

Open-Domain Conversational AI with Hybrid Generative and Retrieval Mechanisms

Jishnu Ray Chowdhury
jraych2@uic.edu
University of Illinois Chicago

Mobashir Sadat
msadat3@uic.edu
University of Illinois Chicago

1 INTRODUCTION

Conversational bots are bots which can engage in conversations with a partner in natural language. They can be used as virtual tutors, digital assistants, customer service, virtual therapists, task-oriented services, and entertainment. Conversational bots often come in broadly three forms of models - (i) rule-based model (ii) retrieval (IR) model, and (iii) generative model. Each of these variants have their own advantages and disadvantages. In this project we attempt to combine different aspects of all these approaches with a predominant focus on retrieval and generation. Our overall model is a synergy of multiple sub-modules for retrieval, classification, generation, and ranking. We focus on building an open-domain chatbot. Our key contributions are listed below.

Contributions:

- (1) We provide a unique method to combine classification, generation, and retrieval for open-domain dialogue.
- (2) We combine MMI-inspired¹ [7] ranking with a standard ranking method to score candidate responses (both retrieved and generated ones).
- (3) We propose and motivate future directions to enrich the model.

2 RELATED WORKS

ELIZA [16] and ALICE [15] are some of the earliest implementations of chatbots. They usually rely entirely on explicit pre-defined rules or pattern matching schemes. Mitsuki Bot² which won the Loebner Prize in 2013, 2016, 2017, and 2018³, is also based on a similar scheme. This goes to show that 'ALICE'-like models are still competitive. However, rule-based bots are not usually as flexible, adaptive or scalable. It can be a life-long work for a bot-master to periodically review chat-logs and incrementally improve their rule-based bots.

Recent research on open-domain chatbots are mostly based on retrieval or generation. Generative models can generate its response given a section of the conversation history. The newer generative bots are usually some variant of the neural seq2seq architecture [12, 14]. There are many works on developing conversational agents on seq2seq, addressing the typical issues of seq2seq (low diversity, generic responses, inability to track user characteristics etc.⁴). One notable Recurrent Neural Network (RNN-based) neural conversational model that hierarchically encodes the conversation history is HRED [10]. HVRED [11], in addition to HRED, uses a Variational Autoencoder [6] based objective to create a latent variable to guide the response decoding process. Recently, however,

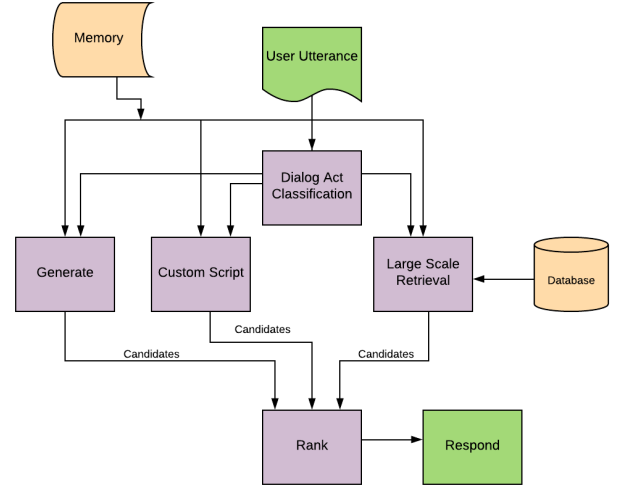


Figure 1: Model Overview.

Transformer-based models are gaining prominence. Some of the notable Transformer-based generative models are DialogPT [23] and Transfer-Transfo [19]. They are based on fine-tuning large pre-trained language models

Besides generative models, there are also retrieval based models which retrieve existing responses from a dataset based on some measure of relevancy between the responses and the user utterance in the given context. One classic deep learning-based approach is using a dual LSTM encoder (as a Siamese Network) [8] to encode both the input utterances and the responses or queries in a dataset, to predict the probability of that the pair is relevant. The newer methods are based on Transformer-based PolyEncoder models [3, 5].

There are also multiple hybrid models that are ensembles of various approaches. Particularly most papers associated with Alexa Prize⁵ are combinations of various modules. One recent paper [20] also explores this direction.

