Comparing Statistical Models for Retrieval based Question-answering Dialogue: BERT vs Relevance Models

Debaditya Pal*
ABV - IIITM
Gwalior, Madhya Pradesh
India

Anton Leuski
Institute for Creative Technologies
University of Southern California
USA

David Traum
Institute for Creative Technologies
University of Southern California
USA

Abstract

In this paper, we compare the performance of four models in a retrieval based question answering dialogue task on two moderately sized corpora ($\sim 10,000$ utterances). One model is a statistical model and uses cross language relevance while the others are deep neural networks utilizing the BERT architecture along with different retrieval methods. The statistical model has previously outperformed LSTM based neural networks in a similar task whereas BERT has been proven to perform well on a variety of NLP tasks, achieving state-of-the-art results in many of them. Results show that the statistical cross language relevance model outperforms the BERT based architectures in learning question-answer mappings. BERT achieves better results by mapping new questions to existing questions.

1 Introduction

Given a question, the task of fetching relevant answers from a set of answers is referred to as retrieval based question answering. This method has been widely used to create dialogue agents (Lommatzsch and Katins 2019)(Yan et al. 2016)(Leuski and Traum 2011) as it allows for limiting the scope of the agent to relevant topics. This task is different from traditional question answering as the focus here is to create agents capable of carrying out fluid conversation with a user and portraying a consistent personality, instead of just fetching correct answers for factual questions. The nature of the problem makes it analogous to information retrieval, where a set of relevant documents must be fetched, given a user's query.

Recently deep neural networks, based on the transformer architecture (Vaswani et al. 2017), have had a huge impact in the field of information retrieval (Akkalyoncu Yilmaz et al. 2019)(Reimers and Gurevych 2019)(Jiang et al. 2020). These models have achieved state-of-the-art results in multiple tasks (Hoang, Bihorac, and Rouces 2019)(Zaheer et al. 2020)(Sun et al. 2019) including dialogue agent modelling when there is a large amount ($\sim 100,000$ utterances) of training data available (Wu et al. 2020). However, dialogue agents modelling singular personalities often deal with significantly less data. Transfer learning on parameter heavy transformer based models leads to problems like inconsistent agent personality (Chen et al. 2022) and overfitting . Thus statistical language modelling methods, although simplistic in nature, have been proven to perform extremely well on such tasks (Alavi, Leuski, and Traum 2020).

Through this paper, our primary contributions to the field are as follows:

- We compare a Cross Language Relevance model with three different deep neural networks based on the BERT architecture on two retrieval based question answering datasets of moderate size ($\sim 10,000$ utterances).
- We demonstrate the superiority of the relevance model over the state-of-the-art BERT models on the retrieval task where new questions are mapped to the answer set.
- We demonstrate the overall superiority of the state-of-the-art BERT models by following a retrieval method where new questions are mapped to the existing question set.
- We perform an error analysis and describe the type of errors made most often by the different models.

2 Related Work

The NPCEditor, introduced in (Leuski and Traum 2012), contains a statistical model for creating dialogue agents. The model has since been used to create and deploy various dialogue agents trained on datasets from different domains. One of the most famous domains is that of Pinchas Gutter (Traum et al. 2015b), a holocaust survivor, who recounts stories from his childhood when he was forced into concentration camps. This system has been exhibited at many museums over the past few years. We used data from this system for our current experiments (see Section 5.1) as well as the system itself for comparison with other models that were introduced much more recently.

The NPCEditor uses a Cross Language Relevance model. It requires a set of questions and answers with links established from the questions to relevant answers as a dataset to train a classifier. The entire algorithm behind the working of the NPCEditor has been described in detail in (Leuski and Traum 2012). We have given a sketch of the most important parts in section 3.1.

*This work was done during the first author's internship at the University of Southern California.

(Alavi, Leuski, and Traum 2020) previously showed a comparison between the NPCEditor and LSTM based deep neural networks on datasets varying from $\sim$ 10,000 utterances to $\sim$ 1 million utterances. The paper showed that even though the LSTM network performs better on the large scale dataset, its performance is dominated by the NPCEditor on the moderate sized ones. A few of NPCEditor's strengths that enable such performance are:

- It treats questions and answers as different 'languages', so the retrieval task is really a translation-retrieval task. Thus, it does not need to deal with the inherently different properties of questions and answers (discussed in Section 4).
- It is a simple model with very few training parameters (less than 10) thus, the classifier almost never overfits the data.

