RESEARCH-ARTICLE

Contextualised Word Embeddings Based on Transfer Learning to Dialogue Response Generation: a Proposal and Comparisons

THOMAZ CALASANS, University of São Paulo, Sao Paulo, SP, Brazil

ANNA HELENA REALI COSTA, University of São Paulo, Sao Paulo, SP, Brazil

EDUARDO RAUL HRUSCHKA, University of São Paulo, Sao Paulo, SP, Brazil

Published: 19 February 2021

ISEEIE 2021: 2021 International Symposium on Electrical, Electronics and Information Engineering
February 19 - 21, 2021
Seoul, Republic of Korea

https://doi.org/10.1145/3459104.3459169
ISBN: 9781450389839

Contextualised Word Embeddings Based on Transfer Learning to Dialogue Response Generation: a Proposal and Comparisons

Thomaz Calasans
Data Science Center, Universidade de São Paulo (USP), São Paulo, Brazil
thomaz.santos@usp.br

Anna Helena Reali Costa
Data Science Center, Universidade de São Paulo (USP), São Paulo, Brazil
anna.reali@usp.br

Eduardo Raul Hruschka
USP Data Science Center and Itaú-Unibanco Data Science Team, São Paulo, Brazil
hruschka@usp.br

ABSTRACT

Contextualised word embeddings have recently become essential elements of Natural Language Processing (NLP) systems since these embedding models encode not only words but also their contexts to generate context-specific representations. Pre-trained models such as BERT, GPT, and derived architectures are increasingly present on NLP task benchmarks. Several comparative analyses of such models have been performed, but so far no one compares the most recent architectures in a dialogue generation dataset by considering multiple metrics relevant to the task. In this paper, we not only propose an encoder-decoder system that uses transfer learning with pre-trained word embeddings, but we also systematically compare various pretrained contextualised word embedding architectures on the DSTC-7 dataset, using metrics based on mutual information, dialogue length, and variety of answers. We use the word embeddings as a first layer of the encoder, making it possible to encode the texts in a latent space. As a decoder, we use an LSTM layer and a byte pair encoding tokenisation, aligned with state-of-the-art dialogue systems recently published. The networks are trained during the same amount of epochs, with the same optimisers and learning rates. Considering the quality of the dialogue, our results show that there is no superior technique on all metrics. However, there are relevant differences concerning the computational costs to encode the data.

CCS CONCEPTS

• **General and reference**; • **Evaluation; Empirical studies;**; • **Computing methodologies**; • **Natural language generation.**;

KEYWORDS

Natural Language Processing, Dialogue Generation, Contextualised Word Embeddings, Empirical Comparisons

ACM Reference Format:
Thomaz Calasans, Anna Helena Reali Costa, and Eduardo Raul Hruschka. 2021. Contextualised Word Embeddings Based on Transfer Learning to Dialogue Response Generation: a Proposal and Comparisons. In *2021 International Symposium on Electrical, Electronics and Information Engineering (ISEEIE 2021), February 19–21, 2021, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459104.3459169

ISEEIE 2021, February 19–21, 2021, Seoul, Republic of Korea

ACM ISBN 978-1-4503-8983-9/21/02…$15.00
https://doi.org/10.1145/3459104.3459169

1 INTRODUCTION

Creating a machine that can act and think like a human being has been a dream of humanity since before the invention of the computer. The search for machines that can think like humans have raised technical and philosophical questions about the very nature of consciousness [4]. One field of computer science that deals with the creation of such a machine is Natural Language Processing (NLP), particularly in the area known as "dialogue generation". Dialogue systems have become increasingly popular in the last few years. In the shape of virtual assistants [12] they can be a simple interface in human-computer interaction, being intuitive and transparent for the user like, for instance, agents like Siri,[1] Alexa,[2] and Google Assistant.[3]

A drawback of the current approaches to dialogue systems is the lack of variety of responses.