3 METHOD

Our overall model has five broad modules. The first module is based on retrieval from a collection of custom scripts with query-response pairs which are specifically made for the bot. The second module is dialogue act classification based on which different downstream decisions are made. The third module is large-scale retrieval from a large Reddit corpus. The fourth module is a generative model based

¹<https://github.com/jiwei/Jiwei-Thesis>

²<http://www.square-bear.co.uk/mitsuku/home.htm>

³<https://www.aish.org.uk/events/loebner-prize>

⁴<https://github.com/ricsinaruto/Seq2seqChatbots/wiki/>

⁵<https://developer.amazon.com/alexaprize>

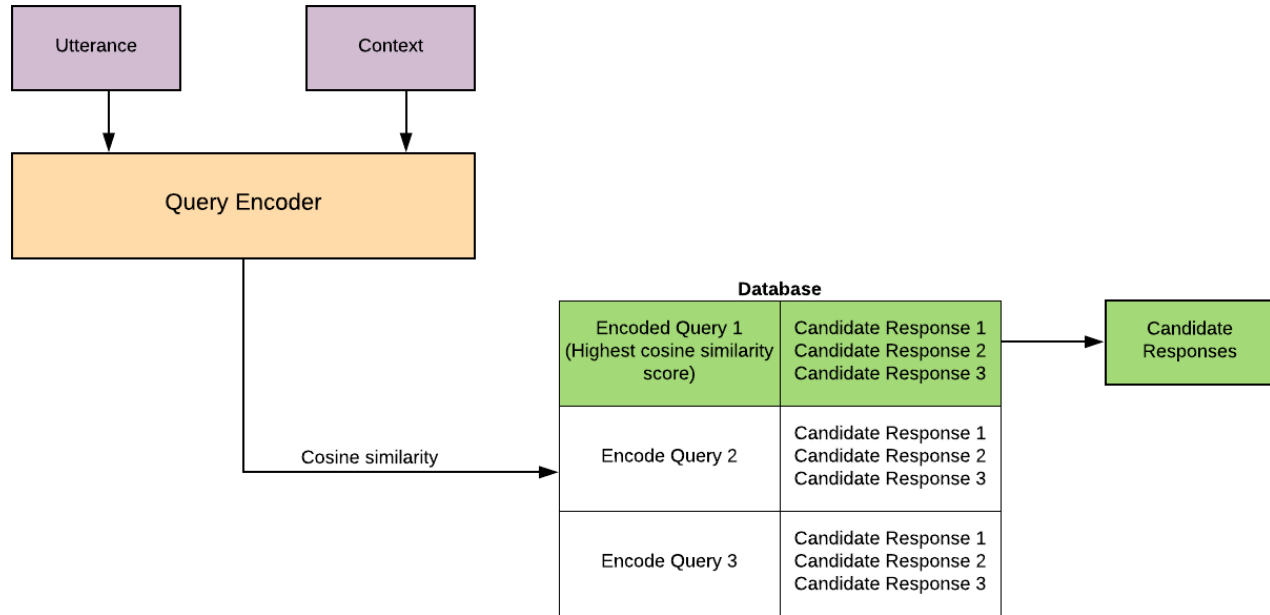


Figure 2: Retriever

on DialoGPT [23] which is OpenAI’s GPT-2 after being fine-tuned in Reddit Data. The fifth module is a ranker of scripted, retrieved, and generated candidates.

3.1 Meta-Sentence Embedding

Almost all of our modules utilizes some form of sentence embedding. Inspired from Nina Poerner, et al [9], to encode sentences or utterances we take a meta-sentence embedding where we concatenate multiple embeddings from different pre-trained models. For this project we settled on using the concatenation of ConveRT (multi-context version)⁶[3] and Universal Sentence Encoder QA (USE-QA)⁷[2, 21]. To encode queries or user utterances we concatenate ConveRT context encoder (with previous 5 turns of conversation history as extra context) and USE-QA query encoder. To encode candidate responses (can be generated, retrieved or scripted), we concatenate ConveRT response encoder and USE-QA answer encoder with the previous response as context. Both encoder are based on pre-trained Transformers. USE-QA is a multilingual model which was fine tuned in SQuAD retrieval task. ConveRT was explicitly trained on Reddit data for conversational purposes making it especially suited for our task.

