Dialogue System Augmented with Commonsense Knowledge

Ilya Lasy

Faculty or Mathematics and Informatics, Vilnius University
Didlaukio st. 47, LT-08303 Vilnius
ilya.lasy@mif.stud.vu.lt

Virginijus Marcinkevičius

Institute of Data Science and Digital Technologies
Vilnius University, Akademijos st. 4, LT-08412 Vilnius
virginijus.marcinkevicius@mif.vu.lt

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 proposed which is then used to incorporate knowledge into existing dialogue datasets. Current state-of-the-art model BlenderBot is finetuned on new datasets which shows improvement in novelty of utterances generated by the model.

Keywords: Natural Language Generation, Dialogue System, Chat-bot, Knowledge Graph, Deep Learning

1 Introduction

Dialogue systems, or chatbots, are highly popular nowadays in various fields such as commerce, education, entertainment, finance, health, etc. Most chatbots are accessed on-line via website popups or through virtual assistants and can quickly provide answers for frequently asked questions and help navigate on a hosted web-site.

DOI: https://doi.org/10.15388/LMITT.2022.7

This kind of chatbots are domain-specific and are not able to maintain a conversation in open domain which is a natural requirement for such cases as virtual assistants (e.g. Amazon Alexa), AI politicians [11], mental health chatbots [21] or any kind of bot that involves continues conversation on general topics.

Current state-of-the-art open-domain chatbots are able to demonstrate personality [26], knowledge [5], empathy [15] and combine all these abilities together [17]. This work focuses on incorporating knowledge into a dialogue system that already has developed aforementioned qualities. In particular, structured knowledge base ConceptNet [20] is used for guiding dialogue generation. ConceptNet was already used in dialogue systems during previous research [24, 25, 27], but most of these researches are focused just on using knowledge in isolation rather than combining knowledge with other features.

This work introduces knowledge extraction algorithm which is used for adding knowledge to training dataset and to interlocutor messages during inference. BlenderBot [17] is used as baseline architecture for the model and is trained on knowledge augmented data. Such knowledge-enhanced version outperforms vanilla version of the model in terms of BLEU [13], ROUGE [10], diversity [9].

2 Methodology

2.1 Baseline

BlenderBot [17] model has been chosen as a baseline model for chatbot. Authors have two different versions of the model: generative and retrieve-and-refine [23]. Generative version showed better results, so they released this version. Therefore, BlenderBot will be referred to as its generative variant further in this paper.

BlenderBot is a seq2seq transformer [22] which main power is in the data it is fine-tuned on. Originally, it was pre-trained on 1.5B training examples from Reddit [1] and fine-tuned on the combination of following **datasets**: ConvAI2 [6], Empathetic Dialogues (ED) [15], Wizard Of Wikipedia (WoW) [5] and Blender Skill Talk (BST) [19]. In Section 3, these datasets will be referred to as "original".

2.2 Knowledge extraction and datasets augmentation

As Blenderbot is finetuned on these datasets without any explicit knowledge guidance, it can hallucinate knowledge that is implicitly saved in the model weights. To address this problem, this work proposes explicit knowledge guidance with knowledge extracted from ConceptNet [20]. ConceptNet is a knowledge graph that connects words and phrases of natural language (*terms*) with labeled edges (*relations*). Each <*term, relation, term*> triple is referred to as an *assertion*.

Each sample in aforementioned datasets is input message labeled with answer to that message. To incorporate knowledge in the dataset, each input message is appended with associated assertions which were extracted using custom knowledge extraction algorithm:

Algorithm 1 Retrieving assertions from message

Input: message, k
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

Steps of the developed algorithm are described below:

1. Given a dialogue message, it is splitted into sentences by sentence segmentation strategy. During dialogue it's not common to have complex sentence boundaries, that is why simple segmentation strategy 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 was decided to use Sentence-BERT model [16] for this step as it outperforms analogues models in this task.

3. Ech 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 [4] are used. Dependency parser builds a dependency tree which describes relationships between words in a sentence.
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 *k* (where *k* is specified input parameter) similar assertions are added to the final assertions set.

