Neural-Augmented Retrieval for Open-Domain Dialogue Systems

Stanford CS224N Custom Project

Eric Frankel
Department of Statistics
Stanford University
ericsf@stanford.edu

Raphael Ruban
Department of Computer Science
Stanford University
ruban@stanford.edu

Rohan Mehrotra
Department of Computer Science
Stanford University
rohanm2@stanford.edu

Abstract

For open-domain neural dialogue agents to be effective conversationalists, they must be able to quickly handle a wide variety of topics that might be encountered in day-to-day conversations. ChirpyCardinal reconciles this challenge using a GloVe-based neural retrieval method and template-infilling scheme to allow discussion on Wikipedia-based knowledge with comparatively low latency. However, recent research has demonstrated that fine-tuned deep language models significantly outperform existing neural retrieval schemes, often at the expense of increased computational costs, through computing contextualized representations of queries and outputs. ColBERT improves on these methods by delaying the interaction between the query and output, decreasing query-response costs while retaining high retrieval quality. We aimed to improve retrieval-augmented generation in ChirpyCardinal in quality of retrieval and generation while minimizing increases in latency. We replaced ChirpyCardinal's GloVe-based retrieval method with ColBERT, experimented with using BART or T5 for contextualized template infilling, and evaluated these schemes through the relevance retrieval output and quality of the infilled template. We found that the use of ColBERT and finetuned BART allowed for the best end-to-end retrieval while not substantially increasingly latency. This suggests the application of large language models for neural generation can be used in real-time open-domain neural dialogue agents.

1 Introduction

In recent years, end-to-end deep neural dialogue agents have been able to sustain a rich conversation at an impressively high level for extended periods of time, raising hopes for their use in a wide variety of applications [1]. However, these dialogue agents are often limited by a lack of real-world knowledge and awareness of current world happenings, which is often what users are most interested in, and which are key parts of natural casual conversation. Moreover, dialogue agents also face additional interaction-based challenges, like an inability to plan for future conversation, and practical issues such as high latency, which decreases perceived conversation quality [2].

To address these shortcomings, Stanford's Chirpy Cardinal chatbot employs several parallel neural models to enable open-domain conversation, including a response module that allows for planning of future utterances as well as a retrieval-augmented generation (RAG) module that can mitigate the chatbot's lack of domain knowledge. This module is triggered upon the mentioning of an entity by the user – whereupon the module uses a GloVe-based neural retrieval method [3] to retrieve

information about the entity from Wikipedia, then uses this information for contextualized infilling of a template, and finally returns the infilled template as a response to the user. By enabling the chatbot to retrieve and incorporate real-world knowledge when making conversation, the RAG module has facilitated significant improvements in Chirpy Cardinal's user engagement and perceived conversational quality [4].

While Chirpy Cardinal relies on GloVe for neural retrieval, the use of fine-tuned large language models have demonstrated their effectiveness in modeling local interactions among query-document pairs rather than naive similarity between the representations of a given query and document. In particular, models based on ELMo and BERT have advanced performance on information retrieval benchmarks by computing deeply-contextualized semantic representations of query-document pairs. These pretrained language models (LMs) help bridge the pervasive vocabulary mismatch between documents and queries [5]. Unfortunately, this comes at the price of increased computational cost and latency associated with computing interactions between words within and across query-document pairs, which can impact user experience and makes efficient model deployment more difficult. Recently, the model ColBERT, which performs "late interaction" by retaining but delaying the query-document interaction seen in LM-based retrieval methods, mitigates these challenges by creating a neural retrieval scheme that retains the benefits of contextual interactions in large language models while allowing for the offline computation of document representations [6].

The high performance of ColBERT on retrieval coupled with its lower computational cost relative to other LM-based retrieval models makes it promising for use in real-time systems like Chirpy Cardinal and other dialogue agent chatbots. Inspired by this, we aimed to improve Chirpy Cardinal's Wikipedia-based RAG module by replacing its existing GloVe-based similarity search for neural retrieval with ColBERT. This constituted making ColBERT available through a REST API and rewiring notions of Wikipedia entries within the chatbot. After such rewiring was performed, we evaluated the quality of the Wikipedia knowledge statements, or "documents," retrieved from a corresponding template and entity, or the "query." We found that the statements retrieved by ColBERT were more relevant than those from GloVe-based retrieval, with statistical significance achieved after sufficient user testing. Additionally, we experimented with using BART [7] and T5 [8] for infilling a template with the best retrieved knowledge statement. We found that while there was no statistically significant difference between using BART or T5 for conditional generation for a given retrieval scheme, the quality of conditional generation significantly improved when paired with ColBERT-retrieved knowledge statements. Finally, we also profiled the latency of our different model permutations; GloVe-based retrieval was faster than that of ColBERT, though comparisons of their overall latency are difficult to make. Altogether, these results indicate that ColBERT should be integrated into ChirpyCardinal as the primary method of information retrieval.