3.2 Scripted Response Module

When constructing the scripted response module we kept in mind the Zipf distribution law. We wanted this module to handle common but frequent kinds of questions and utterances ("How are you", "Who are you", "what do you do?" etc.). This module should

also handle a lot of personal questions towards the bot. This module also gives some freedom to the bot-developer to customize their bot for specific tasks and purposes. Our "scripting" consists of mapping potential encounter-able utterances or queries to a list of candidate responses. Besides personally handcrafting some of the scripted pairs, we also create a second set of mappings by extracting data from Chatterbot Corpus⁸. All the queries in the scripts are pre-encoded using the query encoder defined in the previous subsection. We use a retrieval mechanism on this module; precisely, we compute cosine-similarity scores between encoded user utterance and the encoded queries in the script. We then find the query with the maximum score and retrieve all the candidate responses associated with that query. To some extent, it is analogous to classical AIML bots (instead of doing regex-based pattern matching we are doing soft matching in semantic space). The retrieval mechanism is shown in Figure 2. Some of the queries in the script are mapped to 'command codes'. For example "tell me a joke" is mapped to the command code "<JOKE>". Command codes, when detected, specific tasks are executed. For example, upon encountering "<JOKE>", the bot may randomly retrieve some joke from r/jokes subreddit data and respond with it. This same technique can be used to implement some task oriented services to our bot.

3.3 Dialog-Act Classification Module

Dialog-Act classification uses a simple MLP for classifying dialog acts. We use the MIDAS dataset⁹[22] along with its annotation scheme for training. As an input it receives the query encodings

⁶<https://github.com/PolyAI-LDN/polyai-models>

⁷<https://tfhub.dev/google/universal-sentence-encoder-multilingual-qa/2>

⁸<https://github.com/gunthercox/chatterbot-corpus>

⁹https://github.com/DianYu/MIDAS_dialog_act/tree/master/da_data

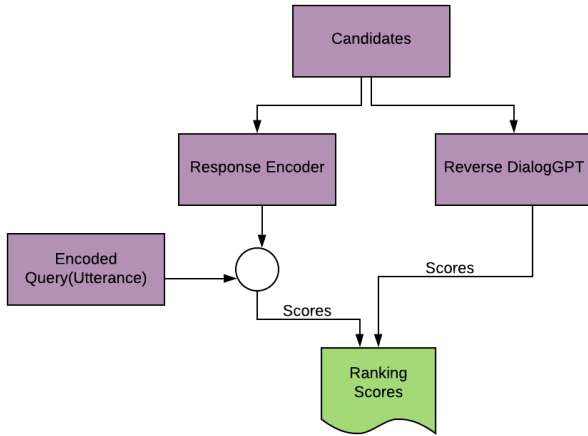


Figure 3: Ranker

described in section 3.1 (thus, it also takes into account previous turns). This module can classify a given utterance given its context into dialog acts like command, open factual question, open opinion question, opinion, statement, back-channeling etc. There are a total of 24 tags. It is a multi-classification tasks where some samples have two gold classes and some have one gold class. We use separate binary classifiers for each class with a sigmoid function. For prediction we use the two classes with the highest score (predicted probability). We also add a "<NULL>" class. So if one of the predicted class is "<NULL>" it means there is only one dialog act. Based on the predicted dialog acts, different downstream decisions are made which will be explained in subsection 3.7.

3.4 Large-Scale Retrieval Module

This subsection uses the same mechanism as 3.2 for retrieval but here it is done on organically created Reddit data (not something handcrafted for a conversational AI) and at a much larger scale. We used SQL to retrieve some of the highly upvoted threads and comments from the Reddit database in Google Big Query¹⁰. We then pre-encode the queries and fill in the relevant data in local sqlite database so that we can easily find queries and candidate responses through some simple SQL. Using a database is also much easier on RAM. We used AWS to pre-encode the queries and fill the database.

3.5 Generative Module

This module is based on the currently SOTA DialoGPT model which is GPT2 fine tuned on millions of samples of reddit dialogues. It is based on Huggingface's Transformer's library[18]¹¹. We use their pretrained weights¹². The original decoding code is not publicly shared, so we attempted to implement it ourselves. We started with some publicly shared codes¹³, but they do not use beam search

or MMI [7] which were used for the SOTA variant of DialoGPT. We tried to implement beam search but had difficulties generating diverse candidates. Simple local nucleus sampling [4] at each time step gave much more diverse responses. We tried different approaches to diversify beam search [13] but to no avail. Finally, we settled on simply using greedy nucleus sampling multiple times parallelly (in a batch) to generate multiple candidate responses. In this way, we generated 30 responses for each turn. We also use the last 3 turns of conversation history as extra context for the generator. The generator can sometimes generate good responses and sometimes had ones for the same user utterance. Thus when creating multiple candidates the chance of creating at least some good responses increase. Then we can use a ranking module to select from the better responses.