It happens because most of the assistants have been developed using a retrieval-based approach, which uses a pre-defined set of phrases to retrieve from, thereby creating a limited set of interactions and clearly distinguishing itself from a true human-human interaction. This may be an acceptable behaviour when solving some very specific tasks for strict domains, but it might reduce the dynamic of the conversation and contribute to disengage the users in open domains, in which it is impractical to predefine all the possible subjects that could be spoken by the user.

To overcome this drawback, it is possible to use neural dialogue generation techniques [16]. There are some generative neural network models with sequence-to-sequence (SEQ2SEQ) architectures that can be trained to map from the current user context to the appropriate response [23]. From combining those models with techniques like beam search, the dialogue system can generate sentences that were never seen in the training set, yielding to a more varied dialogue. The main problem with this technique is the amount of data needed to train it, which is much larger as compared to retrieval-based methods.

It is worth anticipating that, even with such dialogue generation techniques, dialogue systems can still have the same issues as retrieval-based approaches. One issue that is common to both retrieval-based and generative approaches is the incidence of uninformative responses like "I don't know what you are talking about"

[1] https://www.apple.com/siri/
[2] https://www.alexa.com/
[3] https://assistant.google.com/

and "I don't know what you are saying" [15]. This happens in retrieval systems when they do not have an appropriate response to the intention of the user. In generative-based systems, the model can learn that those responses can make sense to almost any dialogue context, leading it to overfit on generic responses. In both cases, those kinds of answers are not specific to any utterance and do not stimulate the user to engage in further conversations.

Aimed at circumventing the problem of the high incidence of non-informative responses generated by the chatbot, our first proposal is to use transfer learning, with pre-trained word embeddings, to leverage the information acquired in other tasks that have larger datasets, thus creating a more accurate dialogue generation model. The model that we used to do that is based on the popular Encoder-Decoder framework.

The second contribution of this paper is to assess the quality of the pre-trained word embeddings by taking into account multiple dimensions for assessing dialogue systems. Our underlying assumption is that for a successful conversation with a human, the chatbot has to not only suitably engage in the conversation, but also stimulate new topics of conversation. For each one of those dimensions, we attribute a quantitative score and use it to compare five of the most recently published pre-trained word embeddings.

In summary, the two main contributions of our paper are: first, we propose a system that uses transfer learning with pre-trained word embeddings; second, we assess the quality of the pre-trained word embeddings with a new framework that takes into account multiple dimensions for assessing dialogue systems. The remainder of this article is organised as follows. Section 2 covers the main categories of dialogue generation systems. It is complemented by Section 3, which gives an overview of contextualised word embeddings, followed by our proposed encoder-decoder system in Section 4. Next, Section 5 describes the metrics proposed to evaluate them. Then, in Section 6, we describe the adopted experimental methodology. Finally, Section 7 presents the obtained results, followed by Section 8 that concludes the paper.

2 DIALOGUE GENERATION

Chatbots based on dialogue generation systems usually fall into two categories according to their objectives: **closed domain** [11] and **open domain** [22]. The former has a limited scope such as booking a movie ticket, a restaurant or finding a product. It aims to complete a user goal quickly with as little as possible interactions with the user, and for that, it can respond to simple patterns from the user's input to identify intentions and entities. Open-domain chatbots, on their turn, should be able to handle a conversation on various subjects and to engage the user with specific and coherent answers.

Multiple techniques have been proposed to solve the task of open domain conversation, from complex systems [9] to end-to-end neural networks trained on large datasets [28]. However, the problem is still considered to be unsolved as no chatbot at the time of writing this article was able to maintain an engaging and coherent conversation with a human for even 20 minutes [13].

End-to-end networks have the advantage of requiring less human-engineered elements, as the entire pipeline is supposed to be learned from data. Those networks can be trained like a machine translation [20] problem that maps the prior context and the user utterance to the expected answer. Such networks started to gain more visibility with the creation of the Transformers [25] architecture, a SEQ2SEQ neural network that has achieved the state-of-the-art performance on multiple NLP tasks. Its advantages come from the fact that it is designed to process sequential data, which is the case of dialogue datasets. It can be trained in a self-supervised way and used with transfer learning methods to leverage the information of large corpora and be fine-tuned for subsequent tasks like dialogue generation.