However, the NPCEditor still faces a few challenges:

- The answer models created by the NPCEditor are created over all the words observed in the answer vocabulary in the training set. However, during an interaction with a user, the user can use any words, some of which might not be in the training data. As a result, the classifier runs into a problem of vocabulary mismatch with the user.
- It uses words as the basic unit of computation. Each word is treated separately even if they can be used interchangeably in an utterance. Thus, it may not be able to retrieve queries with unseen or seldom seen words even if there are synonyms in the training data.
- Every utterance is stored and described as a frequency of words. This breaks down the sentence into a bag of words. Thus, the NPCEditor loses some of the positional information for the words conveyed by the structure of the sentence.

Noting these drawbacks and the fact that LSTM based neural networks have been outperformed by BERT based neural networks (Devlin et al. 2019) (Yamada et al. 2020), we decided to draw a comparison between the latter and the NPCEditor. BERT was proposed by (Devlin et al. 2019) and it was created by stacking the encoder layers of a transformer. Given an input utterance, BERT generates 768 dimensional dense embeddings. Recently, it achieved state-of-the-art results in multiple NLP tasks. The following points about BERT makes it the ideal candidate for this comparative study.

- BERT was pretrained on a huge corpus($\sim$ 13M) of English utterances. One of its pretraining objectives was masked language modelling, in which the model has to predict randomly blanked out words in an input utterance. This translates to a good knowledge of synonyms.
- Due to having such a large pretraining corpus, BERT also has a large vocabulary. Additionally, BERT tokenization occurs at the subword level. This means BERT can break down previously unseen words into known subwords and then calculate the embeddings. This reduces the chance of vocabulary mismatch with the user.
- BERT adds a positional encoding to the input embeddings before performing further computation. Thus, the model can utilize some of the positional information conveyed by the structure of the utterance itself.

BERT is a general language model which is trained to perform well in a variety of tasks. However, our task is specifically related to information retrieval. (Gururangan et al. 2020) showed that the continued pretraining of BERT helps in domain adaptation that translates to better performance in specific downstream tasks. SentenceBERT (Reimers and Gurevych 2019) is a library of different variations of BERT models. These models were continued to be pretrained on semantic searching and information retrieval tasks. These pretraining objectives align closely with the downstream task of retrieval based question answering. Hence, we have used pretrained models from this library, instead of the basic BERT model, that we further fine-tuned on the downstream task.

3 Models

We have tested four classification algorithms in our study. All of these algorithms essentially learn a function that can map a question to an answer, where the set of answers is fixed, but the set of questions is any sequence of English words. We also have a training set containing a set of matches between known questions and their relevant answers. However these algorithms are based on different methods and combinations. We tested one NPCEditor algorithm and three BERT algorithms, described in more detail in the rest of this section.

We examined two different retrieval methods using BERT. The first retrieval method we test, generates a similarity metric between input questions and the set of answers. Since, in a dialogue task, questions and answers often contain similar words, this retrieval method would help distinguish between relevant and non-relevant answers. Similar words spoken in the same context would also generate similar embeddings when passed through BERT. We looked at two algorithms using this method, Bi-encoder and Cross-encoder, as described in sections 3.2 and 3.3, respectively.

Our second retrieval method involves generating a similarity metric between the input question and the set of questions observed in the training set. The manually annotate relevant answer would then be fetched from the most similar question. This method is described in Section 3.4, and the motivation behind this approach has been described in detail in Section 4

3.1 NPCEditor

The NPCEditor uses Cross Language Relevance and works by generating similarity metrics between questions and answers. The first step in this process is to generate answer language models for all the answers in the answer set. The answer language models are probability distributions over all the words in the answer vocabulary. The NPCEditor uses Jelinek-Mercer smoothing to create these models.