2 Related Work

Deep Neural Dialogue Agents. Dialogue systems generally fall into one of two categories: task-oriented systems to solve domain-specific tasks or systems focusing on open-domain dialogue like ChirpyCardinal [2]. In a long line of research, neural models of dialogue generation for such open-domain conversations have shown great success in generating human-like responses. Chatbots like the one developed by Adiwardana et al. (2020) have even been able to engage in high quality conversations over multiple turns [1]. Yet, they are still limited by their knowledge of world events and latency in responding to user inputs. We hope to improve one such end-to-end deep neural dialogue agent, Stanford's Chirpy Cardinal.

Neural Retrieval. At the core of information retrieval are neural ranking models, which use shallow or deep neural networks to rank search results given a specific query [9]. Recently, models based on deep neural networks, which use rich embedding-level representations of queries and documents, have outperformed prior learning-to-rank methods that rely on hand-crafted features [6]; the best-performing models have been those that fine tune existing deep language models (LMs), like ELMo or BERT, for estimating the relevance of a given query to a corresponding document. These fine-tuned large LMs have significantly outperformed existing neural retrieval methods largely based on their ability to compute contextualized representations of query-document pairs, as well as bridging context-dependent differences in vocabulary between a given query and document [5].

These deep neural networks introduce a tension and tradeoff between better performance and more delayed response time. In the setting of a user-facing chatbot, however, latency is particularly critical for quick response times and a good user experience. So, we are turning to a ColBERT-based retrieval method using a `faiss` index to improve retrieval quality without introducing too much latency.

Conditional Generation. Before responding to the user, the neural ranking model's output needs to be incorporated into a meaningful, natural response. One way of doing so is filling in templates – pre-created sentence structures with placeholders – with the retrieved information. In other words, the task is to plug in slots for missing spans with text that is consistent with the preceding and subsequent text, which is known as text infilling [10]. Sequence to sequence models like BART [7] and docT5query [8] are models that are well-suited and can be adapted for this task. We explore the performance of both in this paper.

Retrieval and Generation in ChirpyCardinal. ChirpyCardinal aims to respond with interesting personal opinions or observations while still respecting user initiative. To do so, it fills templates with knowledge statements retreived from Wikipedia. More specifically, once Wikipedia entities are identified from dialogue, these entities and the templates to be filled are passed to a GloVe-based retrieval method. This GloVe search retrieves knowledge statements from an English Wikpedia dump of May 2020. Then, a neural infilling model inspired by Donahue et al. (2020) fills in the template. Specifically, a BART-base model trained on GPT3 generated and handwritten examples fills in the templates before they are reranked. This completes the response generation [4]. In this paper we will be looking at modifications to both the knowledge retrieval and infilling infrastructure, as previously discussed.

3 Approach

We break down our approach to our project into three parts: our approach to retrieval, infilling, and deployment into the existing ChirpyCardinal system.

3.1 Retrieval

One of the earliest methods in neural retrieval is embedding-based retrieval, which computes an embedding representation through a method like GloVe of both the query and the document before returning the query-document pairs with the highest similarity. These naive methods of information retrieval are particularly advantageous because they allow for offline computing of document representations, which is particularly useful in the context of ChirpyCardinal because of the debilitating effect of latency on user experience. Accordingly, we describe our GloVe-based approach as follows: for a given query q and knowledge statements – in this case, "documents" – $d^1, \ldots, d^n$, compute the embedding representations of both as $\vec{q}$ and $\vec{d^1}, \ldots, \vec{d^n}$. For a given similarity metric $\mathrm{sim}(\cdot, \cdot)$, we retrieve the best document to pass on to the infilling step by returning

$$d^* = \underset{i \in \{1,\ldots,n\}}{\arg\max}\ \mathrm{sim}(\vec{q}, \vec{d^i})$$