The use of a model that was pre-trained on a task and that is then reused on a new task with similar characteristics usually yields to the so-called transfer learning approaches [24, 26]. The idea underlying those approaches is that the pre-trained model will leverage the knowledge extracted from the ideally large amount of the available data for the first task to achieve better performance at the second task. These approaches became very popular in the field of deep learning because (i) they allow reducing the amount of data needed to train complex neural network models and (ii) because of the modular nature of neural networks, which makes it easy to connect new layers on top of pre-trained ones, fine-tuning the last layers of the model on a new task.

3 CONTEXTUALISED WORD EMBEDDINGS

The most common way to apply transfer learning methods on NLP tasks is by using a word embedding, which refers to the technique of representing each word, or token,[4] as a dense vector. Typically, these vectors are a representation of a vocabulary with reduced dimensionality and semantic information added [17]. The core idea behind word-embeddings training is that "You shall know a word by the company it keeps!" [7], meaning that the vector for each word is roughly learned by predicting its surrounding words in the corpora.

Initial approaches to word embedding used one vector for each word [2, 17, 18]. They were a fast way to train the models but failed to capture much of the syntactical information and they did not handle polysemic behaviour.[5]

Recent works use a contextual approach, where the embedding of each word depends on its context in the phrase. Those techniques are computationally harder to train, but they usually give better results on NLP benchmarks.[6]

Particularly, models based on the aforementioned Transformers networks are increasingly present on the top of those rankings. This kind of network is based on the encoder-decoder architecture, as shown in figure 1. The encoder side receives the entire input sequence and projects it into an intermediary dimension, whereas the decoder side receives iteratively each decoded token and combines this information with the encoded input to predict what should be the next token. We here highlight two of such model families, each one created using one side of the Transformers architecture, namely: BERT (Bidirectional Encoder Representations from Transformers) [5] and GPT (Generative Pre-trained Transformer) [19].

BERT and its variations [5, 14] adapt the encoder side of the Transformer (left-hand side half of figure 1), meaning that the entire

[4] A token is a string of contiguous characters between two spaces, or between a space and punctuation marks. A token is a result of parsing the document down to the atomic elements of a language.
[5] Polysemy is the capacity for a word or phrase to have different, but related, semantics.
[6] https://paperswithcode.com/area/natural-language-processing

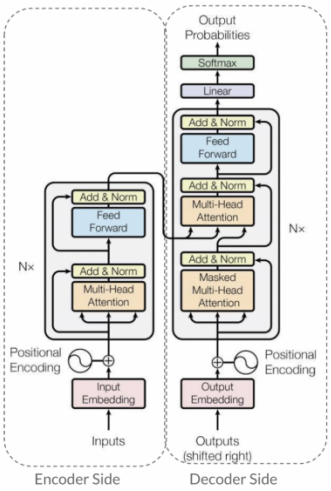

Figure 1: The transformer - model architecture[25].

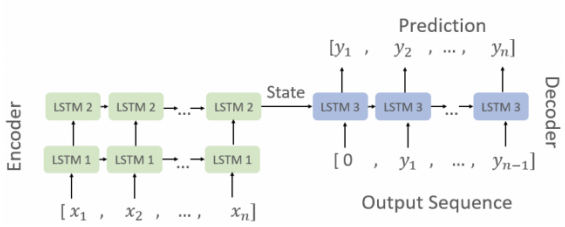

Figure 2: Detailed architecture of the Encoder-Decoder schema. The vector $[x1, x2, \cdots, xn]$ represents the embedded input. State is a fixed size vector and $[y1, y2, \cdots, yn]$ is the output sequence, that is fed with a lag of one time-step to the Decoder side.

input text goes through the model at training and inference time. To train the model, BERT uses a masked language modeling (MLM) loss, where some of the tokens are masked during training time. The model's objective is to reconstruct the original text. Due to being based on MLM loss, BERT is considered a denoising autoencoder. Giving the nature of this loss, to use BERT as a generative model without any adaptation, one has to pre-define the number of tokens that will be generated and pass it as a masked input to the model.