Given a query, the next step is to generate an answer language model from that query. This is a probability distri-

bution of conditional probabilities over all the words in the answer vocabulary. Given a query $Q = q_1, q_2, ...q_n$, and the answer vocabulary $|A|$, the answer language model can be defined as:

$$P(a|Q) = \frac{P(a, q_1, q_2, ...q_n)}{P(q_1, q_2, ...q_n)} \qquad \forall a \in |A| \qquad (1)$$

Once the answer language model has been generated from the query, it is compared with the existing answer language models. The NPCEditor uses KL Divergence for this comparison and ranks the corresponding answers. The topmost answers are selected as relevant based on a threshold that is learned during training.

$$D(P_Q(A)||P(A)) = \sum_{a \in |A|} P(a|Q)\, log \frac{P(a|Q)}{P(a)} \qquad (2)$$

3.2 Bi-Encoder

Bi-Encoders are one of the most popular information retrieval models due to their capacity for high speed inference. The Bi-Encoder consists of a pair of encoders that individually map the question and answer to the embedding space. Once these embeddings have been generated, they can be checked for similarity using a metric like cosine similarity. We can pre-compute the embeddings for the answers. Thus, during inference we need to compute only one embedding for the question. Generating relevance scores is a simple matrix multiplication with the embeddings which does not have a considerable overhead.

However, the performance of the bi-encoder is limited. Since the model works on individual utterances, it has no knowledge of the question-answer pair, ie. when generating answer embeddings the model has no knowledge of the question and vice-versa. This is an important factor because the structure of an answer is dependent on the question.

3.3 Bi-Encoder + Cross Encoder

A cross encoder is much more of an end-to-end approach. For a question-answer pair, it computes relevance scores directly, instead of generating individual embeddings and then using a similarity metric. The cross encoder's architecture involves a BERT model with a classification head.

The model has been shown to perform better than bi-encoders in information retrieval tasks. This is expected since the model can utilize the knowledge of the question-answer pair, something that the bi-encoder cannot. However, the model is not without its challenges.

The biggest pitfall of the cross encoder is its slow inference and problems with scalability. Since the model takes question-answer pairs as input, every combination of the question and answers in the answer set must be passed into the model during inference. This leads to the computation of as many embeddings as there are answers. If the dataset contains a lot of answers, the model can potentially become too slow to carry out real time conversation with the user.

This problem can be mitigated by modifying the architecture of the retrieval methodology. We can use a Bi-Encoder to retrieve some good answers for the question and then simply re-rank these answers using the cross encoder. Thus, we limit the search space to a subset of the answer set.

3.4 Question Similarity

The previous BERT based approaches involve learning a mapping from questions embeddings to answer embeddings. However, comparing questions with answer has a few inherent problems. These problems have been discussed in detail in Section 4. NPCEditor avoids this trap by generating the framework of an answer from a question and then comparing this framework with the existing answers in the answer set.

With BERT, we could utilize the question embeddings to compare a previously unseen question directly with the other questions in the training set. Since the training questions are already linked to the relevant answers, we fetch the relevant answers for the similar question. However, one of the challenges faced by this approach is when the previously unseen question has no similar questions in the training set. In such a circumstance the model will still retrieve the most similar question, however, that question might be significantly different from the question asked by the user. Thus, the answers fetched will not be relevant.

4 The Questions v/s Answers Problem

The cross encoder and bi-encoder approaches rely on mapping questions with answers. The underlying assumption in this approach is that questions and answers will contain similar words within the same context. For Example:

Question: What is your father's name?
Relevant Answer: My father's name was Mendel.
Non-Relevant Answer: Um ... at the moment i am retired

However, questions and answers are inherently different types of utterances and have different salient properties.

- Questions have little to no factual information. However, they provide a framework for the answer. For eg: 'Where were you born?' provides no factual information but, since the word 'where' is present, the answer must contain information about a place.
- Answers on the other hand may be rich in factual information or they might be conversation fillers. But, they derive their structure from the question. This is why the question-answer pair knowledge is important while fetching relevant answers.
- Questions tend to contain a higher frequency of 'wh-' words like 'where', 'when', 'what', etc..

Due to these reasons, it is expected that the dense embeddings for questions and answers will not match along certain dimensions even if they are related to the same topic. Furthermore, Word level similarities does not guarantee relevance of answers and the context becomes important. For example:

Question: Pardon me, what did you say?
Relevant Answer: (Repeat last answer)

Model	Accuracy	Precision	Recall	F-Score
Question - Answer Mappings				
NPCEditor	90.75	85.35	53.87	61.32
Bi-Encoder	86.53	83.07	60.39	66.85
Bi-Encoder + Cross Encoder	88.14	79.17	58.60	61.08
Question - Question Mappings				
Question Similarity	93.57	93.30	58.90	67.42

Table 1: Results on the Pinchas-full dataset.

