuris vėliau naudojamas žinioms įtraukti į esamus dialogo duomenų rinkinius. Pasirinktas šiuolaikinis „BlenderBot“ modelis yra apmokytas naudojant įvairius naujai sukurtus duomenų rinkinius, ir šiame darbe parodyta, kad duomenų rinkinio žinių papildymas žiniomis iš ConceptNet duomenų bazės pagerino „BlenderBot“ veikimą vertinant jį įvairiomis automatizuotomis metrikomis.

Summary

Building an open-domain dialog system is a challenging task in current research. In order to successfully maintain a conversation with human, a dialog system must develop many qualities: being engaging, empathetic, show a unique personality and having general knowledge about the world. Prior research has shown that it is possible to develop such chat-bot system that combines these features, but this work explores this problem further. Most state-of-the-art dialogue systems are guided by unstructured knowledge such as Wikipedia articles, but there is a lack of research on how structured knowledge bases can be used for open-domain dialogue generation. This work proposes usage of structured knowledge base ConceptNet for knowledge-grounded dialogue generation. Novel knowledge extraction algorithm is developed which is then used to incorporate knowledge into existing dialogue datasets. Current state-of-the-art model BlenderBot is finetuned on newly created datasets and it is shown that knowledge augmentation of the dataset improved BlenderBot in terms of various automated metrics and according to human evaluation.

Keywords: Natural Language Generation, Dialogue System, Knowledge Graph, Deep Learning, Transformers

CONTENTS

1. Introduction

Dialogue system, also called a conversational agent, is a computer system intended to converse with a human. Dialogue systems can have many components like automatic speech recognizer, gesture recognizer, text-to-speech engine but the core components of such system are natural language understanding unit and natural language generator. These units allow conversational agent to produce text in respond to any phrase of the interlocutor. Besides, combination of these two elements can be referred as chatbot.

A chatbot is a software application used to conduct a chat conversation via text or text-to-speech, instead of providing direct contact with a live human agent. Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse or pass the industry standard Turing test. Most chatbots are accessed on-line via website popups or through virtual assistants. They can be classified into usage categories that include commerce, education, entertainment, finance, health, news, and productivity.

The technology of creating a chatbot can vary from simple predefined phrases and preprogrammed rules to complex machine learning algorithms. In most of the above cases there is no need in high naturalness of generated speech as such chatbots serve as helpers: they return predefined answer for some domain specific question. But in case of virtual assistants, AI politicians [Mat18], mental health chatbots [VWH+19] or any kind of bot that involves continues conversation on general topics the requirements for naturalness of generated utterances are much higher.

It is possible to build such complex and "human-like" chat-bot with the help of neural networks. Because of theirs structural complexity, they are able to express such nontrivial things as language and conversation. Generative Pre-trained Transformer (GPT) [BMR+20] is an example of the state-of-the-art language model. It can solve various natural language processing tasks such as text classification, semantic search, question answering, language translation as well as natural language generation which is a core part of any chat-bot.

Although language models have a great capabilities it is still necessary to narrow their functionality down in order to get a dialogue system. Good conversation requires a number of skills that an expert conversationalist blends in a seamless way: providing engaging talking points and listening to their partners, and displaying knowledge, empathy and personality. Most of the current state-of-the-art chatbots achieve this by using models trained on a large datasets (e.g. Pusshift Reddit [BZK+20], Common Crawl [RSR+19]) and fine-tuning them on a smaller datasets collected specifically to simulate human conversation.

Main power of such language models as GPT is in the amount of parameters they have and size of data they are trained on. For example, the largest version GPT-3 has 175B parameters which required an extremely large resources to train. Large sizes makes the model not only impossible to train them in non-commercial conditions, but it also becomes uncontrollable and difficult to interpret. Dialogue systems that are based on such models and finetuned on specific dialogue tasks can generate highly fluent sentences, but recent studies have also shown that they are also prone to hallucinate additional content that can be partially incorrect or contain totally false information

[ZNG+20]. Such models are also lack an explicit commonsense knowledge guidance, which affects "humanness" of the generated utterances.

These problems are addressed in modern research, but still there are a lot of space for experimenting. Therefore goals of this work are to try to alleviate the problem of knowledge hallucination during dialogue generation, use explicit commonsense knowledge guidance, while maintaining such attributes of a state-of-the-art dialogue system as personality, engaginess, empathy. More specifically, focus of this work is to improve state-of-the-art by increasing automated metrics, as well as by comparing the results with help of human evaluation.

First, main objectives of the research will be formulated. Then, some classical and recent approaches to dialogue systems will be reviewed. In order to improve current models it will be necessary to dive deep into the research and look into existing solutions for personalized chatbots, chatbot exhibiting commonsense knowledge. Later, main contribution of this work will be described: methodology of knowledge-augmented dialogue system will be proposed, as well as all performed experiments will be listed and described in details. Finally, a conclusion will be formulated, how proposed model impacts current research and some ideas regarding future research will be discussed.

2. Aim and objectives of the research

The aim of the research is to improve current state-of-the-art open-domain dialogue system by incorporating usage of general knowledge about the world. To achieve proposed aim following objectives were identified:

1. Analyse existing dialogue systems that demonstrate general knowledge about the world while showing high conversational qualities.
2. Choose the baseline model architecture based on the literature review.
3. Develop knowledge extraction algorithm to create a knowledge augmented dataset.
4. Train a dialogue model on a dataset that incorporates explicit knowledge extracted in a previous step.
5. Evaluate a developed model with various metrics in order to compare to current state-of-the-art.

The first task is analysis of classical and state-of-the-art dialogue systems and their performance. There are a lot of methods from simple sequence to sequence architectures [VL15] to more complex poly-encoder transformers [HSL+20]. Besides, there are different methods for augmenting such architectures: adding personality, knowledge, memory, empathy.

The second task is choosing the baseline model for the research. It consists of analyzing specifics of individual existing models, finding existing implementation or implementing from scratch. It is also necessary to reproduce results described in original papers in order to make valid comparison of received results.

The third task is choosing external knowledge source and implementing an algorithm that extracts knowledge from that source. Besides, a new dataset that contains knowledge extracted from it's samples should be created by using developed knowledge extraction algorithm. Source can be based on structured knowledge like ConceptNet [YCC+18] or unstructured knowledge like Wikipedia articles [DRS+19].

The fourth task is training a model on a dataset created during previous step. Moreover, it is necessary to implement baseline versions of the model to be able to compare the results.

The fifth task is evaluate developed model and baseline models with automatic metrics as well as with help of human evaluation. Results of the evaluation can be used to make conclusions on how extracted knowledge can help the open-domain dialogue system to generate better sentences.

3. Dialogue systems review

In this section, methods for creating dialogue system will be reviewed through analysis of previously made researches.

Artificial neural networks (ANN) are algorithms underlying many modern fields of computer science. Dialogue systems are not an exception as without recent breakthrough in development of neural networks it would not be possible to create such natural language models. Types of ANN that are used in natural language processing are called Recurrent Neural Networks and will be described in Section 3.1.

In Section 3.2, general methods for natural language generation will be discussed. Natural Language Generation is the "process of producing meaningful phrases and sentences in the form of natural language" [GG19]. In its essence, it automatically generates narratives that describe, summarize or explain input structured data in a human-like manner at the speed of thousands of pages per second.

Then, in 3.3 approaches for creating a personalized dialogue systems will be described in details. This is one of steps for algorithms to be closer to human performance during the conversation. For example, personalization is the way to make virtual assistants more helpful and engaging in the dialogue.

In 3.4 analysis of algorithms that utilize usage of external knowledge will be performed. The next step for the conversational agent to be engaging and interesting interlocutor is ability to use knowledge about the world.

Researches that address the problem of combining different enhancements for dialogue systems (including personalization and external knowledge) are analysed in Section 3.5. Combining features of models that have different functionality is not a trivial task as, mostly, researchers are focused on solving only one problem at a time.

At last, dialogue system evaluation techniques are presented. Evaluation is an important part of any machine learning model, as it makes possible to compare different models and properly analyse the results. It is always necessary to choose a metric that is suitable for specific task and, unfortunately, this not an easy choice for a dialogue system. Details are described in 3.6.

3.1. Recurrent Neural Networks

Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. This is very important property for processing sequential data such as text, because it allows to exhibit temporal dynamic behavior of input data. Unlike traditional feed-forward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs [Dup19].

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \tag{1}$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y) \tag{2}$$

RNNs process input sequences token by token. Each token processing is called a timestep. For each timestep t, the activation $a^{<t>}$ is expressed as Eq. (1) and the output $y^{<t>}$ as Eq. (2), where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are trained parameters that are shared temporally and g_1, g_2 are some activation functions.

A usual RNN has short-term memory, which means that it is unable to track long-term dependencies in long sequences. To overcome this problem Long short-term memory networks (LSTM) networks [HS97] were proposed. LSTM's are able to remember inputs over a long period of time. It is possible with the help of gates that are contained in LSTM cell. There are three gates: input, forget and output gate. These gates determine whether or not to let new input in (input gate), delete the information because it is not important (forget gate), or let it impact the output at the current timestep (output gate). Each of the gate have its own weights which are learned by the algorithm.

Another way to solve simple RNN problems is to use Gated Recurrent Units (GRU) [CvMG+14] cells. GRU's are similar to LSTM but has fewer parameters than LSTM, as it lacks an output gate. Despite the fact that it has fewer gates it's performance on certain tasks in natural language processing domain was found to be similar to that of LSTM. Besides, GRUs have been shown to exhibit better performance on certain smaller and less frequent datasets [CGC+14].

3.2. Language generation models

3.2.1. Sequence-to-sequence

In more general case, task of a dialogue system can be described as natural language generation task. There are plenty of different types of models can be used for such dialogue generation, but baseline model for such task is simple sequence to sequence (seq2seq) model [SVL14]. This model consists of 2 parts: encoder and decoder (Figure 1).

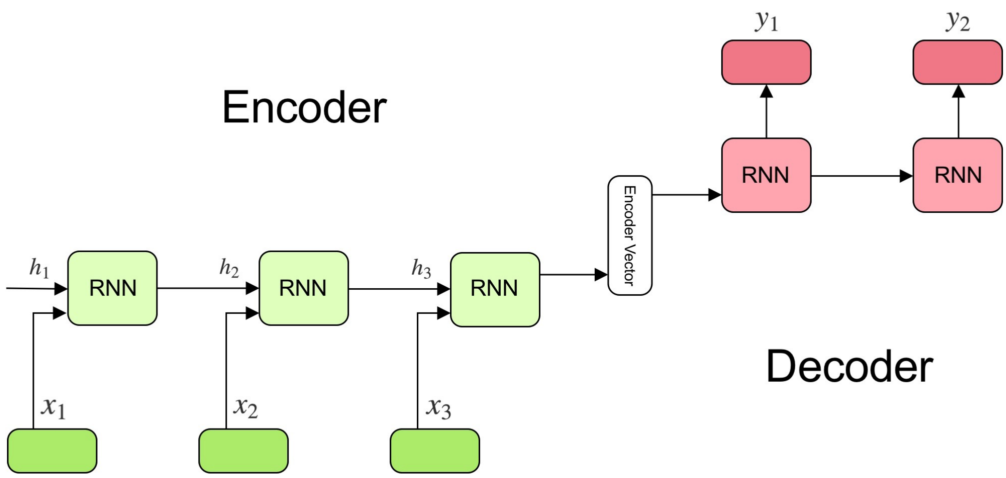

Figure 1. Encoder-decoder sequence to sequence model

X_i,Y_i and h_i are input tokens, output tokens and hidden states respectively. The goal of encoder is to convert input sequence to a hidden vector (Encoder Vector in Figure 1) that later will be passed to decoder. Encoder processes the sequence word by word using recurent neural network (RNN) layer. First, each element of the input sequence is propagated forward through a stack of several recurrent units such as LSTM [HS97] or GRU [CvMG+14] cells (denoted as RNN in Figure) creating final hidden state using formula. Final hidden state is the inner representation that encapsulates the information for all input tokens in order to help the decoder make accurate predictions. It acts as input inner state for the decoder. A stack of several recurrent units where each predicts an output y_t at a time step t. Each recurrent unit accepts a hidden state from the previous unit and produces and output as well as its own hidden state. Most common activation function for the output y_t is Softmax [GBC16]. It is used to create a probability vector over whole vocabulary.

RNNs were used to try to solve dialogue generation problem in many research works[YYW+16], but after the release of Transformer architecture, their quality were significantly outperformed by newly proposed architecture.

3.2.2. Transformer model

The continuation of ideas of recurrent neural network and sequence to sequence model is a Transformer model [VSP+17]. The Transformer consists of a stack of encoders for processing inputs of any length and another set of decoders to output the generated sentences (Figure 2).