Alternatively, because of ColBERT's unique ability to balance representing contextual semantic information and lower computational costs than regular BERT, we use it in the following scheme: first, the documents $d^1, \ldots, d^n$ are encoded using ColBERT's document encoder, which is largely based on BERT. These encodings $\vec{d^1}, \ldots, \vec{d^n}$ are maintained in a `faiss` index, an off-the-shelf library for large-scale vector similarity search [11]. Next, the embedding of the query is computed using ColBERT's query encoder, again largely based on BERT. Therefore, for a given similarity metric $\mathrm{sim}(\cdot, \cdot)$, ColBERT retrieves the best document to pass on to the infilling step by returning

$$d^* = \underset{k \in \{1,\ldots,n\}}{\arg\max} \sum_{i \in [|\vec{q}|} \max_{j \in [|\vec{d^k}]} \mathrm{sim}(\vec{q_i}, \vec{d^k}_j)$$

3.2 Infilling

Neural language generation, even with contextual information, is difficult, and even more so when generated language is used to engage in conversation with a human. Accordingly, we rely on premade

templates[1] to enable generating conversation through conditional generation. We selected BART and T5 as models for conditional generation because of their effectiveness as sequence-to-sequence models in performing infilling. Based on the work established by [10], we consider an "infilling" problem the process of adding in tokens to an incomplete text $\tilde{x}$ and returning a completed text x. We do this by noting that it suffices to predict the missing spans y that replace blank tokens in $\tilde{x}$, framing infilling as learning $p(y|\tilde{x})$.

To learn this, we fine-tune pre-trained instantiations of BART and T5 thorugh the following method: using GPT-3 [12] to generate a sufficiently large dataset, we create a dataset of triplets $(\tilde{x}, e, s)$, where $\tilde{x}$ is a template with missing spans, e is the entity that should be used to infill this span, and s is a statement that should be conditioned on for infilling $\tilde{x}$. After fine-tuning BART and T5 on this dataset, they can then be used at inference-time for infilling similar premade templates used in ChirpyCardinal.

3.3 Deployment

We made ColBERT and the corresopnding neural infilling module accessible through a REST server deployed on the Stanford NLP cluster with sufficient GPU support. In the original GloVe-based module, requests to the infilling module largely consisted of three-sentence knowledge statements taken from an identified entity's Wikipedia page and templates ready for infilling. In our ColBERT-based retrieval model, however, only the templates consistent within a particular identified entity are passed to ColBERT, which performs its own search for the closest knowledge statements. The infilling model, which was either BART or T5, performed conditional generation on the templates upon successful retrieval of knowledge statements before returning the infilled templates in a JSON payload.

4 Experiments

4.1 Data

For all experiments we performed, we use a single Wikipedia dump as our data source. In this case, a May 2020 English Wikipedia dump was used, and only entities with at least 200 cross-references in Wikipedia were kept, resulting in 171, 961 entities in total; furthermore, too abstract (e.g. philosophy, film) or inappropriate entities were removed. However, the precise way this data was accessed by the neural retrieval model differed between ColBERT and GloVe-based retrieval:

- In GloVe-based retrieval, the neural model retrieves 3-sentence long knowledge statements from the main body of the identified entity's Wikipedia page.
- In ColBERT, an existing `faiss` index of 180-token passages of the same Wikipedia articles was used, wich resulted in a total of approximately 21 million knowledge statements. The creation of this `faiss` index was made possible through the existing ColBERT codebase.

We perform an ablation study to assess the impact of these differences in input data; see section 4.4.1.

Our infilling models, BART and T5, were fine-tuned on synthetic data generated by GPT-3 for infilling as described in our approach. The data used for our infilling model consisted of the retrieved knowledge statements from Wikipedia and a template from the aforementioned list.

4.2 Evaluation method

4.2.1 Retrieval Metrics

While traditional information retrieval and infilling models have concrete metrics, such as MRR, there are no concrete "correct" responses for retrieval for a given knowledge statement. Indeed, given that the purpose of these knowledge statements is to produce topical, interesting utterances, defining what constitutes sufficiently "interesting" retrieval is subjective and requires sufficiently large human evaluation. Accordingly, we created 3 cohorts of 20 retrieval templates for different entities ranging from food, music, and geography to current events; we randomly selected which cohort would be

[1] Templates can be found here.

Table 1: Retrieval Relevance. ColBERT achieves statistical significance in outperforming both GloVe and Aug-GloVe in average relevance score. Confidence intervals are two-tailed with $\alpha = 0.05$.