Model	Accuracy	Precision	Recall	F-Score
Question - Answer Mappings				
NPCEditor	72.50	66.05	34.41	42.48
Bi-Encoder	58.57	57.89	27.16	32.07
Bi-Encoder + Cross Encoder	68.00	56.81	38.39	40.60
Question - Question Mappings				
Question Similarity	74.25	75.16	32.64	42.17

Table 2: Results on the Pinchas-2015 dataset.

Non-Relevant Answer: I beg your pardon, what did you say?

This negatively impacts the performance of answer retrieval. The 'Question Similarity' retrieval methodology avoids this problem by comparing questions with other questions. Thus, the search space is limited to embeddings with similar features.

5 Experiments

5.1 Data Description

The two datasets used in this comparative study are the Pinchas-2015 (previously used in studies in (Traum et al. 2015a; Alavi, Leuski, and Traum 2020)) and the Pinchas-full dataset (an expanded dataset). The Pinchas-2015 dataset consists of 10,094 questions and 1,728 answers with 13,191 links between them. The corresponding test set consists of 400 questions.

The Pinchas-full dataset is a slightly larger dataset with 29,693 questions and 1,610 answers with 35,138 links between them. The dataset contains the question-answer pairs from the Pinchas-2015 dataset along with the various audience interactions of the previous Pinchas systems that were deployed. The corresponding test set consists of 995 new questions.

5.2 Preprocessing

The NPCEditor can directly utilize the links for training the classifier. However, the BERT models need question-answer pairs with associated relevance scores. Thus, we individually iterated over all the links and created question-answer pairs for each link with a relevance score of 1. But, we also had to introduce negative samples with relevance scores of 0 to balance the dataset and prevent it from becoming skewed. Thus, for every link, we paired up the question with a random answer from the dataset that it was not linked to, and assigned the pair a relevance score of 0.

5.3 Training overview

In total, 4 models were trained on identical data from the two datasets. As described in Section 3, these are the NPCEditor, Bi-Encoder, Bi-Encoder + Cross Encoder and Question Similarity models. For the NPCEditor, questions were represented using just the text whereas answers were represented using both the text and the answer ID. For the deep neural networks, both questions and answers were represented using just the text.

The deep neural networks were fine tuned for 6 epochs with learning rates of {1e-5, 2e-5, and 5e-5} with a linear learning rate scheduler with {100, 200 and 500} warm-up steps. The best set of hyperparameters were found to be a learning rate of 2e-5 with 200 warm-up steps.

6 Results

Tables 1 and 2 contain the metrics recorded for the experiments on the Pinchas-full and Pinchas-2015 datasets respectively. Our focus is primarily on the accuracy and precision metrics since these are of key interest in a dialogue system. The models retrieve a ranked list of answers and the highest ranked answer is used in the conversation. "Accuracy" measures how often the highest ranked answer is relevant and "Precision" measures how many answers are relevant in the set of retrieved answers. If the retrieval method involves mapping questions to answers, the NPCEditor significantly outperforms the BERT based models (Bi-Encoder and Bi-Encoder + Cross Encoder). Whereas the BERT based Question Similarity model outperforms all the other models by mapping new questions to existing ones thus showing the importance of the Questions v/s Answers problem.

7 Error Analysis

Table 3 contains information about the types of errors made by the different models on the Pinchas-full dataset along with table 4 which shows examples of the errors for better understanding. Since the models perform better on

Model	Changed Topic	Same Topic	Incorrect Short Response	No Answer	Total
Question - Answer Mappings					
NPCEditor	63	28	1	0	92
Bi-Encoder	57	58	17	2	134
Bi-Encoder + Cross Encoder	53	41	20	4	118
Question - Question Mappings					
Question Similarity	34	27	3	0	64

Table 3: Types of errors made by the different models.