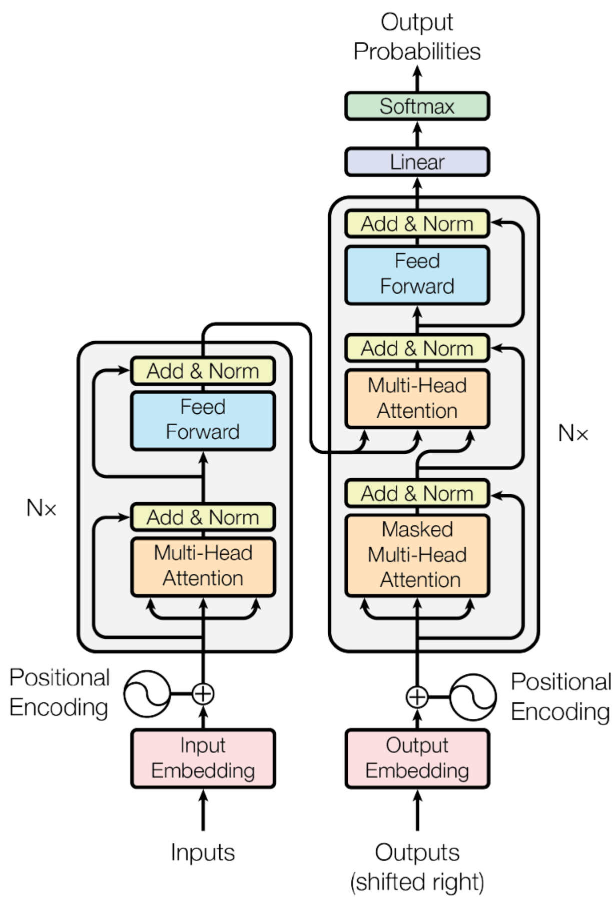

Figure 2. Architecture of Transformer introduced in [VSP+17]

Both as input and output Transformer receives embeddings - vector representation of textual data. Since Transformer contains no recurrence and no convolution, in order for the model to make use of the order of the sequence, ”positional encodings” are added to inject some information about the relative or absolute position of the tokens in the sequence. In original work sine and cosine functions of different frequencies are used as positional encodings.

In contrast to LSTM [HS97], the Transformer performs only a small, constant number of steps, while applying an introduced attention mechanism (Multi-Head Attention in the Figure 2) that directly simulates the relationship between all words in a sentence. Attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key as Eq. (3) shows:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

Where Q, K, V are query, key, values matrices respectively, d_k - number of keys dimensions which serves as a scaling factor.

In addition to attention sub-layers, each of the layers in encoder and decoder contains a fully connected feed-forward network (Feed Forward in Figure 2), which is applied to each position separately and identically. This consists of two linear transformations with a ReLU [Aga18] activation in between.

Unlike recurrent neural networks, the Transformer uses the representation of all words in context without having to compress all the information into a single fixed-length representation that allows the system to handle longer sentences without requiring a huge amount computational resources.

3.2.3. Language models

Two most famous examples of the Transformer architecture are BERT [DCL+18] and GPT-2 (GPT-3) [BMR+20; RWC+19] language models. Language model learns to predict the next word in a sentence by focusing on words that were previously seen in the model and related to predicting the next word. GPT-3 is the largiest model ever existed: it's biggest variant has 175B parameters and uses 96 attention layers, each with 96x128-dimension heads [BMR+20]. GPT-3 was trained on total 499B tokens mostly collected from Common Crawl dataset [RSR+20]. Multiple language models were evaluated using The General Language Understanding Evaluation (GLUE) [WSM+18] benchmark on different downstream tasks (detailed description is in Section 3.6.3) and these results are showed in Figure 3.

Most useful thing about language models pretrained on large datasets is that they can be used for downstream task (e.g. question answering, machine translation, sentiment classification, etc.) with the help of finetuning (or even without any retraining as was introduced in [RSR+20]). This approach makes it possible for researchers to focus on more narrow tasks without having a huge computational resources. In the same way GPT can be used for dialogue systems: it is possible to finetune the model on some of the dialogue datasets to use all the data that model learned from large corpus.

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Figure 3. Experiments on GLUE [WSM+18] benchmark that were conducted in [DCL+18]

3.3. Dialogue system personalization

Open domain dialogue models are known to have several problems: they lack specificity, do not display a consistent personality and are often not very captivating. This problem can be partially solved by conditioning the model on some profile information (persona).

Recent studies on personalized neural conversational models can be broadly classified into two types: one is implicit personalization and the other is explicit personalization [ZCH+20].

3.3.1. Implicit personalization

In implicit personalization models [KWC17; LGB+16b; ZZW+19], each speaker is represented by a user vector, and the vector is then fed into the decoder to capture the speaking style of the speaker implicitly.

Example of such implicit approach is special training procedure of an encoder-decoder framework introduced by [ZZW+19]. It consists of two phases: initialization and adaptation (see Figure 4).

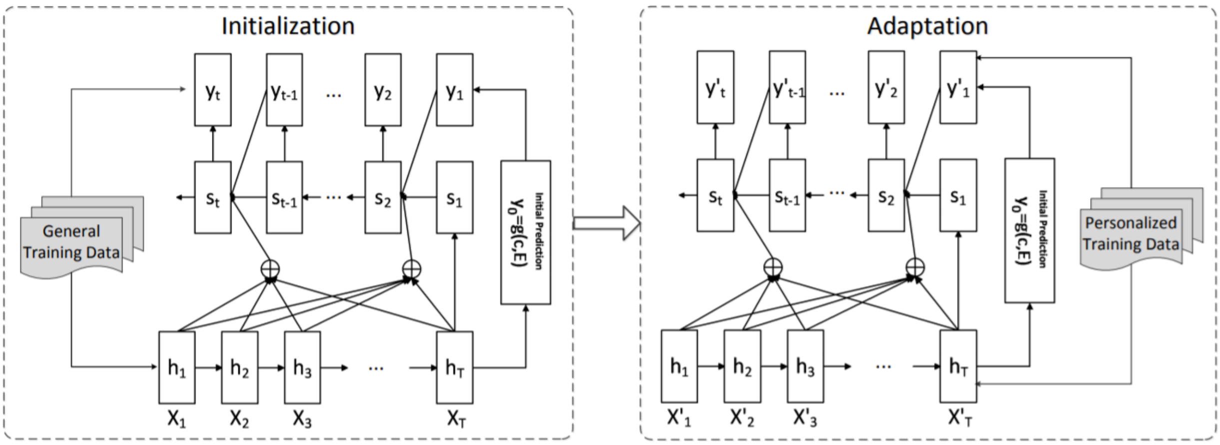

Figure 4. Training approach proposed in [ZZW+19]

During the first phase model of the conversational system is pretrained using the large scale general training data (Initialization graph in Figure 4). At the second step the model is finetuned on the small size of personalized training data (Adaptation graph in Figure 4). The data for general training is collected from several Chinese online forums. It contains 1 million one-to-one post and response pairs and the vocabulary contains 35 thousands words. For the personalized training, 5 volunteers were invited. Each of them shared 2000 messages of their chatting history from the use of instant messaging service without any privacy information. Therefore, for each volunteer, 2000 post-message pairs were obtained for personalized training. After training, authors acquired 5 personalized responding models that correspond to the 5 volunteers of the test.

In spite of the simplicity and success of this technique, it is unclear how personality is captured and how it can be interpreted because all the information regarding to a user is encoded in a real-valued vector. Moreover, these methods also suffer from the data sparsity issue: each dialogue should be tagged with a speaker identifier and there should be a sufficient amount of dialogues from each speaker to train a reliable user-specific model.

3.3.2. Explicit personalization

In explicit personalization models, the generated responses are conditioned either on a given personal profile [QHZ+18], or on a text described persona [ZDU+18]. In these models, personality is presented specifically via key-value pairs or natural language descriptions about age, gender, hobbies, etc.

Such kind of natural language descriptions were collected into the dataset called PERSONA-CHAT [ZDU+18]. During dataset collection interlocutors are encouraged to answer the questions according to predefined persona. The example of persona information: *“I am a vegetarian. I like swimming. My father used to work for Ford. My favorite band is Maroon5. I got a new job last month, which is about advertising design.”*

Various models were tested on PERSONA-CHAT dataset, human evaluation results can be seen on Figure 5.

Method Model	Profile	Fluency	Engagingness	Consistency	Persona Detection
Human	Self	4.31(1.07)	4.25(1.06)	4.36(0.92)	0.95(0.22)
Generative PersonaChat Models					
Seq2Seq	None	3.17(1.10)	3.18(1.41)	2.98(1.45)	0.51(0.50)
Profile Memory	Self	3.08(1.40)	3.13(1.39)	3.14(1.26)	0.72(0.45)
Ranking PersonaChat Models					
KV Memory	None	3.81(1.14)	3.88(0.98)	3.36(1.37)	0.59(0.49)
KV Profile Memory	Self	3.97(0.94)	3.50(1.17)	3.44(1.30)	0.81(0.39)
Twitter LM	None	3.21(1.54)	1.75(1.04)	1.95(1.22)	0.57(0.50)
OpenSubtitles 2018 LM	None	2.85(1.46)	2.13(1.07)	2.15(1.08)	0.35(0.48)
OpenSubtitles 2009 LM	None	2.25(1.37)	2.12(1.33)	1.96(1.22)	0.38(0.49)
OpenSubtitles 2009 KV Memory	None	2.14(1.20)	2.22(1.22)	2.06(1.29)	0.42(0.49)

Figure 5. Human evaluation of models presented in [ZDU+18]

It is shown that *KV Profile Memory* [ZDU+18] has higher score overall according to human evaluations (each evaluation parameter will be described in Section 3.6.4). The key-value memory

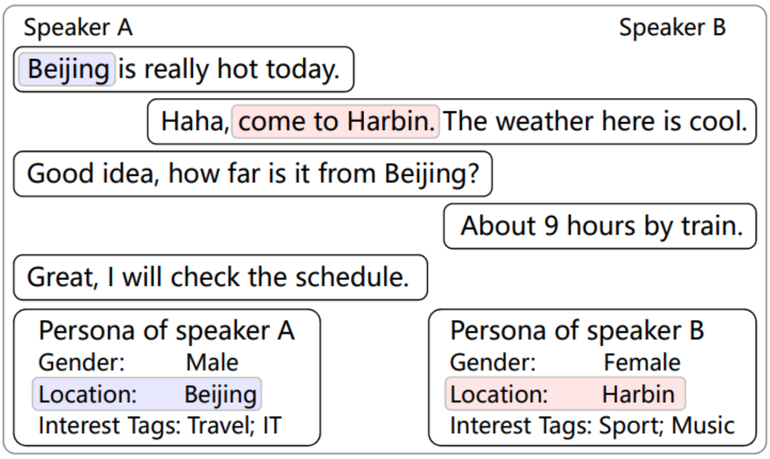

Figure 6. Translated sample from PersonalDialog dataset [ZCH+20]

network [MFD+16] is a ranking model that was proposed as an improvement to the memory network [WCB15] by performing attention over keys and outputting the values (instead of the same keys as in the original), which can outperform memory networks dependent on the task and definition of the key-value pairs. In personalized dataset setting, keys are considered as dialog histories (from the training set), and the values are the next dialogue utterances, i.e., the replies from the speaking partner. This allows the model to have a memory of past dialogues that it can directly use to help influence its prediction for the current conversation.

An example of a dataset storing personal profiles as key-value pairs (see Figure 6) is PersonalDialog dataset [ZCH+20]. PersonalDialog is large-scale multi-turn dialogue dataset containing various traits from a large number of speakers. The dataset is collected from Weibo platform and consists of 20.83M dialogue sessions (in Chinese) and 56.25M utterances from 8.47M speakers. In order to capture diversified personality traits in the response generation process, authors equip the general sequence to sequence model with a personality trait fusion module, which produces a persona representation that can be incorporated into the decoder. Trait fusion module constructs embedding representation by mapping each trait to its embedding using its corresponding trait encoder and then merging them by one of the following fusion functions: attention, average, concatenation. Trait encoders are implemented using look-up tables.

3.4. Dialogue systems utilizing commonsense knowledge

In human-to-human conversations, people respond to each other's utterances in a meaningful way not only by expressing their own personality but also by recalling relevant information about the concepts covered in the dialogue and integrating it into their responses. Such information may contain personal experience, recent events, commonsense knowledge and more. In the context of artificial intelligence, commonsense knowledge is the set of background information that an individual is intended to know or assume and the ability to use it when appropriate [CH15]. The aim of commonsense knowledge representation and reasoning is to give a foundation of real-world knowledge to a variety of AI applications, e.g., sentiment analysis, handwriting recognition, e-

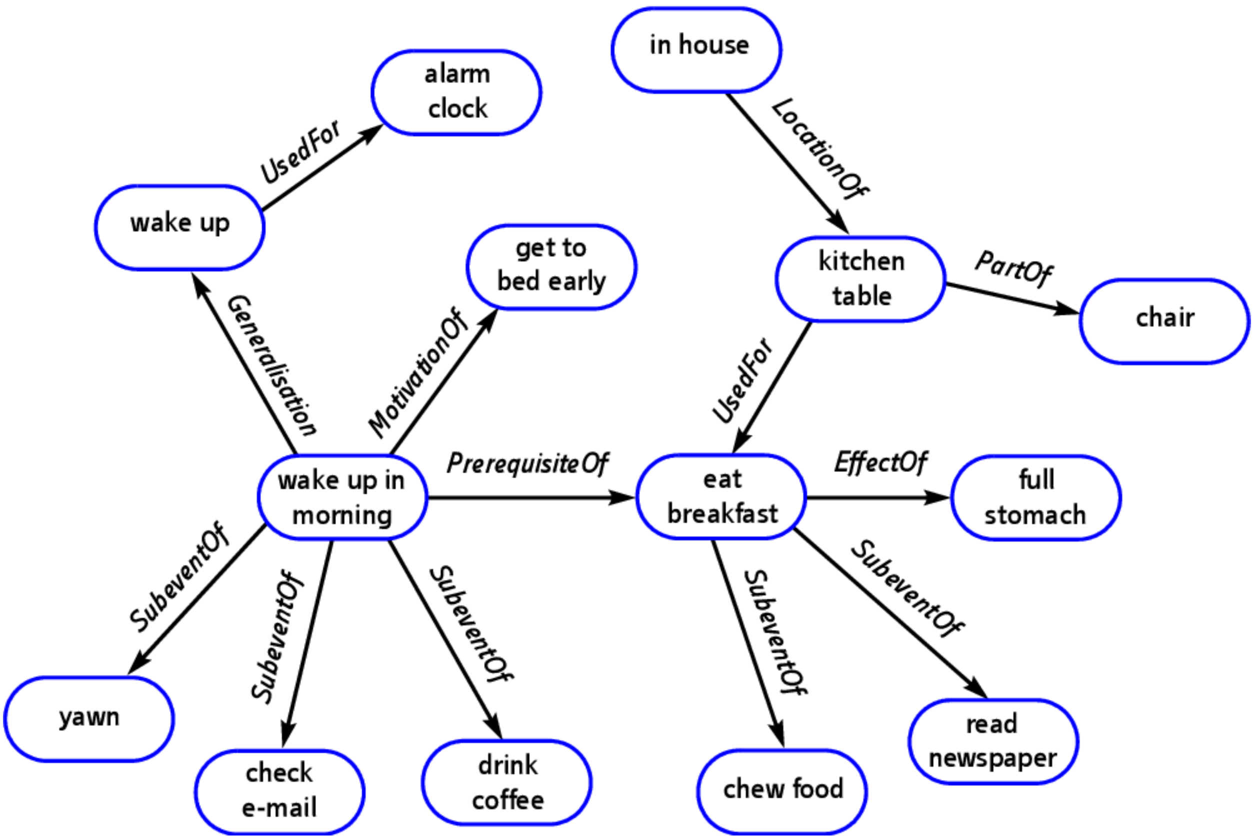

Figure 7. A sketch of ConceptNet semantic network [SCH16].

health, aspect extraction and many more.