Retrieval Method	**ASR@5**	**ASR@7**	**ASR@10**
GloVe	2.35 ± 0.11	2.94 ± 0.32	3.8 ± 0.66
Aug-GloVe	2.56 ± 0.32	3.08 ± 0.41	3.91 ± 0.58
ColBERT	$\mathbf{2.91 \pm 0.20}$	$\mathbf{3.98 \pm 0.29}$	$\mathbf{5.45 \pm 0.73}$

used for a given human evaluator. Next, for each of the retrieval-entity queries, the top-k knowledge statements were retrieved and evaluated in relation to the query. For each query, each top-k retrieval receives a score $s_i \in \{0, \ldots, k\}$ for the number of relevant knowledge statements it contains. Thus, we calculate the **average relevance score at** k (ASR@k) determined by the human evaluator as

$$\text{ARS@}k = \frac{1}{n} \sum_{i=1}^{n} s_i$$

where $n = 20$ is the number of queries in a given cohort. ASR@k thereby gives a notion of the number of relevant documents for a given retrieval model.

However, ASR@k aggregates this relevance across retrieved documents – that is, it does not account for the ranking of the perceived "best" knowledge statement retrieved. Accordingly, we introduce another metric called **adapted mean reciprocal rank at** k (aMRR@k), which is

$$aMRR@k = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{rank_i}$$

where $rank_i$ refers to the rank position of the most relevant document for the i-th query.

4.2.2 Infilling Metrics

Similarly, there are no natural quantitative infilling metrics that can be used without the integration of human evaluators. Accordingly, a similar cohorting scheme to that of evaluating our retrieval models was used. After top k retrieval for a given template was performed, our infilling model performed conditional generation on the template for each retrieved document d_i for $i \in \{1, \ldots, k\}$. For each infilled statement, it was marked as either sensible or not $s_i \in \{0, 1\}$, with the **average sensibility score** being calculated as

$$ASS = \frac{1}{n} \sum_{i=1}^{n} s_i$$

4.2.3 Latency Evaluation

Finally, we profiled the latency of the retrieval process using Python's `time` [13] package over the course of the retrieval and infilling process. We report the average time required for each step in the corresponding processes. Note that these average times are hardware dependent and might change when run on a different system.

4.3 Experimental details

First, a ColBERT model was trained on the MSMARCO dataset [14] for 400,000 iterations to convergence; this pretrained model was then used for embedding our Wikipedia knowledge statements as documents. This model was trained on the Stanford NLP cluster with 2 Nvidia GeForce RTX 3090 GPUs and 100GB of memory. The `faiss` index used in our ColBERT retrieval was created with a maximum document length of 180 tokens and with punctuation masked, a batch size of 256, and a chunksize of 8 using document tokenizer from the pretrained ColBERT model, which itself relied on a pretrained BERT tokenizer; the same hardware specifications were used for creating the `faiss` index as training the retrieval model. For GloVe retrieval, the English Core Web Large pipeline from `SpaCy` [15] was used for token embeddings to create query- and document-level embeddings.

Table 2: Retrieval Response Ranking. ColBERT significantly outperforms both GloVe and Aug-GloVe in adapted mean reciprocal rank.

Retrieval Method	aMRR@5	aMRR@7	aMRR@10
GloVe	0.301	0.285	0.240
Aug-GloVe	0.322	0.303	0.255
ColBERT	**0.532**	**0.498**	**0.433**

Next, pretrained neural infilling models BART and T5, with weights taken from HuggingFace, were finetuned in the manner described in our approach on a dataset of size 4284. These models were trained using the AdamW optimizer with learning rate 1e-5 for four epochs.

The Stanford NLP cluster hosted the server used for managing access to the retrieval and infilling models; this server was allocated 100GB of memory and 2 Nvidia GeForce RTX 3090 GPUs.

4.4 Results

4.4.1 Retrieval

We report the performance of the two neural retrieval models, GloVe-based search and ColBERT, in average relevance score in Table 1. ColBERT significantly outperforms GloVe-based search in all ARS@k that was evaluated, indicating its superior ability to effectively retrieve knowledge statements that are deemed pertinent to the query. First, note that in both models, the number of relevant retrieved statements did not scale linearly with k; this dropoff indicates that in both models, increasing k will not necessarily lead to more relevant or higher quality retrievals. However, note that the dropoff in ColBERT performance is significantly less than that of GloVe, indicating that for ColBERT, at least, there might be a benefit in increasing k up to a particular threshold; further experimentation is required to determine such a threshold in conjunction with balancing potential increases in latency.