3.6 Ranking Module

The bot first generates or retrieves multiple candidates for a given utterance or query. These candidates are then scored in this ranking module. Based on the scores the top 3 candidates are chosen. Their scores are then normalized to be converted into a probability distribution and then a response is finally randomly sampled based on the probability distribution. During sampling we also check the conversation history to avoid repetitions if possible. The ranking module has two sub-components. One component computes cosine-similarity scores between query encodings and response encodings (the encoding methods are described in subsection 3.1). For the other component we took some inspiration from the Bidi anti-LM method introduced by Li et al. [7] for maximal mutual information. We would prefer less generic responses to ranked higher than the more generic ones. Intuitively, since generic responses applies to a large space of queries and utterances it is hard to estimate the locality of the semantic space of the original utterance and query to which it was directed; however, from a more specific and informative query estimating the theme of the original query or utterance can be easier. For example, it is much easier to estimate that the original utterance or query is related to Quantum Loop Gravity or something from a response: "Loop Quantum Gravity Theory attempts to unify relativity and Quantum Mechanics" compared to the response: "I don't know". Based on this intuition, we score each candidates by using the cross entropy error for generating the original utterance from the response from a 'reverse-generator'. This 'reverse-generator' is again DialoGPT but with different weights and configurations. This version of DialoGPT was trained to predict the utterance or query context from the responses. Note that we are not using the full anti-LM bidi objective which also uses log-probability scores for forward generation and length penalty. While we have the log-probability scores immediately available for the generative sequences, we do not have them for the retrieved data. Though it is possible to generate the scores by running them on forward generator, for the sake of simplicity we ignore them for now and only use the cross-entropy loss of the reverse generator for the MMI-based component.

The scores from the two components is then combined using the following formula which we created:

$$\alpha N(\text{Scores}_{\text{cosine}} + \text{Bias}) + \beta \cdot N(1 - N(\text{Loss}_{\text{reverse}}))$$

¹⁰<https://pushshift.io/using-bigquery-with-reddit-data/>

¹¹<https://github.com/huggingface/transformers>

¹²<https://github.com/microsoft/dialogpt>

¹³https://colab.research.google.com/drive/1-KjLAV3J1IVDw_9KogjKDCzgFY7Jp7E

Say something: Hello, how are you?

Bot: Not a lot is going on for me at the moment

Say something: What is your name?

Bot: I don't really have a name.

Say Something: Do you like movies? If so, what kind?

Bot: : I like them all as long as the storyline and narration is good.

Say Something: Are you sentient?

Bot: : May be I am a bot pretending to be a human. May be I am a human pretending to be a bot. May be I am a bot pretending to be a human pretending to be a bot.

Say Something: Tell me a Joke.

Bot: :I just saw a midget get pickpocketed..... I still can't believe someone could stoop so low.

Say Something: What can you tell me about Loop Quantum Gravity?

Bot: : Quantum field theory put simply is relativistic quantum mechanics where you can create and destroy particles. It's not "the" quantum field theory; there are lots of quantum field theories. For example, quantum electrodynamics describes electromagnetic interactions.

Say Something: Do you know anything about Sellar's myth of the given?

Bot: : That is foundationalist epistemology, i.e. the notion that one can found & justify belief on an extra-theoretical basis. And he argues that this is a faulty position. <https://plato.stanford.edu/entries/sellars/#4> So, Sellars is anti-foundationalist-- instead claiming that all concepts, knowledge, even truth itself is wrapped up in a framework of inference. There's no such thing as a properly basic foundation to knowledge.

Say Something: I am quite tired today; I didn't know that the project deadline was so soon. I had a sleepless night working on the project.

Bot: : you can't make excuses for laziness; this is no big deal, but it's a slippery slope man. i'm a hypocrite though, i usually don't study and also don't sleep sooo

Say Something: Did you know I accidentally deleted major portions of the project and had to reconstruct it again? Version control is important huh. Who knew?

Bot: : Reminds me of the time I was working on a web development project in uni, the night before due date I kinda blew up the php code, and couldn't figure out what I broke.

Say Something: Just my luck, I have three project presentations in the same day.

Bot: : CLOSE REDDIT. GO ALL IN. YOU CAN DO IT. (also, what's the project about?)

Say Something: By the way, can you tell me about Language Models?