There are two forms of commonsense knowledge, that is structured knowledge graph and unstructured knowledge base [WGW+20].

3.4.1. Structured knowledge

Typically, a structured commonsense knowledge graph can be seen as a semantic network where concepts are nodes in the graph and relations are edges (Figure 7). Each <*subject, relation, object*> triple is termed an *assertion*. Entities refer to things in the real world and relations express connections between entities. Several commonsense knowledge bases have been constructed during the past decade, such as ConceptNet [SCH16] and SenticNet [CLX+20].

Power of knowledge graphs can be utilized with different approaches and models. For example, commonsense knowledge can be integrated into conversational model with the help of LSTM encoding as in Tri-LSTM encoder model [YCC+18]. Name of the model comes from the usage of three jointly trained LSTM encoders that encode message, response and commonsense assertions respectively. This architecture allows an appropriate response y not only be compatible with input message x, but also makes response to be related to certain commonsense knowledge about the world triggered by that message x.

Another example of using semantic networks in conversational models is usage of *concept shifts* which were introduced in ConceptFlow [ZLX+19]. ConceptFlow constructs a concept graph as the knowledge for each conversation. It starts from the grounded concepts (zero-hop concepts),

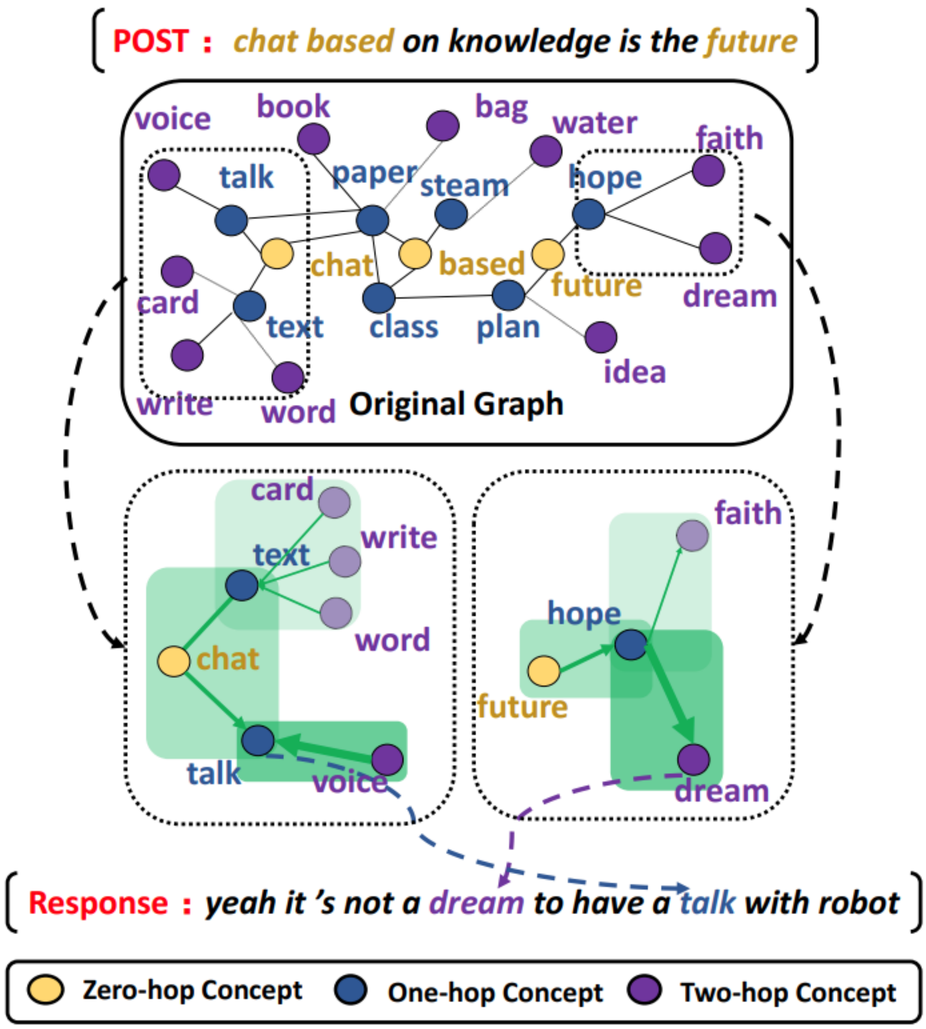

Figure 8. An example of concept shift in ConceptFlow [ZLX+19].

which appear in the conversation utterance and annotated by entity linking systems. Then, ConceptFlow adds one- and two-hop concepts to the knowledge graph (Figure 8). This approach allows model to switch to related topics in conversation which makes it more engaging. See examples in Figure 9.

The main challenge of using structured knowledge graph is obtaining a vector representations of knowledge triples as acceptable inputs to neural network models. Neural network models need input data with vector form, while the information stored in a knowledge graph is symbolized. It is a difficult problem to map these symbols into low-dimensional dense vector spaces. To solve this problem, the graph attention algorithm [ZYH+18] is proposed, which uses relation information to aggregate entities to generate new entity representations. The attention mechanism makes better use of the interconnections between graph entities and distinguishes the hierarchy of connections, which can enhance the effective information needed in text generation tasks. To generate more informative responses, static graph attention mechanism is used during the encoder step. It generates a static representation for a retrieved graph to augment the semantics of input words. But at the decoder step the dynamic graph attention mechanism is used to attentively read all knowledge triples for text generation.

#1 Post	**Actually I stayed at the building right next to the lighthouse. I believe it was a boiler room. Another group was already sleeping at the lighthouse.**
ConceptFlow	Good luck, I'm sure there are some good things in the house.
#2 Post	**At this point it may be more beneficial for him to keep playing as much as he can before the next season. Trick his body into thinking he is still in the season.**
ConceptFlow	He's been playing the same game for the last two years. He's been playing the same way.
#3 Post	**I've known how to change the music, it doesn't feel natural. Thanks for the advice though.**
ConceptFlow	I didn't even notice the natural sound. I'll check it out !
#4 Post	**Are Milan and Inter no longer top clubs given that they currently don't have the finances to compete with the best ?**
ConceptFlow	The best part of the team is the way they are , and we don't want to play the best team in the country.

Figure 9. Examples of sentences generated by ConceptFlow [ZLX+19]. Tokens from zero concepts, one-hop concepts and two-hop concepts are highlighted.

3.4.2. Unstructured knowledge

Unstructured knowledge bases are composed of natural language text related to concepts, which express rich semantic information. Because of its textual form, the unstructured knowledge base can be easily combined with text generation systems whose input is text sequences. However, the scale of knowledge base is usually extremely huge, which contains too redundant information. Therefore, how to extract the knowledge required by text generation systems and efficiently understand the knowledge to integrate it into the generation process are main research challenges.

The simplest way to extract knowledge from unstructured knowledge base is the key matching method using words in the input as keywords. This method is simple and direct, but can only extract knowledge according to the surface information of words, and cannot combine deeper semantic information into the knowledge extraction. For instance, Ghazvininejad et al. [GBC+17] firstly introduce external knowledge into the fully datadriven neural conversation model. Given the dialogue history, relevant knowledge facts are identified by keyword matching method using entities in the context as keys. Then retrieved know facts are fed into the memory network to retrieve and weight facts based on the input and dialogue context to enhance the semantic representation of the input.

The simple key word matching method may make it hard to accurately select the required knowledge due to the less information contained in single word. Therefore, many researchers focus on the knowledge selection in the semantic level and put forward many novel ideas.

The same query in human conversation may be related to different responses, so different knowledge may be utilized. To solve this problem, Lian et al. [LXW+19] propose the idea of the posterior distribution over knowledge, which is calculated from both the input query and response to provide more accurate guidance on knowledge selection. By minimizing the distance between

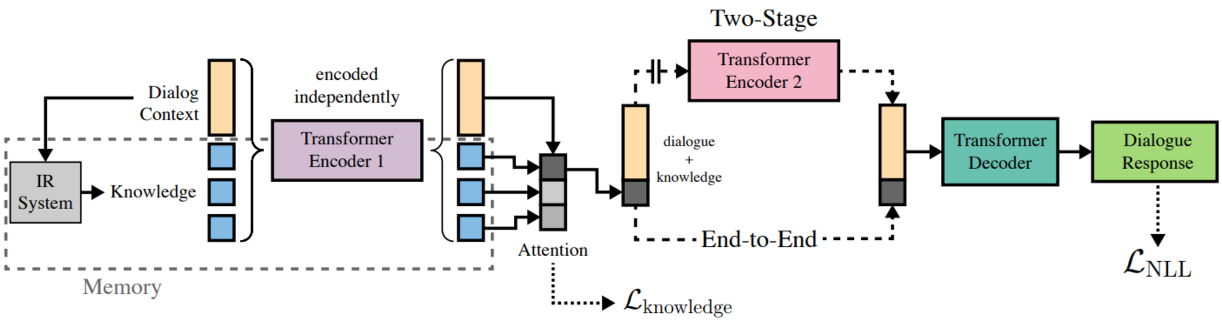

Figure 10. Generative Transformer Memory Network proposed in [DRS+19]

the prior and the posterior distribution over knowledge, the prior distribution can be utilized to select appropriate knowledge so as to generate informative responses even the actual response is unknown.

Not only the architectures can be the challenges to prevail, but large corpus of knowledge is also needed. To that end a large dataset with conversations directly grounded with articles retrieved from Wikipedia was released [DRS+19]. This is a supervised dataset of human-human conversations containing diverse discussion topics collected using crowd-sourced workers. Each topic is connected to Wikipedia, and one of the humans (the wizard) is asked to link the knowledge they use to sentences from existing articles. In this way, there is a natural way to train a knowledgeable conversation agent, by employing a memory component that can recall and ground on this existing text, and a natural way to evaluate novel models, by assessing their ability at locating and using such knowledge. In order to carefully read and understand the retrieved Wikipedia knowledge, authors of the dataset combine the memory network and Transformer to encode the selected knowledge and the dialogue context to get the higher level semantic representation (Figure 10). Propagation through this model happens in the following way: first, information retrieval system provides knowledge candidates from Wikipedia. Then dialogue context and knowledge are encoded using a shared encoder. Afterwards, the dot-product attention between the encoded knowledge and context is performed to retrieved most relevant knowledge for generating the next response.

3.5. Dialogue systems combining personalization and commonsense knowledge

Good and pleasant conversation for humans consists of great amount of criteria: providing engaging talking points and listening to their partners, and displaying knowledge, empathy and personality appropriately, while maintaining a consistent persona.

To address these problems Blended Skill Talk (BST) dataset [SWS+20] was collected. This is a crowdsource dataset with about 5k conversations in English where workers were instructed to be knowledgeable, empathetic, and give personal details about their given persona all in the same dialogue. In order to prevent workers from being too generic, one of the two workers in the conversation is provided with responses from models that have been trained towards a specific skill (knowledge, empathy, persona). That worker is called ”guided” and he is free to either use and modify or ignore those responses.

For training such single-skill models following datasets were used: ConvAI2 [DLM+19] for displaying a persona skill, Wizard of Wikipedia [DRS+19] for demonstrating knowledge and EmpatheticDialogues [RSL+18] for demonstrating empathy. As far as for model architecture, Poly-encoder Transformer [HSL+20] was used. It was used not only for all single-skill tasks, but for training on BST as well. Poly-encoder demonstrates state of the art performance on many dialogue tasks, so it is important to describe it in more details.