GPT, on the other hand, adapts the decoder side of the Transformer (right-hand side half of figure 1), where the input text is observed by the model in a sequential way. At every iteration, the model receives all the previous tokens to predict the next one. For that reason, GPT is considered a generative model.

4 PROPOSED ENCODER-DECODER SYSTEM

To deal with the differences between the generation task of both model families, we propose the use of an Encoder-Decoder architecture. This kind of architecture consists of a neural network that can be abstracted by having two sides, the encoder and the decoder. The encoder side receives the input and projects it in a fixed-size vector, which will be later processed by the decoder to generate the output.The output of the decoder is a sequence of tokens. For that reason, this is a generative model — and not a retrieval one — that tackles the problem of a limited variety of responses.

Since this kind of network has the property of projecting the input in a fixed-size vector, it is appropriate to deal with inputs of different natures without the need for further adaptations. To illustrate this property, we use the embedding model as the input of the encoder layer of our model. More precisely, we use the traditional encoder-decoder architecture [3] represented by figure 2 with some hyperparameters proposed by us. Both the encoder and decoder sides consist of a single layer LSTM [10] block (with 512 hidden units).

The encoder part receives the input text embedded by the pretrained model as a sequence of vectors. This sequence goes through the first LSTM layer and, then, after processing the entire input sequence, the internal state of the last LSTM block is used to initialise the decoder block. The initialised decoder layer receives at each iteration one of the tokens of the output text, delayed by one step and embedded by a non-initialised embedding layer (with 256 dimensions). To tokenise the output text we use a byte-pair encoding [21] with 8K BPE subwords.

5 EVALUATION METRICS

Even with the recent improvements in the field of dialogue generation, chatbots are still very far from having human-like behaviour [13]. From this perspective, to account for the most common deviations we propose three evaluation metrics inspired on a recent work [16]: Mutual Information (MI), Information Flow (IF), and Dialogue Length (DL). We shall note that Li et al. [16] use them in a context different from ours, i.e., as reward functions for reinforcement learning.

5.1 Mutual Information (MI)

To evaluate if the generated response makes sense in the context of the dialogue, we measure the mutual information between the response $r$and the combination of the context $c$with the last user utterance u:

$$MI = \frac{1}{N_r} log p_{seq2seq}(r|u,c) + \frac{1}{N_u} log p_{bkw\_seq2seq}(u|r), \quad (1)$$

where N_r is the amount of tokens on the generated response,$N_u$is the amount of tokens on the last utterance, $p_{seq2seq}(r|u,\ c)$ refers to the probability of generating the response r giving the previous dialogue (u, c), and $p_{bkw\_seq2seq}(u|r)$ refers to the probability of the last user utterance $u$giving the response r. Both probabilities are calculated using the cross-entropy function. In particular, the former is computed by using(u, c)as input and $r$as output for our model, and the latter using $r$as input and $u$as output for a model trained with the inverse order of the dialogue.

5.2 Information Flow (IF)

In order to avoid repetitive cycles of conversation, we propose a metric that evaluates the new information added by each response. Models that tend to generate responses too similar to the last user utterance get a lower IF score:

$$IF = -log cos\,(h_r, h_u)\,, \quad (2)$$

Table 1: Reddit dialogue samples.

A) TIL the entire goldeneye 007 game on the n64 is only 12 mb.
B) This game also introduced the first deadly full automatic pencil.
C) I love how thousands of groups of pre-internet kids all independently decided to call it the pencil
John Travolta turned down the role of Forest Gump.
Momma always said life is likely a royal with cheese.
C) If I remember correctly, I'm pretty sure it conflicated with shooting of Pulp Fiction

where $cos(h_r, h_u)$ measures the similarity of the embedded vectors h of the user utterance u and the generated response $r$using the cosine similarity, which is a measure of similarity between two non-zero vectors of an inner product space.