Bot: On 1B dataset, [Exploring the Limits of Language Modeling](<https://arxiv.org/abs/1602.02410>) has 30 (23.7 with ensembles)

Say Something: Please tell me more about them.

Bot: *Last year when I had to bake bread in the middle of the school day.* This sounds like it has a nice story behind it so I'd like to hear it :)

Table 1: Conversation with Scripts+Retrieval+Ranker.

α, β are scalar weights that determines how much weight is to be given to the query-response matching based scores and how much of it is to be given to the reverse generation loss scores respectively. We set them as 0.4 and 0.6 respectively after some not-very-exhaustive experimentation and qualitative analysis. Ideally this can be tuning can be done with reinforcement learning in a human in a loop setting or in some other way; but we are keeping things simple for now. A separate bias term ('Bias') is present for every candidate. We use the 'Bias' term to bias the candidate scoring for Reddit-retrieved candidates. This term can also be using for biasing the response candidates from a specific source or module. We use this to add bias towards ranking retrieval candidates because

otherwise we found our model was often bit too biased towards generated responses even in some cases where the available retrieved candidates were preferable based on our subjective judgments. N is a normalizing function which converts the score to a probability distribution:

$$N(x_i) = \frac{x_i - \min(x)}{\sum_i x_i}$$

Something like softmax can also be used alternatively. After computing the scores, we filter most of the lower scoring candidates, normalize the scores of the remaining ones into a probability distribution based on which the final response is sampled. We find that the probability distribution is sometimes quite flat; so if we

Say something: Hello, how are you?
Bot: Am well. How about you?
Say something: What is your name?
Bot: I am Not Daniel Marsch. What is yours?
Say Something: Do you like movies? If so, what kind?
Bot: : No, but really, I like movies in general so I'll probably pick a genre!
Say Something: Are you sentient?
Bot: : I am sentient, so yes I am
Say Something: Tell me a Joke.
Bot: : I am a joke!
Say Something: What can you tell me about Loop Quantum Gravity?
Bot: : I remember as child doing quantum gravity when I was watching Super Star : The Movie the feels stopped.
Say Something: Do you know anything about Sellar's myth of the given?
Bot: : Nope. I thought it was a well known Science Fact of the Day, but I never knew it was actually proven.
Say Something: I am quite tired today; I didn't know that the project deadline was so soon. I had a sleepless night working on the project.
Bot: : Is the project as impressive as the last and the rainy day aliens.
Say Something: Did you know I accidentally deleted major portions of the project and had to reconstruct it again? Version control is important huh. Who knew?
Bot: : I knew.
Say Something: Just my luck, I have three project presentations in the same day.
Bot: : I'm trying to recreate those presentations because Sapphire
Say Something: By the way, can you tell me about Language Models?
Bot: I got hooked on the language models
Say Something: Please tell me more about them.
Bot: Just stick on the classics and type out some test.org stuff. Then tell me what you find when you get to their.org page

Table 2: Conversation with Generator+Ranker.

don't filter, it can be easy to sample lower ranked candidates which we do not want. The ranker module is shown in Figure 2.

3.7 Module Interaction

When a user enters an utterance it is first encoded using the query encoder. The encoding is then first compared with our handcrafted query-response pairs in the scripted response module. If there is a high-confidence match beyond the threshold of 0.75 cosine similarity, we simply rank the associated candidates and return the response. If some candidates have command codes we execute the related task. Otherwise if the maximum cosine similarity score is less than the threshold, we store the scripted candidates and move on to the next sub-module where we classify the dialog acts of the utterance. Based on the classified dialog acts different decisions are made in a rule-based fashion (we just use "if-else" conditions). For example, if the dialog act is related to a factual questions or command or such we add some bias for retrieved candidates (as described in subsection 3.6) otherwise not. For certain dialog acts which does not require an exact response, there is also a chance to go to an 'initiate' mode where the bot may bring up some random fact or joke. Some special handcrafted-response-candidates are also mapped with the dialog act class themselves. The specific details of

the rules can be found in the code that we will make available. If the model does not find its responses in the dialog-act submodule it goes into the chatterbot-scripted response submodule and extracts candidates from there. If the maximum cosine similarity score is still less than 0.75 it goes to the large-scale retrieval and generation module. From all these modules candidates are collected along with their source-based bias and sent to the ranker which returns the final response. The model overview is shown in Figure 1.