We also report the performance of GloVe-based search and ColBERT on adapted mean reciprocal rank in Table 2, which reflects ColBERT's superior ability to rank the most relevant documents higher compared to GloVe-based search. As k increases the aMRR in both models decreases, which reflects changing evaluator perception on the best possible knowledge statement when exposed to more retrieved documents; Again, ColBERT's dropoff in performance is significantly less than that of of GloVe-based search. These results on aMRR are revealing: ColBERT's best retrieval is on average in approximately rank 2, while GloVe's best retrieval is on average between rank 3 to 4. Accordingly, during use in ChirpyCardinal, ColBERT will more consistently provide higher quality knowledge statements as context to the infiller model relative to GloVe-based search.

Aug-Glove Ablation. To ensure that differences in model performance were not the result of differences in the data provided to the two models, we performed an ablation study where GloVe performed similarity search using the 180-token knowledge statements stored in ColBERT's `faiss` index; since these knowledge statements were not stored in data structure that associated each statement with a particular entity, we used ColBERT to retrieve the top 100 knowledge statements corresponding to the given query to simulate providing access to many potential knowledge statements while also creating the necessary data overlap between the two models. The GloVe-based search on the new knowledge statements, denoted Aug-GloVe, achieved a statistically insignificant improvement in performance relative to GloVe on both ARS and aMRR as seen in Tables 1 and 2. These results indicate that ColBERT's superior performance, as suspected, is due to its superior ability to represent semantic relationships between query-document pairs rather than something inherent to the data it has access to.

4.4.2 Infilling

We report the performance of the permutations of the two retrieval models and two infilling models on ASS in Table 3. First, our results demonstrate the the combination of ColBERT and BART outperform the other possible model combinations for infilling; this is promising for use in ChirpyCardinal. However, more revealing is the large difference in performance of the retrieval methods irregardless of whether BART or T5 is being used. The infilling statements created by the conditional generation

Table 3: Retrieval Response Ranking. ColBERT significantly outperforms both GloVe and Aug-GloVe in adapted mean reciprocal rank.

Retrieval Method	GloVe		ColBERT	
	BART	T5	BART	T5
ASS	0.484 ± 0.03	0.446 ± 0.02	$\mathbf{0.642 \pm 0.04}$	0.590 ± 0.05

models based on context from GloVe were significantly less sensible than those from ColBERT, indicating that the context provided to the infilling models significantly affects perceived sensibility of the infilled templates.

4.4.3 Latency

In Table 5 of the appendix, we report our measurements of the latency for each component of ChirpyCardinal. We find that ColBERT's retrieval process is slower than that of GloVe-based search, with there being no significant difference in infilling time compared between T5 and BART when used in conjunction with either of the two retrieval models. However, the process of retrieval differs between the two methods in fundamental ways: in GloVe-based search, the initial knowledge statements are already retrieved before being passed to ColBERT, whereas ColBERT retrieves the optimal indices of knowledge statements similar to the query, which are then used to look up the actual text of the statements. Accordingly, this imbalance in procedure lies at the heart of the differences in latency; empirically, the latency of ColBERT does not affect the user experience of interacting with the chatbot.

5 Analysis

In Table 4 are example inputs and outputs to a given retrieval or infilling model, comparing and constrasting the performance of GloVe-based retrieval versus ColBERT in conjunction with a BART model for infilling.

The difference in infilled statements demonstrates the challenge of implementing such an end-to-end infilling system as well as limitations in GloVe-based search. Note that in isolation, the results of retrieval and infilling are somewhat logical given the inputs: the retrieval step returns a knowledge statement that provides a statement that alludes to physical locations, while the infilling step effectively uses the knowledge statement to fill in the blanks of the template based on the mentioned locations. However, the output of the infilling process is a statement that is factually questionable because of the knowledge statement not matching the semantics of the query. In contrast, ColBERT's best knowledge statement closely matches the semantic information of the query, making the infilling output factual and semantically viable. This high level pattern, with ColBERT returning better knowledge statements leading to better infilled ouputs, holds across other conversation topics in the chatbot and proves promising for ColBERT's use in ChirpyCardinal more generally.