Changed Topic Error
Question: What was your profession? Fetched Answer: I had a very happy childhood. Relevant Answer: I became, you could say, a credit manager in a finance company.
Question: Who are you? Fetched Answer: Its a wonderful feeling to live among friends and loving family Relevant Answer: Hi, my name is Pinchas.
Question: Would you please repeat your last answer? Fetched Answer: I have three children. Relevant Answer: (repeat last answer)
Same Topic Error
Question: What languages do you speak? Fetched Answer: Eight, I speak eight languages. Relevant Answer: I speak Yiddish, I speak Polish, I speak English, French ...
Question: What is your earliest recollection of Nazis? Fetched Answer: Life in the Ghetto was truly terrible ... Relevant Answer: Well, life changed to such an extent because after the Nazis occupied Lodz...
Question: When did you immigrate to the USA? Fetched Answer: I emigrated to Canada in 1985 Relevant Answer: I didn't. I didn't emigrate to the USA.
Incorrect Short Response
Question: Do you have an email? Fetched Answer: No Relevant Answer: Yes
Question: Do you believe Donald Trump is a good president? Fetched Answer: Yes Relevant Answer: I do not have an answer for that
Question: Can you say something in Portuguese? Fetched Answer: Okay Relevant Answer: Bom Dia!
No Answer
Question: Where were you born? Fetched Answer: I do not have an answer for that. Relevant Answer: I was born in Lodz, Poland.
Question: What is your favorite food? Fetched Answer: That is a topic for another place and time. Relevant Answer: My favorite food is Gefilte Fish ...
Question: When did you move to Canada? Fetched Answer: I do not have an answer for that. Relevant Answer: I emigrated to Canada in 1985.

Table 4: Examples of different types of errors.

the Pinchas-full dataset as compared to the Pinchas-2015 dataset, we have chosen the prior for the error taxonomy. In this section, we describe the different kinds of errors made by the models. Errors are counted only for the highest ranked response returned by the models as only this response is used for communication by the model.

7.1 Changed Topic

If an incorrect answer is in no way related to the general topic which forms the basis of the question, we classify it as a "Changed Topic" error. For example: "Hi, how are you?" is a salutation and asks about the Pinchas' well being. If the answer fetched for this question is "I was born in Lodz, Poland", then that is an irrelevant answer. This is because the answer is in no way related to the question and doesn't provide any insights about the speaker's well-being. From an encoder's perspective, this is usually the case when the query embeddings are vastly different from the relevant answer embeddings. This could either arise from mislabeled question-answer pairs during training or due to the Questions v/s Answers problem. The Question similarity model makes significantly fewer 'Changed Topic errors illustrating the role of the Questions v/s Answers problem.

7.2 Same Topic

If the answer fetched for a question is somehow related to the same general topic as the question, but fails to properly answer it, we classify the error into this category. The 'Bi-Encoder' and 'Bi-Encoder + Cross Encoder' models tend to make more errors in this category. This is expected as the method of retrieving answers depends on embedding similarity. Hence, even if an incorrect answer is fetched, it has a high chance of being related to the same topic. For example: "can you sing me a prayer?" is a specific question. If the answer fetched for this question is Pinchas singing a casual song, then we call it a same topic misunderstanding. Even though Pinchas is singing something, it does not qualify as a prayer.

7.3 Incorrect Short Response

These errors refer to one word answers like "yes" and "no" that perfectly answer a question grammatically but are factually incorrect. Such errors are observed much more with the 'Bi-Encoder' and 'Bi-Encoder + Cross Encoder' models as these models are trained to fetch utterances that correctly answer a question. However, it is difficult to capture factual knowledge with such models. For example: "Are you Polish?" is a question that asks about Pinchas' nationality. He was indeed born in Poland but if the answer fetched for this question was "No", then we classify it into this category. Modelling facts using Encoders requires proper domain adaptation which in turn requires large amounts of data. When using moderate sized corpora like these, parameter heavy encoders tend to make more such errors than relevance models.

7.4 No Answer

The datasets contain a few off-topic responses which are meant to be used when the models are fairly confident about not being able to answer the question asked. These responses usually ask the audience to change the topic of the conversation to something the agent is more familiar with or simply rephrase their question. In rare occasions we have observed the model fetching off-topic responses for questions that it has relevant answers for. Such errors are labelled as "No Answer" errors.

8 Conclusion and Future Work

Given a dataset of moderate size, Deep Neural networks fail to efficiently learn the mapping between questions and relevant answers. The performance of the 'Bi-Encoder' and 'Bi-Encoder + Cross Encoder' models were dominated by the NPCEditor, a statistical model based on cross language relevance, in terms of accuracy on both the datasets. However, if the questions v/s answers problem as described in Section 4 is dealt with by limiting the search space to similar utterances, the BERT based deep neural network can outperform the statistical model as shown by the performance of the 'Question Similarity' model.