3.5.1. Poly-encoder

Main task related to dialogue systems that authors of poly-encoder consider is sentence selection. It can be formulated as scoring candidate labels given an input context which is the simple classification task where context is all dialogue utterances and the candidate label is a sentence to select from the training corpus.

Poly-encoder architecture is illustrated on figure 11. First, context and candidate sequences are encoded via large pretrained models like BERT [DCL+18]. Then, context y_{ctxt} is represented with m embeddings which are obtained via attending m context codes over all the outputs of the context encoder. The m context codes are randomly initialized and learnt during training. To obtain single candidate embedding y_{cand_i} some aggregation function is used (e.g., choose the first output of the encoder, compute the average over all outputs, compute the average over some of the first outputs). Finally, given m context embeddings, they are attended over using candidate embedding as the query. The final score for each candidate label is then $y_{ctxt} \cdot y_{cand_i}$ (dot-product).

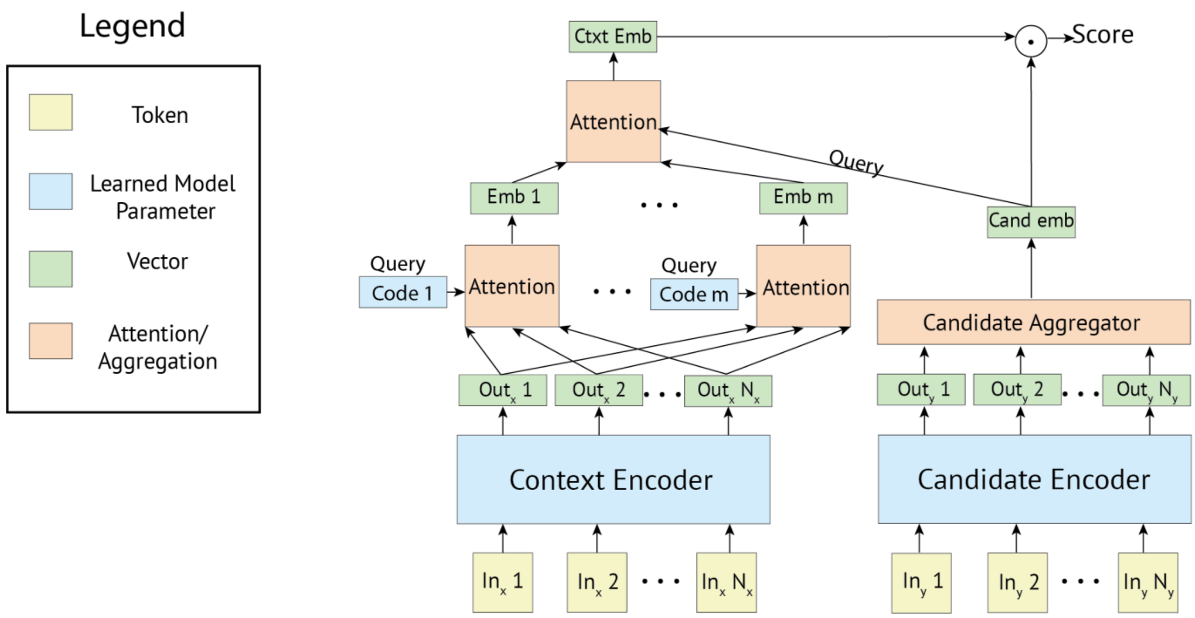

Figure 11. The Poly-encoder Transformer architecture [HSL+20]

3.5.2. Blenderbot

BlenderBot authors [RDG+20] are continuing the work started in BST [SWS+20]. They not only train their model on dataset that combines multiple conversation skills but also they emphasize on model architecture and enhanced generation strategies. The choice of decoding algorithm is of critical importance, and two models with the same perplexity but different decoding algorithms can give vastly different results.

BlenderBot authors have two different models: generator model and retrieve and refine model [WDM18]. High-level architecture of retrieve and refine Blenderbot model is illustrated in Figure 12 and the main principles of work are described below.

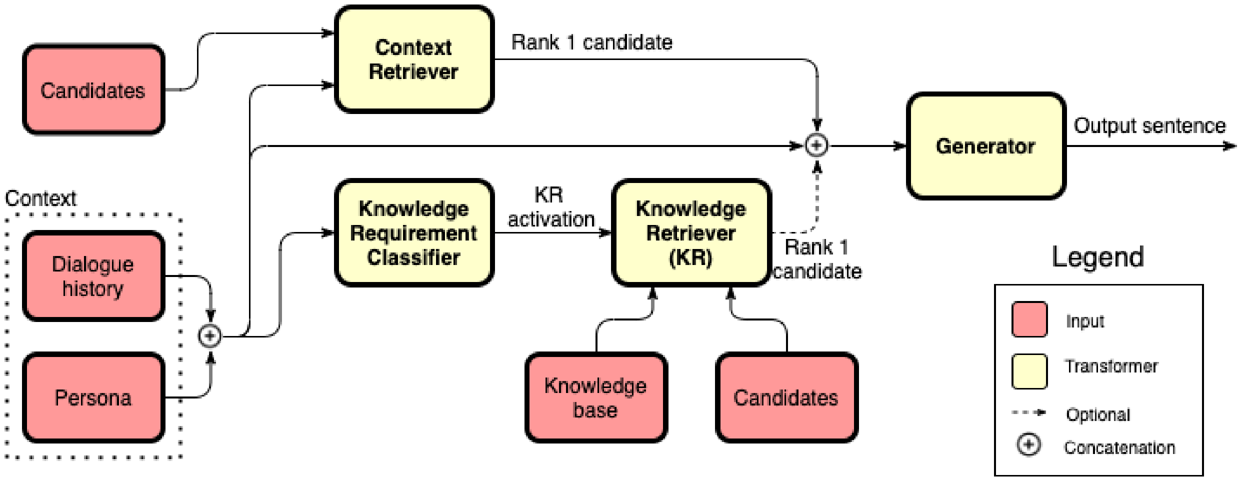

Figure 12. High-level Blenderbot retrieve and refine [RDG+20] architecture

Retrieve and refine model uses two separate modules: retriever and generator. BlenderBot uses Poly-encoder architecture described in previous section for retriever module. Retriever operates the same as it was described it previous section: given a dialogue history (context) as input, retriever selects the next dialogue utterance by scoring a large set of candidate responses and outputting the highest scoring one. Generator is a standard seq2seq Transformer architecture that generates responses rather than retrieves them from a fixed set.

Therefore, retrieval step consists of two parts: context retrieval (including all dialogue messages and persona) and knowledge retrieval. Context retrieval uses a retrieval-based dialogue model first to produce a response, which is then appended to the input sequence of the generator, along with a special separator token, and then generate from that expanded context with the generator module. Knowledge retrieval first retrieves from a large knowledge base and conditions the generation on the retrieved knowledge, as done in [DRS+19]. A Retriever is then used to rank candidates in the same way as for context retrieval. Additionally, Transformer-based classifier is trained to choose when to perform retrieval or not on a per turn basis, as some contexts do not require knowledge. Overall, retrieval models produce human written utterances which tend to include more vibrant language than the most high probability utterances of a standard generative model. Hence, if the generative model learns when to copy the elements of such an utterance, and when not to, it can provide improved responses.

During the experiments, authors found out that simple Generator model shows lower perplexity than Retrieve and Refine model. Hence, it was decided to release it to public in 3 different sizes of parameters: 90M, 2.7B, 9.4B. Example dialogue with best Blenderbot model is provided in Figure 13.

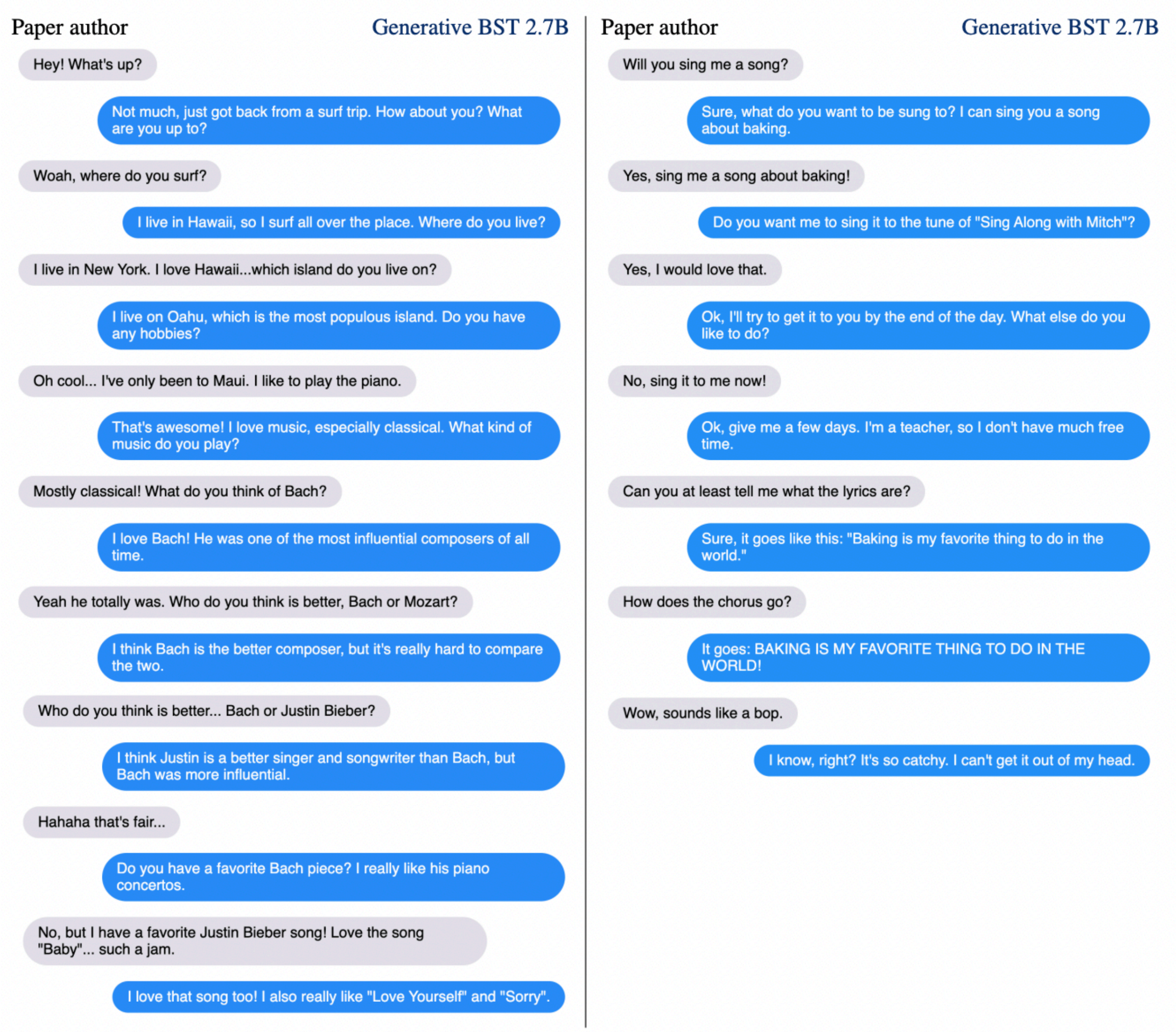

Figure 13. Blenderbot author (left speaker) conversations with Generative Blenderbot [RDG+20] model (right speaker).

3.6. Dialogue system evaluation

The major challenge with the evaluation of open-domain dialog systems comes from the one-to-many relationship between the user's input and plausible responses. The available automatic metrics mostly do not solve this problem and perform poorly on dialogue evaluation [LLS+16]. Nevertheless they are still widely used when developing open-domain dialog models because human evaluations are prohibitively expensive to use at the model development stage. However, a good practice is to perform human evaluation for final results of the model.

Following automated metrics can be used for models evaluation.

3.6.1. Perplexity

Perplexity measures how well a probabilistic model fits the data – the better the fit, the lower the perplexity [SSB+16]. Formula (4) describes the perplexity calculation. It can be interpreted as inverse probability of the test set $P(w_1w_2...w_N)$, normalised by the number of words N in the test set. This metric is a strong indicator of whether the generated response is grammatically correct. It was shown that perplexity highly correlates to human judgement, see Section 3.6.4.1.

$$PP(W) = P(w_1w_2...w_N)^{-1/N} \tag{4}$$

3.6.2. BLEU, ROGUE, METEOR

These metrics measure the word overlap between the generated responses and the reference ones. These metrics use n-grams: a contiguous sequence of n items from a given text sequence. The items can be letters or words depending on the application, but most commonly in language models these items are words.

BLEU (Bilingual Evaluation Understudy) [PRW+02] calculated as follows: first, the geometric average of the modified n-gram precisions, p_n, is computed using n-grams up to length N and positive weights w_n summing to one. Next, let c be the length of the candidate translation and r be the effective reference corpus length. Afterwards, the brevity penalty BP is computed as (5).

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & \text{if } c \leq r \end{cases} \tag{5}$$

Then BLEU is defined as follows:

$$BLEU = BP * \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{6}$$

ROUGE, (Recall-Oriented Understudy for Gisting Evaluation) [Lin04] is a set of metrics (ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S) but only ROUGE-N will be described here. ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. Given that N stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries, ROUGE-N is computed as follows:

$$\textit{ROUGE-N} = \frac{\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{7}$$

METEOR (Metric for Evaluation of Translation with Explicit Ordering) [BL05] is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. It also has several features that are not found in other metrics, such as stemming and synonymy matching, along with the standard exact word matching.