6 Conclusion

Despite advances in large language models and natural language understanding, open-domain chatbots lack knowledge that would allow them to maintain rich and topical conversations. While ChirpyCardinal addresses these concerns through the use of a Wikipedia-based response generator, its use of GloVe for information retrieval leaves much to be improved. In this project, we further advance the ability of ChirpyCardinal to maintain rich conversation by replacing GloVe-based similarity search with ColBERT and experimenting between two different neural infilling models, BART and T5. We found that ColBERT returned significantly more relevant knowledge statements more frequently than GloVe, which led to better infilling performance by BART and T5. This validated our hypothesis that a powerful language model could be integrated into a real-time chatbot with minimal effect on user experience through latency. However, this work was not without limitations, the main one being the reliance on human evaluation for quantitative metrics and the subjectivity that naturally accompanies such evaluation. Another limitation was the lack of concrete comparisons between the latency of using ColBERT versus that of GloVe, as the process of creating the requisite input data

Table 4: Retrieval and infilling example with given template: "Koala primarily lives in [place]"

	GloVe	**ColBERT**
Top-k Knowledge Statements	naturalist and popular artist John Gould illustrated and described the koala in his three-volume work The Mammals of Australia (1845–63) and introduced the species, as well as other members of Australia's little-known faunal community, to the general British public Comparative anatomist Richard Owen, in a series of publications on the physiology and anatomy of Australian mammals, presented a paper on the anatomy of the koala to the Zoological Society of London In this widely cited publication, he provided the first careful description of its internal anatomy, and noted its general structural similarity to the wombat ...	The koala or inaccurately koala bear (Phascolarctos cinereus) is an arboreal herbivorous marsupial native to Australia It is the only extant representative of the family Phascolarctidae and its closest living relatives are the wombats which are members of the family Vombatidae The koala is found in coastal areas of the mainlands eastern and southern regions inhabiting Queensland New South Wales Victoria and South Australia It is easily recognisable by its stout tailless body and large head with round fluffy ears and large spoon-shaped nose The koala has a body length of and weighs Fur colour ranges from silver grey to chocolate brown Koalas from the northern populations are typically smaller and lighter in colour than their counterparts further south These populations possibly are separate subspecies but this is disputed Koalas typically inhabit open eucalypt woodlands and the leaves of these trees make up most of their diet Because this eucalypt diet has limited nutritional and caloric content koalas are largely sedentary and sleep up to 20 hours a day ...
Best Knowledge Statement	Naturalist and popular artist John Gould illustrated and described the koala in his three-volume work The Mammals ...	The koala or inaccurately koala bear (Phascolarctos cinereus) is an arboreal herbivorous marsupial native to Australia It is ...
Infilled Statement	Koala primarily lives in Australia and Britain.	Koala primarily lives in southern Australia.

occurred in different parts of the code. Future work could include improving latency of the infilling step conditional generation model by eliminating the expensive and likely unnecessary copying of words between slots to be infilled during decoding in BART/T5, as well as further code optimization that could further decrease latency. Improvements to ChirpyCardinal might also include end-to-end evaluation of the neural-augmented generation pipeline and the creation of additional templates for infilling.

References

[1] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[2] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

[3] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[4] Ethan A. Chi, Chetanya Rastogi, Alexander Iyabor, Hari Sowrirajan, Avanika Narayan, and Ashwin Paranjape. Neural, neural everywhere: Controlled generation meets scaffolded, structured dialogue. 2021.

[5] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu Liu. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*, 2019.

[6] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[8] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. From doc2query to doctttttquery. *Online preprint*, 6, 2019.

[9] Bhaskar Mitra and Nick Craswell. An introduction to neural information retrieval. *Found. Trends Inf. Retr.*, 13:1–126, 2018.

[10] Chris Donahue, Mina Lee, and Percy Liang. Enabling language models to fill in the blanks. *arXiv preprint arXiv:2005.05339*, 2020.

[11] Jeff Johnson Hervé Jegou, Matthijs Douze. Faiss: A library for efficient similarity search, 2017.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[13] Vincent Driessen. times - PyPI (version 0.7), 2014.

[14] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[15] Explosion. en_core_web_lg - spaCy (version 3.2), 2016.

A Key Information to include

- Mentor: Ethan Chi
- External Collaborators (if you have any): None
- Sharing project: No

B Appendix

B.1 Latency

Here we report the outcome of our profiling of the latency of the two methods. We find the following times:

Table 5: Average Latency Times

Neural Method	**GloVe**	**ColBERT**	**BART**	**T5**
Latency (s)	~ 0.02	~ 0.65	~ 1.34	~ 1.28