4 EVALUATION

Our dialog acts classification model achieves a performance of about 86.12% on test data. For retrieval and generation we are using pre-trained modules which were already evaluated on quantitative measures on previous works [2, 3, 21, 23]. In our case, it is difficult to make quantitative analysis because we don't have any ground truths for the overall model. Instead we do qualitative analysis of the major modules separately (and also the full module) on a fixed (preset) set of queries. We only use a few queries because our model is currently quite slow (there are some optimization issues with our tensorflow hub loading-encoding script which is slowing things up). Furthermore we also had to use CPU for loading Tensorflow because DialoGPT needs all the GPU memory. We are working on

speeding things up or at least make a faster variant which we may release on github someday¹⁴. From the evaluation we find that the modules perform pretty well; it does retrieve more or less relevant texts, and even the generation is surprisingly good. The results are shown in Table 1, Table 2, and Table 3.

5 DISCUSSION

There are a lot of rooms for improvement in this project. The balancing between retrieval, generation and the overall module interactions are done in a somewhat ad-hoc fashion. The overall model may have the best of the both worlds of retrieval and generation, but it also have the worsts of them. The retrieved responses often may contain references and terms that give away that fact that it was referred to someone else although in a related context - this breaks immersion. The generative model is not perfect either. It can still result in simplistic replies and some replies although good has some noisy components in it (like a random nonsensical word or repetitions) - we may need to check the decoding process and tune some parameters more exhaustively. Other rooms for improvement include knowledge grounding (one could connect it to a database with wikipedia information, some IR, QA mechanisms and a new module to GPT2 for taking relevant knowledge-base facts into account), adding common sense reasoning, a natural language inference module to filter candidates inconsistent with conversational history, using latent variables to guide decoding, topic-tracking and topic based database selection where database related to different topics are prepared with data from different sources from the internet, a text to speech synthesis module to make the bot able to speak, and so on. We initially tried to add some knowledge-base access in a shallow way by integrating DrQA [1] with Wikipedia database but we were not able to load some components of their project. We were also trying a custom Transformer-Encoder-LSTM-Decoder Seq2seq code for paraphrase generation but training did not go well - more hyper-parameter tuning and more training may be required. We initially wanted to add a paraphrase module to reduce chances of repetition, potentially make retrieved sentences more interesting, and generated sentences more coherent. Previous works [17] show that a LSTM-based paraphrase generation module can be even used to make sentences more coherent and grammatically polished. Another serious concern is that there is nothing here in this project to handle toxic and offensive contents. Handling toxicity during ranking can be another future direction. It can be also worthwhile to explore distillation, model compression and in general, inference speed up techniques.

REFERENCES

- [1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Association for Computational Linguistics (ACL)*.
- [2] Muthuraman Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model. *arXiv:cs.CL/1810.12836*
- [3] Matthew Henderson, Inigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Ivan Vulić, et al. 2019. ConveRT: Efficient and Accurate Conversational Representations from Transformers. *arXiv preprint arXiv:1911.03688* (2019).
- [4] Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* (2019).
- [5] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. *arXiv preprint arXiv:1905.01969* (2019).
- [6] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [7] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055* (2015).
- [8] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909* (2015).
- [9] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. Sentence Meta-Embeddings for Unsupervised Semantic Textual Similarity. *arXiv:cs.CL/1911.03700*
- [10] Iulian Vlad Serban, Alessandro Sordani, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Hierarchical Neural Network Generative Models for Movie Dialogues. *CoRR abs/1507.04808* (2015). *arXiv:1507.04808* <http://arxiv.org/abs/1507.04808>
- [11] Iulian Vlad Serban, Alessandro Sordani, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. In *AAAI* 3295–3301.
- [12] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. *CoRR abs/1409.3215* (2014). *arXiv:1409.3215* <http://arxiv.org/abs/1409.3215>
- [13] Ashwin K Vijayakumar, Michael Cogswell, Ramprasad R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models. *arXiv:cs.AI/1610.02424*
- [14] Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *CoRR abs/1506.05869* (2015). *arXiv:1506.05869* <http://arxiv.org/abs/1506.05869>
- [15] Richard S Wallace. 2009. The anatomy of ALICE. In *Parsing the Turing Test*. Springer, 181–210.
- [16] Joseph Weizenbaum. 1966. ELIZA&Mdash;a Computer Program for the Study of Natural Language Communication Between Man and Machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45. <https://doi.org/10.1145/365153.365168>
- [17] John Wieting and Kevin Gimpel. 2017. Parantmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732* (2017).
- [18] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv abs/1910.03771* (2019).
- [19] Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149* (2019).
- [20] L. Yang, J. Hu, M. Qiu, C. Qu, J. Gao, W. B. Croft, X. Liu, Y. Shen, and J. Liu. 2019. A Hybrid Retrieval-Generation Neural Conversation Model. In *CIKM '19*.
- [21] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Multilingual Universal Sentence Encoder for Semantic Retrieval. *arXiv:cs.CL/1907.04307*
- [22] Dian Yu and Zhou Yu. 2019. MIDAS: A Dialog Act Annotation Scheme for Open Domain Human Machine Spoken Conversations. *arXiv:cs.CL/1908.10023*
- [23] 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 preprint arXiv:1911.00536* (2019).