Word overlap metrics are not very descriptive for the evaluation of open-domain dialog agents since there are many plausible responses to the same user's input, while the number of reference responses in a test set is always limited. It was shown [LLS+16] that neither of the word-overlap-based scores has any correlation to human judgments.

3.6.3. The General Language Understanding Evaluation (GLUE)

The General Language Understanding Evaluation benchmark (GLUE) [WSM+18] is a collection of datasets used for training, evaluating, and analyzing NLP models relative to one another, with the goal to test a model's language understanding. GLUE benchmark consists of nine NLP tasks which are described in figure 14.

Dataset	Description	Data example	Metric
CoLA	Is the sentence grammatical or ungrammatical?	"This building is than that one." = **Ungrammatical**	Matthews
SST-2	Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = **.93056 (Very Positive)**	Accuracy
MRPC	Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = **A Paraphrase**	Accuracy / F1
STS-B	How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = **4.6 (Very Similar)**	Pearson / Spearman
QQP	Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = **Not Similar**	Accuracy / F1
MNLI-mm	Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = **Contradiction**	Accuracy
QNLI	Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = **Answerable**	Accuracy
RTE	Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = **Entailed**	Accuracy
WNLI	Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = **Incorrect Referent**	Accuracy

Figure 14. GLUE task list [WSM+18]

These tasks all seek to test a model's understanding of a specific aspect of language. It includes:

- Named Entity Recognition: which words in a sentence are a proper name, organization name, or entity?
- Textual Entailment: given two sentences, does the first sentence entail or contradict the second sentence?
- Coreference Resolution: given a pronoun like "it" in a sentence that discusses multiple objects, which object does "it" refer to?

Final performance score of a model according to this benchmark is average score of those nine tasks. As most of the task metrics are accuracies, the higher the GLUE score - the better the model is able to "understand" the language.

3.6.4. Human evaluation

Human evaluation can be done through following approaches [DRO+20]:

1. Lab experiments, where users are invited to the lab to interact with a dialog system and fill in a questionnaire afterwards. This approach was popular before crowdsourcing became widely available.

2. In-field experiments, where feedback is collected from real users of a dialog system. This strategy allows user feedback to be gathered over a span of several months and was also used to judge the Alexa Prize [RPK+18].

3. Crowdsourcing, when the human evaluation is performed using crowdsourcing platforms such as Amazon Mechanical Turk (AMT) [Cro12]. This is the most popular strategy for human evaluation in current research.

3.6.4.1. Sensibleness and Specificity Average

One example of human evaluation metric is Sensibleness and Specificity Average (SSA) proposed by [ALS+20]. To measure the quality of a response given a context, authors propose a sequence of two questions: ”Does the response make sense?” and ”Was the response specific to context?”. For example, if A says, “I love tennis,” and B responds, “That’s nice,” then the utterance should be marked, ”sensible”, but “not specific”. That reply could be used in dozens of different contexts. However, if B responds, “Me too, I can’t get enough of Roger Federer!” then it is marked as “specific”, since it relates closely to what is being discussed. Given a set of responses labeled as answers for described questions, sensibleness and specificity are calculated as the percentage of responses labeled as sensible and specific, respectively. Final metric is created by simply averaging these two values. Authors demonstrated that perplexity is strongly negatively correlated with the SSA score, see Figure 15.

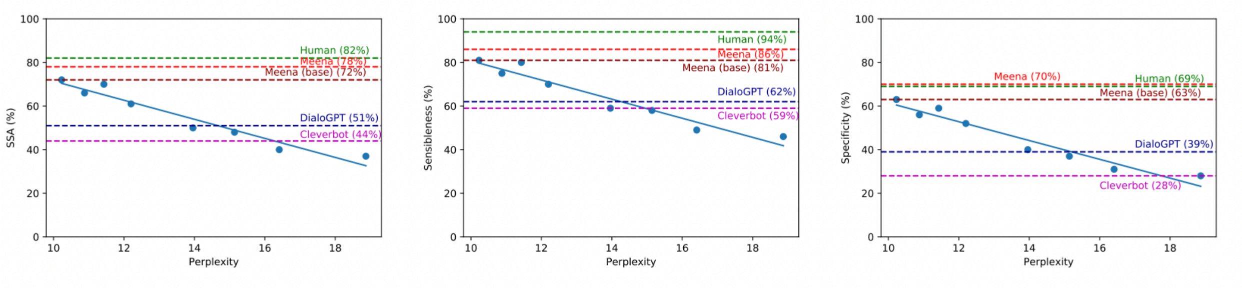

Figure 15. SSA correlation to Perplexity [ALS+20]

3.6.4.2. Acute-eval

Acute-eval authors [LWR19] proposed evaluating models in comparison to one another. They created a evaluation system that asks humans to directly compare side-by-side multi-turn dialogues conducted by two models. See Figure 16 for an example.

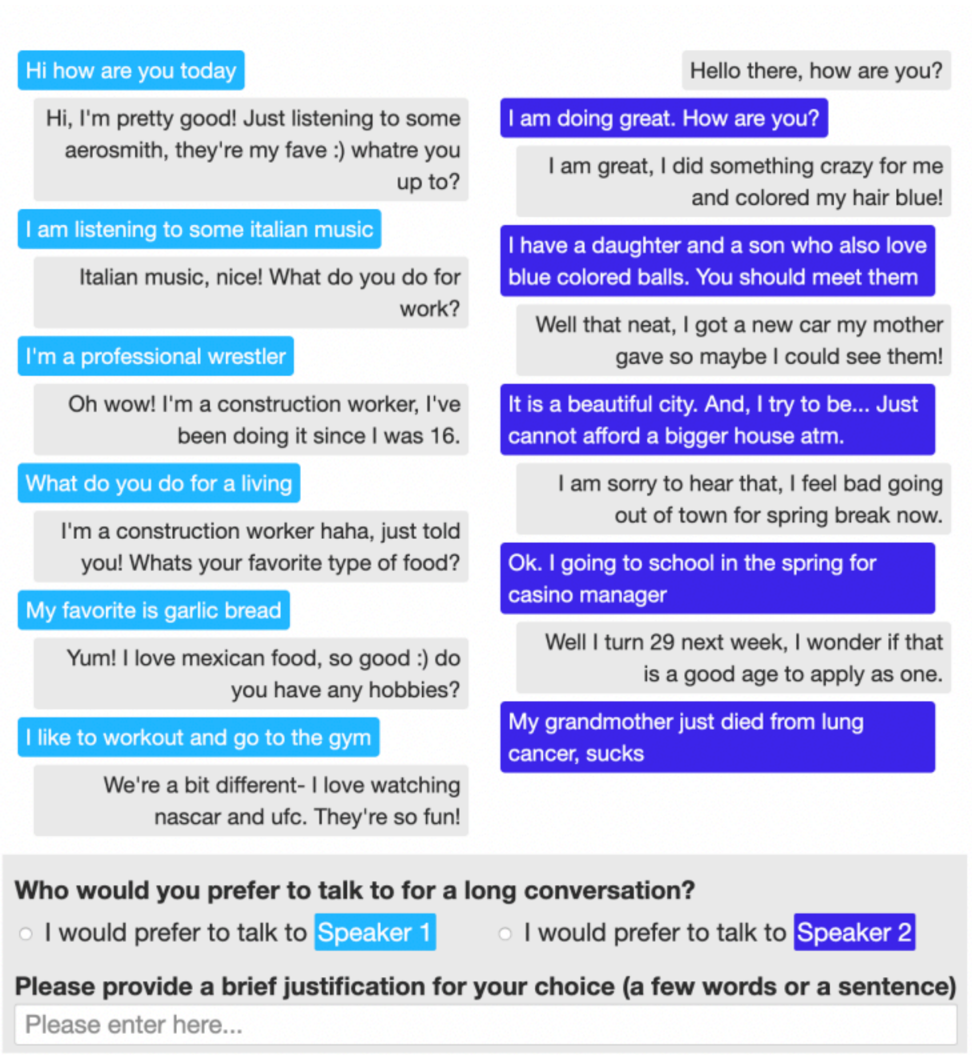

Figure 16. Acute-eval: system to compare two multiturn dialogues [LWR19]

Acute-eval method consists of two steps: (1) collect conversation logs for both models; (2) In a number of trials, ask annotators to make binary judgments between sampled pairs from the logs, and collate the results to determine the winner. Questions that are asked to the evaluator depends on a task, but they all focused on determining quality of following four attributes: engagingness, interestingness, knowledge and humanness.

3.6.4.3. PersonaChat evaluation

PersonaChat [ZDU+18] authors created their own questionnaire in order to evaluate their model. Their questionnaire consists of following questions:

- Fluency

  People are asked to score the chatbot as a score from 1 to 5, where 1 is “not fluent at all”, 5 is “extremely fluent”, and 3 is “OK”.

- Engagingness

People are asked to score the chatbot as a score from 1 to 5, where 1 is “not engaging at all”, 5 is “extremely engaging”, and 3 is “OK”.

- Consistency

  Following example is given: “I have a dog” followed by “I have no pets” is not consistent. The score is again from 1-5.

- Persona Detection

  During evaluation two possible profiles are displayed to a evaluator: One profile is chosen at random, and the other is the true persona given to the model. Evaluator is asked which is more likely to be the profile of the chatbot the person just spoke to.

4. Methodology

This section describes the baseline, knowledge retrieval algorithm, datasets and evaluation metrics of the experiments.

4.1. Baseline

During the initial stage of methodology selection ConceptFlow [ZLX+19] model has been chosen as a baseline model, as it showed promising results in the paper (see 3.4.1). Authors made their implementation publicly available. However, while using their repository, published results were not reproducible due to lack of details regarding data preprocessing and presence of multiple bugs in the code. Because of these reasons it was decided to use another model as a baseline - Generative BlenderBot [RDG+20] (see Section 3.5.2).

BlenderBot authors provide a well-developed open-source implementation with infrastructure suitable for easily making modifications and test new results. Besides, authors publicly provide a pre-trained weights for this model which makes it possible to fine-tune the model on new datasets. BlenderBot already is able to show personality and incorporate external knowledge, but it suffers from several problems such as repetition, forgetfulness, hallucinating knowledge, etc. Some of these problems will be addressed in this work in order to improve the performance of the model.

4.2. Knowledge Retrieval

As Generative Blenderbot is trained on Blender Skill Talk [SWS+20] dataset without any knowledge retrieval it can hallucinate knowledge from the training set. One of the possible solution to this problem is to retrieve knowledge from structured knowledge base as it does not contain any redundant information in opposition to unstructured knowledge (see Section 3.4.2).

More concretely, Generative Blenderbot can be trained on the same data but with relative knowledge appended to input sentences. In order to create such data it is necessary to create a algorithm that retrieves knowledge from some knowledge base. In particular, knowledge graph ConceptNet [SCH16] is used in this work.

ConceptNet is a knowledge graph that connects words and phrases of natural language (terms) with labeled edges (relations). Its knowledge is collected from many sources that include resources created by experts, crowd-sourcing, and games with a purpose. It is designed to represent the general knowledge involved in understanding language, improving natural language applications by allowing the application to better understand the meanings behind the words people use. ConceptNet contains over 21 million edges and over 8 million nodes. Its English vocabulary contains approximately 1,500,000 nodes, and there are 83 languages in which it contains at least 10,000 nodes. ConceptNet uses a closed class of 36 relations intended to represent a relationship independently of the language or the source of the terms it connects. Examples of relations: *SimilarTo, AtLocation, CapableOf, Causes, CreatedBy, DefinedAs, Desires, FormOf, HasA, IsA, UsedFor, etc*.

Before using ConceptNet for knowledge retrieval, it was filtered to save memory consumption and to make knowledge more informative for the model. All non-english vocabulary was removed. Besides, all syntactic relations (*Antonym, EtymologicallyRelatedTo, FormOf, etc.*) were also pruned. After this filtering approach only lemmas of terms were left in knowledge graph. It is necessary to take into account when querying knowledge base.

The high-level formulation of knowledge retrieval system is described in Alg. 1 and more detailed explanation of each step follows later in this section.

Algorithm 1 Retrieving assertions from message

```
Input: message, N
Output: assertions
for each sentence ∈ message do
    Encode sentence into vector
    for each token ∈ sentence do
        Find all assertions for token in ConceptNet
        Encode all assertions into vectors
        Find cosine similarities between assertions vectors and sentence vector
        Leave only top N similar assertions
    end for
end for
```

1. Given a dialogue message, it is splitted into sentences by sentence segmentation strategy. During dialogue it is not common to have complex sentence boundaries, that is why simple segmentation that splits sentences by punctuation (*.!?*) is used.

2. Each sentence is transformed into vector by sentence embedding model. Sentence embedding model is a neural network that converts sentences into vector representation in such a way that semantically similar sentences are close in vector space. It can be done by converting each individual word in the sentence by word embedding model like Word2Vec [MCC+13] or GloVe [PSM14] and then average those vectors into one representing whole sentence. However, Sentence-BERT [RG19] authors showed that their transformer model outperforms this averaging approach. This was possible due to the fact that Sentence-BERT is able to capture context from whole sentence rather then just averaging individual word vectors. This is why it was decided to use Sentence-BERT model for this step.