Therefore, knowledge guided versions of four original datasets described in Section 2.1 were created using Algorithm 1. In Section 3, newly created datasets are marked as "with assertions". Knowledge extraction is also used during real time conversation with new model. Assertions are extracted from user's message and appended to input of the model, conditioning output not only on the user's sentence, but on explicit knowledge. Current state-of-the-art model BlenderBot was finetuned on these new datasets to check hypothesis that general knowledge could improve performance of the dialog system.

2.3 Evaluation Metrics

Following evaluation metrics are used to evaluate the quality of generated responses: Perplexity (PPL) [18], BLEU [13], ROUGE [10] are used for measuring novelty, relevance and repetitiveness; Distinct-1, Distinct-2 [9] are used for diversity.

Perplexity explicitly measures the model's ability to account for the syntactic structure of the dialogue (e.g. turn-taking) and the syntactic

structure of each utterance (e.g. punctuation marks). In dialogue, the distribution over the words in the next utterance is highly multi-modal, e.g. there are many possible answers, which makes perplexity particularly appropriate because it will always measure the probability of regenerating the exact reference utterance.

Distinct is an algorithm for evaluating the textual diversity of the generated text [9]. 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.

3 Experiments

BlenderBot model was released in 3 different sizes of parameters: 90M, 2.7B, 9.4B. Blenderbot implementa- tion as well as all used datasets are provided by ParlAI framework [12]. Initial experiments are performed on the smallest model, but starting from Section 3.3, 2.7B version of the model will be used.

All algorithms and models used in experiments were implemented in Python programming language. Spacy [8] along with nltk [3] libraries were used for various text preprocessing operations: sentence segmen- tation, tokenization, dependency parsing, lemmatization. NetworkX [7] was used for performing operations with ConceptNet knowledge graph. All deep learning models are implemented with help of PyTorch [14].

90M model was fine-tuned using GeForce RTX 3070 GPU with 8 GB of video memory. 2.7B version of the model was trained using cluster with 2 V100 with total of 64 GB video memory.

3.1 Datasets performance comparison

Two fine-tuning sessions of the BlenderBot were ran during this initial experiment. First, it was necessary to reproduce results of the baseline paper[17] and finetune BlenderBot on original dataset described in Section 2.1. Second, BlenderBot was fine-tuned on newly created datasets with assertions specified in Section 2.2. Model was 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. During results estimation original validation split was used for each dataset. Results of the finetuning are provided in Table 1.

Table 1. 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**

Results show that there is no significant improvement on BST and ConvAI2 datasets, but perplexity and novelty are better when measured on Wizard of Wikipedia dataset. Reason for this can be due to the fact that WoW dataset contains a lot of factoids and having explicit knowledge in the input can help to model dependencies between utterance and general knowledge.

3.2 Increasing model size

Because of previous results, it was hypothesized that the size of the model (90M) does not allow to capture complex relationships between input data and extracted knowledge. It was decided to repeat previous experiment with 2.7B parameter model. Results provided in Table 2 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 show in the table).

Results show that increasing model size indeed increases difference between model trained on original data and model trained on knowledge enhanced data. BlenderBot 2.7B showed best results in both BLEU and ROUGE while fine-tuned on dataset with assertions.

Table 2. Comparison of different sizes of BlenderBot

		Novelty (↓)		Diversity (↑)	
Model	**PPL**	**BLEU-1**	**ROUGE-1**	**InterDISTINCT-1**	**IntraDISTINCT-1**
BlenderBot 90M	15.82	0.1365	0.1801	**0.0434**	0.8357
BlenderBot 90M (with assertions)	15.58	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**

3.3 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 2.2, 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 3.

Table 3. 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

3.4 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 1 and 2 respectively. Both dialogues have the same initial sentence (*"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.

```
Bot #1: Hi, how are you?
Bot #2: I'm doing well. How about you? What do you like to do in your free time?
-----
Bot #1: I'm good. Reading is one of my favorite things to do. Do you like reading?
Bot #2: Yes, I do. I like to read fiction and non-fiction. What is your favorite genre?
-----
Bot #1: I like historical fiction. It's a mix of real events and fictional ones.
Bot #2: What is the name of the documentary? I would love to watch it. I have never heard of it.
-----
```