5.3 Dialogue Length (DL)

It is a hard task to define when a dialogue has ended. Other than a farewell message, in interactions with chatbots, an important element to define the continuation of the dialogue and the user engagement is the indicator of "I don't know" or IDK [1]. When a chatbot misunderstands the user's intentions and, as a consequence, starts answering with IDKs, the dialogue is soon coming to an end. To evaluate the propensity of the model to misunderstand the user's utterance, we propose a metric that measures the probability of IDKs:

$$DL = -\frac{1}{N_S}\sum_{s \in S}\frac{1}{N_s} log p_{seq2seq}(s|r), \tag{3}$$

where S is a list of IDK responses manually constructed, N_S is the number of IDK phrases on the list, N_s is the number of tokens in each IDK phrase, and $p_{seq2seq}(s|r)$ is the log-likelihood of the answer to be generated by each model. Even though there are multiple ways to generate an IDK, we expect that they will fall in a similar region of the space.

6 EXPERIMENTAL METHODOLOGY

This section describes the datasets used in the experiments performed to compare the pre-trained embeddings, the list of the pre-trained models, and the parameters of the training process.

6.1 Training Data

The selected use case considers the context of an open domain undirected dialogue, where each actor is free to speak what comes to mind without a specific goal or direction. We extracted the dialogue corpus from a collection of public domain conversations known as DSTC-7 [8]. This dataset contains over 2 million dialog turns collected from the Reddit[7] forum. Table 1 shows some examples of excerpts from Reddit dialogues. The training data comprises the period from Jan-2013 to Dec-2016 and the validation data is from the period from Jan-2017 to Mar-2017.

We applied some filters to have a higher quality dataset. Responses, where any of the following conditions apply, were removed:

1. Dialogue depth higher than 5;
2. Response contains a URL;
3. Response length higher than 130 characters;
4. Message potentially offensive;
5. Message repeated on the dataset more than 30 times;
6. Percentage of stop words[8] in the message above 70%.

After running the filters, we had left approximately 1 million pairs of dialogue history/response for training and 55k pairs for validation.

6.2 Pre-trained Models

We selected five state of the art pre-trained models that are among the top scores on multiple NLP benchmarks:

BERT [5]: The original BERT, on the small version.

ALBERT[14]: A more parameter-efficient version of BERT.

XLNet [27]: An autoregressive version of BERT.

GPT [28**]:** The original generative pre-training model, on its small version.

GPT-2 [19]: The second generation of GPT, trained with more data.

All models have the same embedding dimension of 768. We used a widely known Python implementation publicly available.[9]

6.3 Model Training

The models were trained during 20 epochs each, with 2,000 batches with a size of 64 observations/batch. This means that, on average, each network got to process the entire dataset 2.5 times. We used the Nadam optimiser [6] with a learning rate of 1e-4 and the sparse cross-entropy loss.

7 RESULTS

Using the methodology just described in the previous section to compare the pre-trained models, we got table 2, where we compare the five embedding models with the metrics discussed in Section 5. Also, we use an additional metric that is the average time that each model took to embed a text of the validation dataset.

We can observe that no model is superior to the others on the proposed metrics. XLNet demonstrated to be the embedding method with the best performance on mutual information, which is related to the way that traditional dialogue generation models are evaluated. However, it has the highest computational cost and is not the best model on the other metrics that are also important in a multi-turn dialogue. Since our methodology relied on training a model with each embedding technique for the same amount of epochs and not the same amount of time, the increase in computational cost

[7] https://www.reddit.com/

[8] Stopwords are words which are filtered out before or after processing of text data, and usually refers to the most common words in a language.

[9] https://github.com/huggingface/transformers

Table 2: Comparison results. MI stands for Mutual Information, DL for Dialogue Length, IF for Information Flow and Emb.Time for the average time (in milliseconds) to embed an observation.