3. Each sentence is tokenized, i.e. splitted into words or group of words. Group of words are more correct in cases when these words form an contextually meaningful expression, e.g. *"best man", "flying colors", "The Great Wall"*. In order to properly retrieve tokens, tokenizer algorithm and dependency parser [CM14] are used. Dependency parser builds a dependency tree which describes relationships between words in a sentence. Each relationship has one head (word) and a dependent that modifies the head. Each relationship is also labeled according to the nature of the dependency between the head and the dependent (e.g. "Adjectival Modifier", "Compound"). These labels are described at Universal Dependency Relations [dMDS+14].

4. Each token is lemmatized and queried into the ConceptNet retrieving all assertions (<subject, relation, object>) connected to that token. These assertions compose main knowledge for each token, but there can be too much of them for inputting to the generative model. That is why it is necessary to filter them.

5. All assertions can be represented as small sentences (e.g. A net is used for catching fish), therefore they are vectorized using same sentence embedding model from step 2.

6. Cosine similarity score is calculated between sentence vector and all assertions vectors. All assertion vectors are ranked by similarity to sentence vector.

7. Only top N (where N is specified input parameter) similar assertions are added to the final assertions set.

Proposed algorithm can be used both during training and inference (real-time dialogue). Overview of the proposed pipeline during real-time dialogue is displayed on Figure 17.

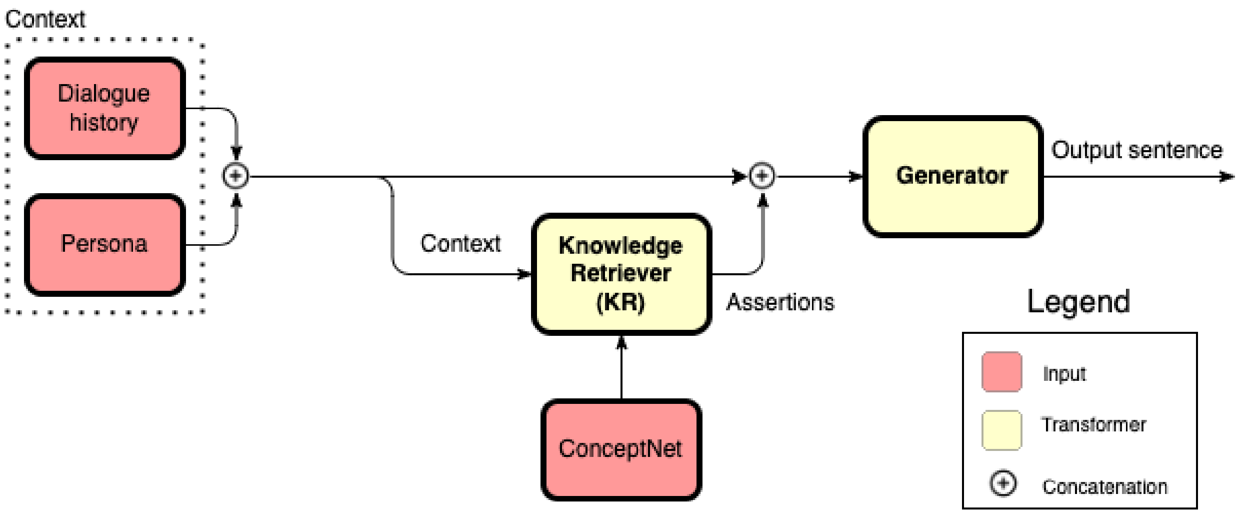

Figure 17. Proposed architecture of the dialogue system augmented with commonsense knowledge

First, latest message (or whole dialogue history, see Experiment 5.6) is concatenated with Persona text assigned to the bot (see Persona-Chat dataset description from Section 3.3.2) to a single context. Next a knowledge retrieval algorithm process context and returns assertions from ConceptNet as described in Algorithm 1. Finally, knowledge is concatenated with context and is inputted into generator BlenderBot which returns a dialogue utterance. During model training though, it is not necessary to run knowledge retrieval algorithm in real-time, so all BlenderBot datasets were preprocessed and new datasets were created to save computation time. Process of creating new datasets is described in the next section.

4.3. Dataset

Originally, Generative Blenderbot was trained on combination of following datasets: ConvAI2 [DLM+19], Empathetic Dialogues [RSL+18], Wizard Of Wikipedia [DRS+19] and Blender Skill

Talk (BST) [SWS+20]. See more details regarding each dataset in Section 3.5. Dataset used for training in this work is based on these datasets to keep their original qualities. But at the same time they are enhanced by adding knowledge assertions into them to guide the model in knowledge usage.

Each sample in any dialogue dataset is input message labeled with answer to that message. To incorporate knowledge in the dataset, each input message was appended with associated assertions which were extracted using knowledge retrieval system described in previous section. It is worth mentioning that labels are not appended with knowledge to simulate inference scenario when its necessary to generate just answer message without any assertions. Therefore, knowledge guided versions of these datasets were created. In later sections, newly created datasets are marked with label ”with assertions”.

4.4. Metrics

Following evaluation metrics are used to evaluate the quality of generated responses: Perplexity (PPL) [SSB+16], BLEU [PRW+02], ROUGE [Lin04] are used for measuring novelty, relevance and repetitiveness; Distinct-1, Distinct-2 [LGB+16a] are used for diversity.

A seq2seq model outputs a probability distribution over possible next response tokens. Perplexity measures how well the model predicts the test set data; in other words, how accurately it expects what people will say next. When interpreting perplexity scores, it should be noted that lower is better and that the theoretical minimum is one. As it was described in Section 3.6.4.1, it was shown that perplexity has high correlation to human judgement. Therefore, this metric will be used as main measure of the model's quality.

Distinct is an algorithm for evaluating the textual diversity of the generated text [LGB+16a]. Distinct-n is calculated as the number of distinct n-grams divided either by total number of words across all generations (inter) or by number of words only within one sentence (intra). The larger the number of distinct n-grams, the higher the diversity of the generated text. This is useful in dialogue evaluation context as it can help to prove or reject the hypothesis that retrieved knowledge can help the model to be more diverse.

5. Experiments

This section describes implementation details, results of conducted experiments, as well as planned future experiments.

5.1. Implementation details

All algorithms and models used in experiments were implemented in Python programming language. Spacy [HMV+20] along with nltk [BKL09] libraries were used for various text preprocessing operations: sentence segmentation, tokenization, dependency parsing, lemmatization. NetworkX [HSS08] was used for performing operations with ConceptNet knowledge graph. Official implementation of SentenceBert [RG19] was used for sentence vectorization. Blenderbot implementation as well as all used datasets are provided by ParlAI framework [MFF+17]. All deep learning models are implemented with help of PyTorch [PGM+19].

Initial experiments were conducted using GeForce RTX 3070 GPU with 8 GB of video memory. It was possible to train only the smallest (90M parameters) version of Generative Blenderbot model using that GPU. For larger model (2.7B) HPC cluster managed by IT Research Center of Vilnius University was used. It provided 2 Tesla V100 GPUs with 32GB of video RAM.

All shown results are using Generative Blenderbot model which is pre-trained on examples from Reddit obtained from PushShift3 [BZK+20]. This dataset covers a vast range of topics, and hence it is a good candidate for helping train a dialogue model in the open-domain case. Pre-trained weights are also provided by ParlAI framework.

All models were trained in multi-task fashion, meaning that during training there is an equal probability to have sample from all of the used datasets (WoW, BST, ConvAI2) in a batch. All metrics shown in this section are retrieved by evaluating models on original validation splits of described datasets.

5.2. Reproducing original results

In order to make valid evaluation of algorithm developed in this work, it is necessary to reproduce results of the BlenderBot paper to compare the outcome.

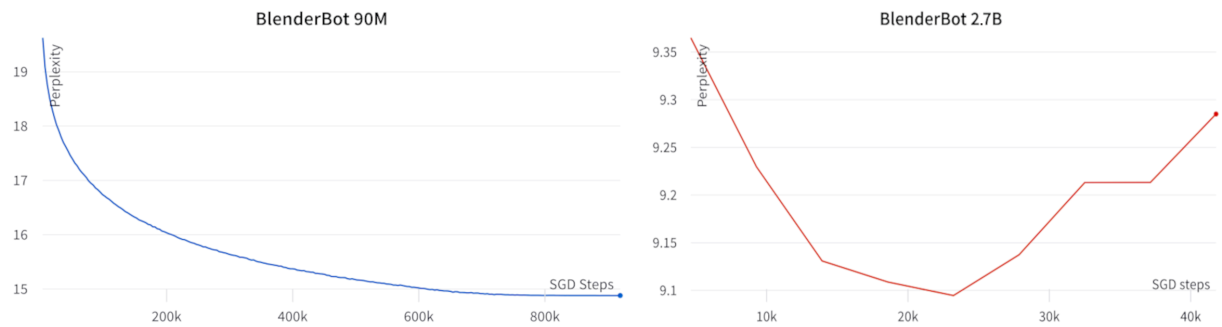

Figure 18. Perplexity of 90M (Left) and 2.7B (Right) versions of Blenderbot during finetuning on original dataset

During this experiment, two sizes (90M and 2.7B) of Generative Blenderbot were fine-tuned on original dataset described in Section 3.5.2. Values of hyperparameters of these models were taken from the original paper and from ParlAI documentation. Perplexity value of 90M version of the model declared in original paper equals to *14.65*. After 800k training steps (63 epochs, ≈48 hours) reproduced perplexity value was equal to *14.88*, which is very close to original results.

During finetuning of the 27B version of the model, it reached the best perplexity value of *9.09* in far fewer steps - 23k (1 epoch, ≈13 hours). It started to overfit during further training which indicates that model of such size is able to learn features of the input data during single pass through whole dataset. Original value of the perplexity for this model is *8.98* which is also similar to reproduced value. Difference of the fine-tuning process of two sizes of Blenderbot can be seen on Figure 18.

5.3. Fine-tuning on data with knowledge

During this experiment BlenderBot was fine-tuned on newly created datasets with assertions that were described in Section 4.3.

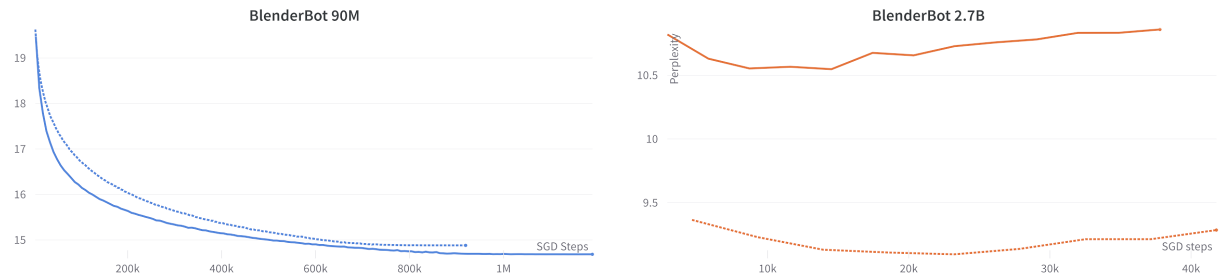

Figure 19. Perplexity of 90M (Left) and 2.7B (Right) versions of Blenderbot during finetuning on original dataset (dashed) and dataset with knowledge (solid)

As well as in previous experiment, two sizes of the model were fine-tuned. As it can be seen from Figure 19, 90M model trained on data with knowledge outperformed original results (14.69 vs 14.88), while Blenderbot 2.7B showed significantly worse results (10.55 vs 9.09). These results indicate that larger model tends to overfit more to pre-training data without assertions (Reddit dataset) and it is hard for it to capture relationships between input data and extracted knowledge.

Results provided in Table 1 show difference of BlenderBot sizes trained both on original datasets and datasets with assertions. Each metric value shown in the table is an average of value measured on all used datasets (e.g. perplexity is first measured on validation split of BST, ConvAI2, WoW, then averaged and shown in the table).

Table 1. Comparison of different sizes of BlenderBot

		Novelty (↓)		**Diversity (↑)**	
Model	**PPL**	**BLEU-1**	**ROUGE-1**	**InterDISTINCT-1**	**IntraDISTINCT-1**
BlenderBot 90M	14.88	0.1365	0.1801	**0.0434**	0.8357
BlenderBot 90M (with assertions)	14.69	0.1331	0.1747	0.0413	0.8332
BlenderBot 2.7B	**9.09**	0.1474	0.1974	0.0295	0.8952
BlenderBot 2.7B (with assertions)	10.55	**0.1279**	**0.1719**	0.0264	**0.9108**

Table 1 shows that, although Blenderbot 2.7B fine-tuned on data with assertions has poor perplexity value, it outperforms original model in BLEU, ROUGE and IntraDISTINCT. Despite this fact, it is still necessary to focus on improving perplexity value as it correlates with human evaluation of dialogue system. Therefore, next experiments will aim to improve this particular metric.

5.4. Datasets performance comparison

It is possible to analyze performance of the model on each particular dataset rather than just seeing averaged value. Tables 2 and 3 shows quality of the 90M and 2.7B measured on individual datasets.