Figure 1. Example self-chat of BlenderBot 2.7 fine-tuned on original data

```
Bot #1: Hi, how are you?
Bot #2: I'm doing well. How about you? What do you like to do in your free time?
-----
Bot #1: I'm good. Reading is one of my favorite things to do. Do you like reading?
Bot #2: Yes, I do. I like to read fiction and non-fiction. What is your favorite genre?
-----
Bot #1: I'm a big fan of science fiction. My favorite author is Isaac Asimov.
Bot #2: I have never read any of his books, but I have heard of him. He was born in 1948, right?
```

Figure 2. 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 2) dialogue seems more meaningful. During first dialogue, bot#2 asks *"What is the name of the documentary?"* after the sentence *"I like historical fiction"* which seems like not an appropriate reaction.

4 Conclusion

Dialogue systems that are able to use knowledge about the world while having another conversational qualities are studied during this work. A

novel knowledge extraction algorithm was developed in order to augment existing dialogue datasets and fine-tune state-of-the-art chatbot model BlenderBot. 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. Although there are no significant changes in diversity (0.91 in IntraDISCTINCT for augmented BlenderBot vs 0.89 for original BlenderBot). Current results can be improved by replacing knowledge extraction algorithm with more complex approach (e.g. predicting knowledge paths [2]), increasing model size or changing model architecture [25].

References

[1] *The Pushshift Reddit Dataset*, Jan. 2020. Zenodo. doi: 10.5281/zenodo.3608135. URL https://doi.org/10.5281/zenodo.3608135.

[2] M. Becker, K. Korfhage, D. Paul, and A. Frank. CO-NNECT: A framework for revealing commonsense knowledge paths as explicitations of implicit knowledge in texts. *CoRR*, abs/2105.03157, 2021. URL https://arxiv.org/abs/2105.03157.

[3] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., 2009.

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

[5] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of wikipedia: Knowledge- powered conversational agents, 2018. URL https://arxiv.org/abs/1811.01241.

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

[7] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[8] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. doi: 10.5281/zenodo.1212303.

[9] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055, 2015. URL http://arxiv.org/abs/1510.03055.

[10] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

[11] M. Matsuda. Politics 2028: Why artificial intelligence will replace politicians. https://www.is.gd/oeQLI7, 2018. accessed 2021-01-12.

[12] 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.

[13] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. Bleu: a method for automatic evaluation of machine translation. page 8, 10 2002. doi: 10.3115/1073083.1073135.

[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chil- amkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. 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*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[15] H. Rashkin, E. M. Smith, M. Li, and Y. Boureau. I know the feeling: Learning to converse with empathy. *CoRR*, abs/1811.00207, 2018. URL http://arxiv.org/abs/1811.00207.

[16] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.URL https://arxiv.org/abs/1908.10084.

[17] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith, Y.-L. Boureau, and J. Weston. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

[18] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models, 2016.

[19] E. M. Smith, M. Williamson, K. Shuster, J. Weston, and Y. Boureau. Can you put it all together: Evaluating conversational agents' ability to blend skills. *CoRR*, abs/2004.08449, 2020. URL https: //arxiv.org/abs/2004.08449.

[20] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975, 2016. URL http://arxiv.org/abs/1612.03975.

[21] A. N. Vaidyam, H. Wisniewski, J. D. Halamka, M. S. Kashavan, and J. B. Torous. Chatbots and conversational agents in mental health: A review of the psychiatric landscape. *Can. J. Psychiatry*, 64(7): 456–464, July 2019.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[23] J. Weston, E. Dinan, and A. Miller. Retrieve and refine: Improved sequence generation models for dia- logue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search- Oriented Conversational AI*, pages 87–92, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5713. URL https://www.aclweb.org/anthology/W18-5713.

[24] T. Young, E. Cambria, I. Chaturvedi, M. Huang, H. Zhou, and S. Biswas. Augmenting end-to-end dialog systems with commonsense knowledge, 2018.

[25] H. Zhang, Z. Liu, C. Xiong, and Z. Liu. Conversation generation with concept flow. *CoRR*, abs/1911.02707, 2019. URL http://arxiv.org/abs/1911.02707.

[26] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too?, 2018.

[27] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4623–4629. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/643. URL https://doi.org/10.24963/ ijcai.2018/643.