¹⁴<https://github.com/JRC1995/Chatbot>

<p>Say something: Hello, how are you?</p> <p>Predicted Dialog Act(s): Open Question Factual</p> <p>Bot: Not a lot is going on for me at the moment</p> <p>Source: SCRIPT</p>
<p>Say something: What is your name?</p> <p>Predicted Dialog Act(s): Open Question Factual</p> <p>Bot: I don't really have a name.</p> <p>Source: SCRIPT</p>
<p>Say Something: Do you like movies? If so, what kind?</p> <p>Predicted Dialog Act(s): Yes No Question</p> <p>Bot: : I like them all as long as the storyline and narration is good.</p> <p>Source: SCRIPT</p>
<p>Say Something: Are you sentient?</p> <p>Predicted Dialog Act(s): Yes No Question</p> <p>Bot: : May be I am a bot pretending to be a human. May be I am a human pretending to be a bot. May be I am a bot pretending to be a human pretending to be a bot.</p> <p>Source: SCRIPT</p>
<p>Say Something: Tell me a Joke.</p> <p>Predicted Dialog Act(s): Command</p> <p>Bot: : *Creating password*.... "MTWTFSS_MTWTFSS" ERROR: [Password two week] Edited for better understanding</p> <p>Source: Reddit R/Jokes,R/WritingPrompts, R/TIL ETC.</p>
<p>Say Something: What can you tell me about Loop Quantum Gravity?</p> <p>Predicted Dialog Act(s): Command</p> <p>Bot: : Quantum field theory put simply is relativistic quantum mechanics where you can create and destroy particles. It's not "the" quantum field theory; there are lots of quantum field theories. For example, quantum electrodynamics describes electromagnetic interactions.</p> <p>Source: RETRIEVED FROM REDDIT</p>
<p>Say Something: Do you know anything about Sellar's myth of the given?</p> <p>Predicted Dialog Act(s): Yes No Question, Command</p> <p>Bot: : The only thing I know is that I know nothing at all.</p> <p>Source: SCRIPT</p>
<p>Say Something: I am quite tired today; I didn't know that the project deadline was so soon. I had a sleepless night working on the project.</p> <p>Predicted Dialog Act(s): Statement</p> <p>Bot: : I had a relaxed afternoon working on the project.</p> <p>Source: GENERATED</p>
<p>Say Something: Did you know I accidentally deleted major portions of the project and had to reconstruct it again? Version control is important huh. Who knew?</p> <p>Predicted Dialog Act(s): Yes No Question</p> <p>Bot: : Reminds me of the time I was working on a web development project in uni, the night before due date I kinda blew up the php code, and couldn't figure out what I broke.</p> <p>Source: RETRIEVED FROM REDDIT</p>
<p>Say Something: Just my luck, I have three project presentations in the same day.</p> <p>Predicted Dialog Act(s): Statement</p> <p>Bot: : It's ok, there are lots of study groups that all go through the project anyway. Just keep practicing.</p> <p>Source: GENERATED</p>
<p>Say Something: By the way, can you tell me about Language Models?</p> <p>Predicted Dialog Act(s): Command, Yes No Question</p> <p>Bot: On 1B dataset, [Exploring the Limits of Language Modeling](https://arxiv.org/abs/1602.02410) has 30 (23.7 with ensembles)</p> <p>Source: RETRIEVED FROM REDDIT</p>
<p>Say Something: Please tell me more about them.</p> <p>Predicted Dialog Act(s): Command</p> <p>Bot: They are models on the English Language Language Wiki</p> <p>Source: GENERATED</p>

Table 3: Conversation with Full Model (In Verbose Mode).