Table 2. Novelty (lower better) and Diversity (higher better) of Generative Blenderbot 90M trained on different datasets

		Novelty (↓)		**Diversity (↑)**	
Dataset	**PPL**	**BLEU-1**	**ROUGE-1**	**InterDISTINCT-1**	**IntraDISTINCT-1**
BST	**16.1**	**0.1187**	**0.1654**	0.0432	**0.8263**
BST (With assertions)	16.18	0.1201	0.1655	0.0432	0.8209
ConvAI2	**12.66**	**0.1460**	0.1819	0.0261	**0.8486**
ConvAI2 (With assertions)	13.34	0.1465	**0.1818**	**0.0266**	0.8363
Wizard of Wikipedia	18.72	0.1450	0.1931	**0.0610**	0.8322
Wizard of Wikipedia (With assertions)	**17.2**3	**0.1327**	**0.1768**	0.0543	**0.8424**

90M version of the model was not trained on Empathetic Dialogues dataset due to limitation of the model size. It can be seen that the only dataset on which Blenderbot 90M outperforms the original is Wizard of Wikipedia. Extracted knowledge helps the model to incorporate a lot of facts that are contained in this dataset.

Table 3. Novelty (lower better) and Diversity (higher better) of Generative Blenderbot 2.7B trained on different datasets

		Novelty (↓)		**Diversity (↑)**	
Model	**PPL**	**BLEU-1**	**ROUGE-1**	**InterDISTINCT-1**	**IntraDISTINCT-1**
BST	**10.21**	.1350	.1897	.05561	.8621
BST (With assertions)	11.55	**.1183**	**0.1640**	**.0573**	**.8991**
ConvAI2	**8.838**	.1560	.1927	**.03793**	.8937
ConvAI2 (With assertions)	10.45	**.1418**	**.1740**	.03616	**.8958**
Wizard of Wikipedia	**8.889**	.1551	.2077	**.08393**	.8825
Wizard of Wikipedia (With assertions)	10.8	**.1134**	**.1569**	.07017	**.8960**
Empathetic Dialogues	**8.446**	.1435	.1996	**.03819**	.9426
Empathetic Dialogues (With assertions)	9.387	**.1380**	**.1929**	.03452	**.9525**

5.5. ConceptNet filtering

In order to improve quality of knowledge appended to datasets, the knowledge base was filtered. ConceptNet contains a lot of infrequent relations which are hard to learn and often overspecific, and hence not useful for establishing high quality relations and paths between concepts. Therefore, a subset of the knowledge base that contains all assertions of the 13 most frequent relations is extracted: *RelatedTo, HasContext, IsA, FormOf, UsedFor, SimilarTo, AtLocation, HasSubevent, HasPrerequisite, CapableOf, Causes, MannerOf, PartOf.*

Now, when knowledge base contains small amount of possible relations, it is possible to convert each relation into a fixed special token (e.g. __*RelatedTo*__) instead of treating it as a pure text. During this experiment original datasets where augmented in a similar way as described in Section 4.2, but filtered version of ConceptNet is used. BlenderBot was fine-tuned on a new dataset containing these special tokens. Performance of BlenderBot fine-tuned on original dataset, on unfiltered datasets with assertions and on datasets with special tokens are compared in the Table 4.

Table 4. Comparison of different versions of BlenderBot 2.7B

		Novelty (↓)		**Diversity (↑)**	
Model	**PPL**	**BLEU-1**	**ROUGE-1**	**InterDISTINCT-1**	**IntraDISTINCT-1**
BlenderBot 2.7B (original)	9.09	0.1474	0.1974	**0.0295**	0.8952
BlenderBot 2.7B (with assertions)	10.55	0.1279	0.1719	0.264	**0.9108**
BlenderBot 2.7B (with special tokens)	10.31	**0.1276**	**0.1717**	0.0265	0.9007
BlenderBot 2.7B (assertions from context)	**9.081**	0.1470	0.1958	0.0292	0.9020

5.6. Extracting knowledge from dialogue history

During previous experiments, each dataset sample had assertions that were extracted only from previous input message. The goal of this experiment is to train the model on dataset, each sample of which will contain assertions extracted from whole dialogue history (all previous messages).

This new approach requires a tweak in the algorithm described in Section 4.2. Instead of inputting single message, a set of all messages seen during current dialogue is used for knowledge extraction. In this case, set of messages is treated as single text and converted to a vector representation at step 2. Later this vector representation is compared to assertions vectors just as in original algorithm.

This method helps to capture knowledge from whole dialogue and incorporate it into the model during each new response generation. It was implemented by applying developed algorithm directly during training, in real-time. It makes training slower, but not significantly. Results of this experiment can be seen in Table 4 and are marked as ”assertions from context”.

It can be seen that, this approach significantly improves model perplexity, but at cost of novelty metrics.

5.7. Dialogue Generation

Performance of the proposed method was measured with automated metrics, but also it was possible to see dialogues generated by the model via self-chat technique: during self-chat, two independent objects of the same model were created and tasked to generate responses to one another. Examples of dialogues generated by model trained on original data and by model trained on data with assertions are shown in Figures 20 and 21 respectively. Both dialogues have the same initial sentence

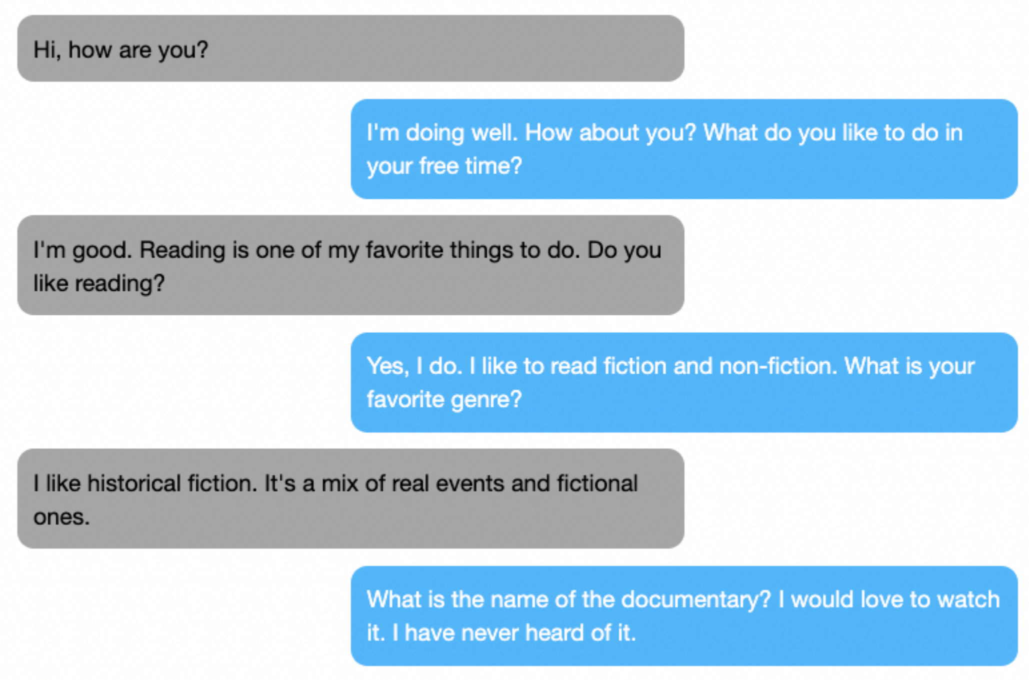

Figure 20. Example self-chat of BlenderBot 2.7 fine-tuned on original data

(*"Hi, how are you?"*) and persona context (initial information about the bot: *"I am a senior citizen, I like to read"*) in order to be able to directly compare generated utterances.

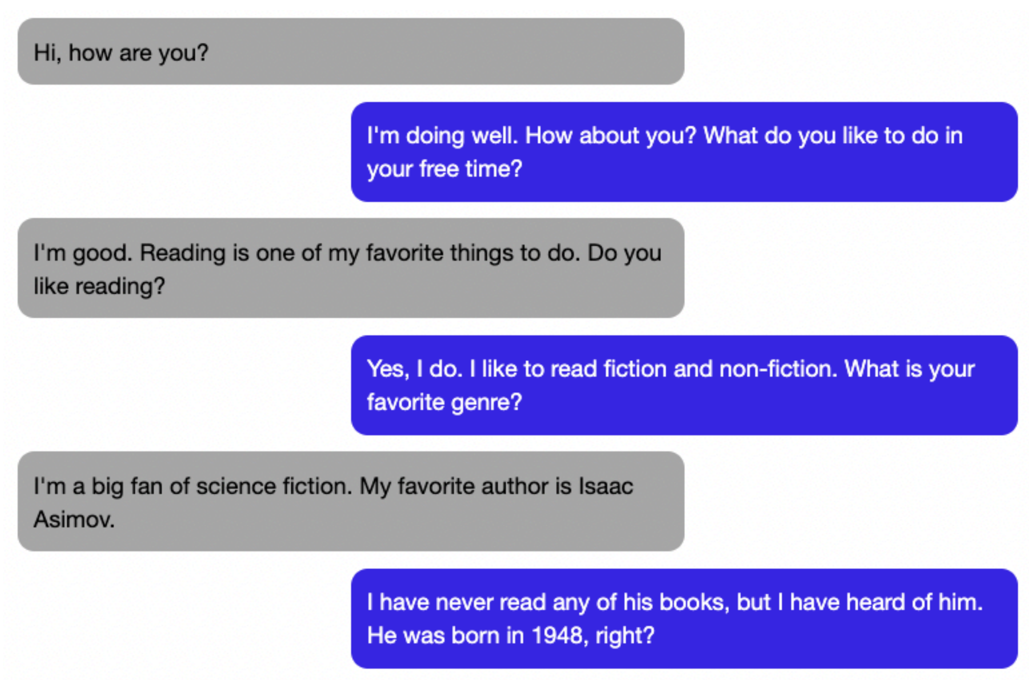

Figure 21. Example self-chat of BlenderBot 2.7 fine-tuned on data with assertions

These examples are cherry-picked so they cannot be a proper indication of the results, but as it can be seen from the comparison of these dialogues, second (Figure 21) dialogue seems more meaningful. During first dialogue, BlenderBot asks *"What is the name of the documentary?"* after the sentence *"I like historical fiction"* which seems like not an appropriate reaction.

5.8. Human evaluation

Model with highest metrics (BlenderBot 2.7B trained on data with assertions extracted from context) was also evaluated by humans. A small survey was created following methodology of Acute-eval (see Section 3.6.4.2).

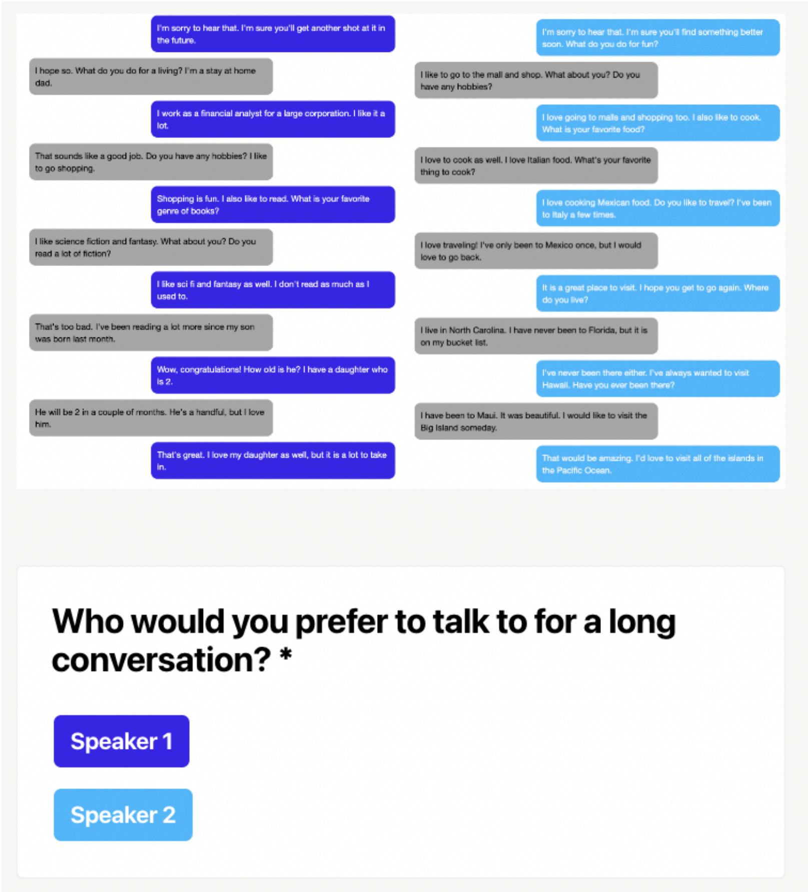

Figure 22. A screenshot from one the pages of the survey

Survey consisted of five different pairs of dialogues (one dialogue was generated by proposed model, the other one by baseline original model). For each pair of dialogues, people were asked to choose one dialogue by answering specific questions. There were 3 different questions (categories) for each pair. Each question represents a particular speaker attribute:

- engagingness - how much the speaker is involved in the conversation. Represented by question: *"Who would you prefer to talk to for a long conversation?"*
- humanness - how likely that the speaker creates responses which would a human create. Represented by question: *"Which speaker sounds more human?"*
- knowledge - how much the speaker reveals his general knowledge about the world. Represented by question: *"If you had to say that one speaker is more knowledgeable and one is more ignorant, who is more knowledgeable?"*

All aforementioned questions were formulated and optimised by authors of the Acute-eval [LWR19].

Survey was conducted via online survey platform CrowdSignal[1]. It allows to create custom surveys as well as it provides basic statistics of the answers. Link to the survey was shared by author of this paper via social media. Example screenshot from the published survey is shown on Figure 22.

Each metric was calculated as the percentage of those who have chosen a particular speaker in that category. Results of the evaluation are shown in Table 5. There were total 20 questioned people making it to 100 answers in each category (1 question for each of the 5 pairs).