Embedding	MI	DL	IF	Emb.Time
BERT	5.37	3.63	0.172	**5.7**
ALBERT	3.67	**3.65**	**0.226**	6.2
XLNet	**9.64**	3.62	0.132	10.9
GPT	7.70	3.55	0.135	6.0
GPT-2	6.02	3.62	0.131	**5.7**

may have played a significant role in the performance of XLNet. ALBERT, on the other hand, is slightly superior on dialogue length and the clear winner regarding information flow. However, it shows the worst performance for mutual information.

8 CONCLUSION

We proposed an encoder-decoder system that uses transfer learning with pre-trained word embeddings. Also, we assessed the quality of the pre-trained word embeddings with a new framework that takes into account multiple metrics. Our main finding is that there is no clear winner for all the metrics used to evaluate the generation of dialogues. From this perspective, our results can help practitioners to choose a more suitable word embedding for dialogue generation problems.

We shall note, however, that our analyses are limited to a single dataset, making use of a particular set of hyperparameters. As future work, we plan to build a dialogue generation system that combines transfer learning with the best performing word embedding, making use of a reinforcement learning approach.

ACKNOWLEDGMENTS

This work was supported by CAPES (Finance Code 001), CNPq (Grants 425860/2016-7 and 307027/2017-1), and Itaú-Unibanco. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding institutions.

REFERENCES

[1] Ian R Beaver. 2018. Automatic Conversation Review for Intelligent Virtual Assistants. https://digitalrepository.unm.edu/cs_etds/93 (2018).
[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. arXiv:1607.04606 (2016).
[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078 (2014).
[4] Ron Chrisley. 2008. Philosophical foundations of artificial consciousness. Artificial Intelligence in Medicine 44, 2 (2008), 119–137.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423
[6] Timothy Dozat. 2016. Incorporating Nesterov Momentum into Adam. ICLR 2016 Workshop (2016).
[7] John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. Studies in Linguistic Analysis, Philological Society, Oxford (1957).
[8] MichelGalley, ChrisBrockett, XiangGao, BillDolan,and Jianfeng Gao. 2018. End-to-End Conversation Modeling: Moving beyond Chitchat. http://workshop.colips.org/dstc7 (2018).
[9] Nuria Haristiani. 2019. Artificial Intelligence (AI) Chatbots Language Learning Medium: An inquiry. In Journal of Physics: Conference Series, Vol. 1387. IOP Publishing, 012020.
[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural Computation 9, 8 (1997), 1735–1780.
[11] Vladimir Ilievski, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2018. Goal-oriented chatbot dialog management bootstrapping with transfer learning. arXiv:1802.00500 (2018).
[12] H. N. Io and C. B. Lee. 2017.. Chatbots and conversational agents: A bibliometric analysis. 2017 IEEE International Conference on Industrial Engineering and Engineering Management. (2017.). https://doi.org/10.1109/IEEM.2017.8289883
[13] Chandra Khatri, Anu Venkatesh, Behnam Hedayatnia, Raefer Gabriel, Ashwin Ram, and Rohit Prasad. 2018. Alexa Prize—State of the Art in Conversational AI. AI Magazine 39, 3 (2018), 40–55.
[14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. arXiv:1909.11942 (2019).
[15] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, San Diego, California, 110–119. https://doi.org/10.18653/v1/N16- 1014
[16] Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep Reinforcement Learning for Dialogue Generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, 1192–1202. https://doi.org/10.18653/v1/D16-1127
[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems. 3111–3119.
[18] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 1532–1543.
[19] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. https://openai.com/blog/language-unsupervised/
[20] Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, 583–593.
[21] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. arXiv:1508.07909 (2015).
[22] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In Thirtieth AAAI Conference on Artificial Intelligence. 3776–3783
[23] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence- learning-with-neural-networks.pdf
[24] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI Global, 242–264.
[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems. 5998–6008.
[26] Jeremy West, Dan Ventura, and Sean Warnick. 2007. Spring research presentation: A theoretical foundation for inductive transfer. Brigham Young University, College of Physical and Mathematical Sciences 1 (2007), 32.
[27] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv:1906.08237 (2019).
[28] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. arXiv:1911.00536 (2019).