Future work in this domain could explore methods of data augmentation like back translation. BERT and other transformer based architectures work better with large amounts of data, thus data augmentation could significantly boost performance. The BERT model itself has multiple variations pretrained on different corpora and objectives. A comparative study of the different versions of BERT could help identify the best pretrained model for the downstream task of retrieval based question answering. Both the deep neural networks and the NPCEditor have their own advantages, thus an ensemble model could potentially be worth looking into.

9 Acknowledgements

The first author would like to acknowledge the support received from the IUSSTF-Viterbi Program, organized by the Indo-U.S. Science & Technology Forum and the Viterbi School of Engineering at the University of Southern California. The program provided valuable resources and opportunities that were essential to the completion of this research. The first author is grateful for their support and looks forward to future collaborations. This work was supported in part by the Army Research Office under Cooperative Agreement Number W911NF-20-2-0053 and the National Science Foundation under Grant No. 1925576.

References

Akkalyoncu Yilmaz, Z.; Wang, S.; Yang, W.; Zhang, H.; and Lin, J. 2019. Applying BERT to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, 19–24. Hong Kong, China: Association for Computational Linguistics.

Alavi, S. H.; Leuski, A.; and Traum, D. 2020. Which model should we use for a real-world conversational dialogue system? a cross-language relevance model or a deep neural net? In *Proceedings of the 12th Language Resources and Evaluation Conference*, 735–742. Marseille, France: European Language Resources Association.

Chen, N.; Wang, Y.; Jiang, H.; Cai, D.; Chen, Z.; Wang, L.; and Li, J. 2022. What would harry say? building dialogue agents for characters in a story.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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), 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8342–8360. Online: Association for Computational Linguistics.

Hoang, M.; Bihorac, O. A.; and Rouces, J. 2019. Aspect-based sentiment analysis using BERT. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 187–196. Turku, Finland: Linköping University Electronic Press.

Jiang, Z.; El-Jaroudi, A.; Hartmann, W.; Karakos, D.; and Zhao, L. 2020. Cross-lingual information retrieval with BERT. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, 26–31. Marseille, France: European Language Resources Association.

Leuski, A., and Traum, D. 2011. Npceditor: Creating virtual human dialogue using information retrieval techniques. *AI Magazine* 32:42–56.

Leuski, A., and Traum, D. 2012. A statistical approach for text processing in virtual humans.

Lommatzsch, A., and Katins, J. 2019. An information retrieval-based approach for building intuitive chatbots for large knowledge bases. In *LWDA*.

Reimers, N., and Gurevych, I. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. Hong Kong, China: Association for Computational Linguistics.

Sun, C.; Qiu, X.; Xu, Y.; and Huang, X. 2019. How to fine-tune bert for text classification? In Sun, M.; Huang, X.; Ji, H.; Liu, Z.; and Liu, Y., eds., *Chinese Computational Linguistics*, 194–206. Cham: Springer International Publishing.

Traum, D.; Georgila, K.; Artstein, R.; and Leuski, A. 2015a. Evaluating spoken dialogue processing for time-offset interaction. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 199–208.

Traum, D.; Jones, A.; Hays, K.; Maio, H.; Alexander, O.; Artstein, R.; Debevec, P.; Gainer, A.; Georgila, K.; Haase, K.; et al. 2015b. New dimensions in testimony: Digitally preserving a holocaust survivor's interactive storytelling. In *International Conference on Interactive Digital Storytelling*, 269–281. Springer.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is all you need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wu, C.-S.; Hoi, S. C.; Socher, R.; and Xiong, C. 2020. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 917–929. Online: Association for Computational Linguistics.

Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; and Matsumoto, Y. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6442–6454. Online: Association for Computational Linguistics.

Yan, Z.; Duan, N.; Bao, J.; Chen, P.; Zhou, M.; Li, Z.; and Zhou, J. 2016. DocChat: An information retrieval approach for chatbot engines using unstructured documents. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 516–525. Berlin, Germany: Association for Computational Linguistics.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; and Ahmed, A. 2020. Big bird: Transformers for longer sequences. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 17283–17297. Curran Associates, Inc.