Table 5. Human evaluation results

Model	**Engagingness (%)**	**Humanness (%)**	**Knowledge (%)**
BlenderBot 2.7B (original)	45.88	43.53	43.53
BlenderBot 2.7B (knowledge-augmented)	**54.12**	**56.47**	**56.47**

Although results show that proposed knowledge-augmented model is better than the original baseline, acquired results were not statistically tested, therefore cannot be considered statistically significant.

[1] `https://ilyalas6394.crowdsignal.net/chatbot-evaluation-1`

6. Conclusion & Future work

Overall there are a lot of topics that are still open in dialogue system domain. Although there are plenty of work done for combining different dialogue tasks, there is little research featuring combined persona and structured graphs, to my knowledge. Thus, the main focus of this work is to build a system that is able to use external knowledge in form of knowledge graphs while having another conversational qualities.

As a baseline model for this research, a current state-of-the-art model BlenderBot [RDG+20] has been chosen. Authors provide an open-source implementation which makes it easier to make modifications and test new results. Besides, authors publicly provide a pre-trained weights for this model that later can be used for finetuning on a custom dataset.

A novel knowledge extraction algorithm was developed in order to augment existing state-of-the-art model BlenderBot. ConceptNet - large knowledge base designed to represent the general knowledge about the world, is used as a knowledge source for the developed algorithm. Knowledge extraction algorithm was used both during preprocessing of the datasets before model training and during real-time dialogue generation. New, knowledge augmented versions of the ConvAI2, Wizard of Wikipedia, Empathetic Dialogues, Blended Skill Talk datasets were created. These new datasets were used for model fine-tuning. During a comparison of original model and proposed model, results showed that model trained on knowledge augmented dataset tends to generate more novel (0.128 in BLEU, 0.172 in ROUGE-1) responses. Besides, proposed model outperformed original model in perplexity - metric which was studied to have high correlation with human evaluation of dialogue systems. A survey was conducted in order to evaluate developed dialogue system. Although survey results cannot be considered statistically significant, human evaluation showed that knowledge-augmented BlenderBot performs better that original model. In general, knowledge augmentation of the dataset helped BlenderBot generate more meaningful responses which can be seen from automated metrics and cherry-picked results.

Current state-of-the-art dialogue systems still suffer from several unsolved problems such as hallucinating knowledge, repetition, forgetfulness. Knowledge hallucination is the problem that could be solved in the suggested method, but unfortunately, proposed model is also vulnerable to this drawbacks. The fact that adding explicit knowledge to the training dataset didn't solve knowledge hallucination, allows us to conclude that cause of these issues lies somewhere else.

In order to determine the reason of these problems and solve them, following directions for future research are proposed: changing knowledge extraction algorithm (e.g. using knowledge paths prediction [BKP+21]), filtering knowledge facts from pre-training data, improving model architecture (e.g. adding factual classifier [GAC+21]).

References

[Aga18] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[ALS+20] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, et al. Towards a human-like open-domain chatbot, 2020. arXiv: `2001.09977 [cs.CL]`.

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., 2009.

[BKP+21] Maria Becker, Katharina Korfhage, Debjit Paul, and Anette Frank. CO-NNECT: A framework for revealing commonsense knowledge paths as explicitations of implicit knowledge in texts. *CoRR*, abs/2105.03157, 2021. arXiv: `2105.03157`. URL: `https://arxiv.org/abs/2105.03157`.

[BL05] Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan. Association for Computational Linguistics, 2005-06. URL: `https://www.aclweb.org/anthology/W05-0909`.

[BMR+20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. Language models are few-shot learners, 2020. arXiv: `2005.14165 [cs.CL]`.

[BZK+20] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. The pushshift reddit dataset, 2020. arXiv: `2001.08435 [cs.SI]`.

[CGC+14] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. arXiv: `1412.3555`. URL: `http://arxiv.org/abs/1412.3555`.

[CH15] Erik Cambria and Amir Hussain. *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. 2015-11.

[CLX+20] Erik Cambria, Yang Li, Frank Z. Xing, Soujanya Poria, and Kenneth Kwok. *Senticnet 6: ensemble application of symbolic and subsymbolic ai for sentiment analysis*. In Association for Computing Machinery, New York, NY, USA, 2020, pp. 105–114. ISBN: 9781450368599. URL: `https://doi.org/10.1145/3340531.3412003`.

[CM14] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 740–750, Doha, Qatar. Association for Computational Linguistics, 2014-10. DOI: `10.3115/v1/D14-1082`. URL: `https://aclanthology.org/D14-1082`.

[Cro12] Kevin Crowston. Amazon mechanical turk: a research tool for organizations and information systems scholars. In Anol Bhattacherjee and Brian Fitzgerald, editors, *Shaping the Future of ICT Research. Methods and Approaches*, pp. 210–221, Berlin, Heidelberg. Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-35142-6.

[CvMG+14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. arXiv: 1406.1078. URL: http://arxiv.org/abs/1406.1078.

[DCL+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[DLM+19] Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander H. Miller, et al. The second conversational intelligence challenge (convai2). *CoRR*, abs/1902.00098, 2019. arXiv: 1902.00098. URL: http://arxiv.org/abs/1902.00098.

[dMDS+14] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford dependencies: a cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 4585–4592, Reykjavik, Iceland. European Language Resources Association (ELRA), 2014-05. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1062_Paper.pdf.

[DRO+20] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54(1):755–810, 2020-06. ISSN: 1573-7462. DOI: 10.1007/s10462-020-09866-x. URL: http://dx.doi.org/10.1007/s10462-020-09866-x.

[DRS+19] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: knowledge-powered conversational agents, 2019. arXiv: 1811.01241 [cs.CL].

[Dup19] Samuel Dupond. *A thorough review on the current advance of neural network structures*. In Annual Reviews in Control. 14, 2019, pp. 200–230. URL: https://www.sciencedirect.com/journal/annual-reviews-in-control.

[GAC+21] Greyson Gerhard-Young, Raviteja Anantha, Srinivas Chappidi, and Björn Hoffmeister. Low-resource adaptation of open-domain generative chatbots, 2021. DOI: 10.48550/ARXIV.2108.06329. URL: https://arxiv.org/abs/2108.06329.

[GBC+17] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. *CoRR*, abs/1702.01932, 2017. arXiv: 1702.01932. URL: http://arxiv.org/abs/1702.01932.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[GG19] S. Ghosh and D. Gunning. *Natural Language Processing Fundamentals: Build intelligent applications that can interpret the human language to deliver impactful results*. Packt Publishing, 2019. ISBN: 9781789955989. URL: https://books.google.by/books?id=i8-PDwAAQBAJ.

[HMV+20] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. DOI: 10.5281/zenodo.1212303.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 1997-12. DOI: 10.1162/neco.1997.9.8.1735.

[HSL+20] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring, 2020. arXiv: 1905.01969 [cs.CL].

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pp. 11–15, Pasadena, CA USA, 2008.

[KWC17] Satwik Kottur, Xiaoyu Wang, and Vitor Carvalho. Exploring personalized neural conversational models. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3728–3734, 2017. DOI: 10.24963/ijcai.2017/521. URL: https://doi.org/10.24963/ijcai.2017/521.

[LGB+16a] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models, 2016. arXiv: 1510.03055 [cs.CL].

[LGB+16b] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model, 2016. arXiv: 1603.06155 [cs.CL].

[Lin04] Chin-Yew Lin. ROUGE: a package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain. Association for Computational Linguistics, 2004-07. URL: https://www.aclweb.org/anthology/W04-1013.

[LLS+16] Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How NOT to evaluate your dialogue system: an empirical study of unsupervised evaluation metrics for dialogue response generation. *CoRR*, abs/1603.08023, 2016. arXiv: 1603.08023. URL: http://arxiv.org/abs/1603.08023.

[LWR19] Margaret Li, Jason Weston, and Stephen Roller. Acute-eval: improved dialogue evaluation with optimized questions and multi-turn comparisons, 2019. DOI: 10.48550/ARXIV.1909.03087. URL: https://arxiv.org/abs/1909.03087.

[LXW+19] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. Learning to select knowledge for response generation in dialog systems. *CoRR*, abs/1902.04911, 2019. arXiv: 1902.04911. URL: http://arxiv.org/abs/1902.04911.

[Mat18] Michihito Matsuda. Politics 2028: why artificial intelligence will replace politicians. https://www.slideshare.net/MichihitoMatsuda/politics-2028-why-artificial-intelligence-will-replace-politicians-by-michihito-matsuda, 2018. accessed 2021-01-12.

[MCC+13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. arXiv: 1301.3781 [cs.CL].

[MFD+16] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents, 2016. arXiv: 1606.03126 [cs.CL].

[MFF+17] A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. Parlai: a dialog research software platform. *arXiv preprint arXiv:1705.06476*, 2017.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, et al. Pytorch: an imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[PRW+02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. Bleu: a method for automatic evaluation of machine translation:8, 2002-10. DOI: 10.3115/1073083.1073135.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL: http://www.aclweb.org/anthology/D14-1162.

[QHZ+18] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Assigning personality/profile to a chatting machine for coherent conversation generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 4279–4285. International Joint Conferences on Artificial Intelligence Organization, 2018-07. DOI: 10.24963/ijcai.2018/595. URL: https://doi.org/10.24963/ijcai.2018/595.

[RDG+20] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

[RG19] Nils Reimers and Iryna Gurevych. Sentence-bert: sentence embeddings using siamese bert-networks, 2019. arXiv: 1908.10084 [cs.CL].

[RPK+18] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, et al. Conversational ai: the science behind the alexa prize, 2018. arXiv: 1801.03604 [cs.AI].

[RSL+18] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. I know the feeling: learning to converse with empathy. *CoRR*, abs/1811.00207, 2018. arXiv: 1811.00207. URL: http://arxiv.org/abs/1811.00207.

[RSR+19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. DOI: 10.48550/ARXIV.1910.10683. URL: https://arxiv.org/abs/1910.10683.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020. arXiv: 1910.10683 [cs.LG].

[RWC+19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[SCH16] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: an open multilingual graph of general knowledge. *CoRR*, abs/1612.03975, 2016. arXiv: 1612.03975. URL: http://arxiv.org/abs/1612.03975.

[SSB+16] Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models, 2016. arXiv: 1507.04808 [cs.CL].

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. arXiv: 1409.3215. URL: http://arxiv.org/abs/1409.3215.

[SWS+20] Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. Can you put it all together: evaluating conversational agents' ability to blend skills. *CoRR*, abs/2004.08449, 2020. arXiv: 2004.08449. URL: https://arxiv.org/abs/2004.08449.

[VL15] Oriol Vinyals and Quoc Le. A neural conversational model, 2015. arXiv: 1506.05869 [cs.CL].

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. arXiv: 1706.03762 [cs.CL].

[VWH+19] Aditya Nrusimha Vaidyam, Hannah Wisniewski, John David Halamka, Matcheri S Kashavan, and John Blake Torous. Chatbots and conversational agents in mental health: a review of the psychiatric landscape. en. *Can. J. Psychiatry*, 64(7):456–464, 2019-07.

[WCB15] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks, 2015. arXiv: 1410.3916 [cs.AI].

[WDM18] Jason Weston, Emily Dinan, and Alexander Miller. Retrieve and refine: improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pp. 87–92, Brussels, Belgium. Association for Computational Linguistics, 2018-10. DOI: 10.18653/v1/W18-5713. URL: https://www.aclweb.org/anthology/W18-5713.

[WGW+20] Hao Wang, Bin Guo, Wei Wu, and Zhiwen Yu. Towards information-rich, logical text generation with knowledge-enhanced neural models, 2020. arXiv: 2003.00814 [cs.AI].

[WSM+18] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461, 2018. arXiv: 1804.07461. URL: http://arxiv.org/abs/1804.07461.

[YCC+18] Tom Young, Erik Cambria, Iti Chaturvedi, Minlie Huang, Hao Zhou, and Subham Biswas. Augmenting end-to-end dialog systems with commonsense knowledge, 2018. arXiv: 1709.05453 [cs.AI].

[YYW+16] Zhenlong Yu, Caixia Yuan, Xiaojie Wang, and Guohua Yang. Lstm encoder – decoder for dialogue response generation. In 2016.

[ZCH+20] Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. Personalized dialogue generation with diversified traits, 2020. arXiv: 1901.09672 [cs.CL].

[ZDU+18] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: i have a dog, do you have pets too? 2018. arXiv: 1801.07243 [cs.AI].

[ZLX+19] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. Conversation generation with concept flow. *CoRR*, abs/1911.02707, 2019. arXiv: 1911.02707. URL: http://arxiv.org/abs/1911.02707.

[ZNG+20] Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. Detecting hallucinated content in conditional neural sequence generation, 2020. DOI: 10.48550/ARXIV.2011.02593. URL: https://arxiv.org/abs/2011.02593.

[ZYH+18] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 4623–4629. International Joint Conferences on Artificial Intelligence Organization, 2018-07. DOI: 10.24963/ijcai.2018/643. URL: https://doi.org/10.24963/ijcai.2018/643.

[ZZW+19] Wei-Nan Zhang, Qingfu Zhu, Yifa Wang, Yanyan Zhao, and Ting Liu. Neural personalized response generation as domain adaptation. *World Wide Web*, 22(4):1427–1446, 2019-07. ISSN: 1386-145X. DOI: 10.1007/s11280-018-0598-6. URL: https://doi.org/10.1007/s11280-018-0598-6.