Knowledge-Enhanced Language Models: A Comparative Study of RAG and Embedding Methods

Stanford CS224N Custom Project

Adarsh Ambati
Department of Mathematics
Stanford University
adarsh1@stanford.edu

Nikash Chhadia
Department of Computer Science
Stanford University
chhadia@stanford.edu

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in various NLP tasks. However, they are still somewhat limited in terms of accessing and logically employing precise knowledge, and are hence prone to hallucination. The performance of LLMs in this regard can be improved by incorporating knowledge from external sources. As methods for doing so are often focused on individually and in isolation, our project aims to explore and compare various methods to achieve this enhancement. Specifically, we investigate Retrieval-Augmented Generation (RAG) with constructed knowledge graphs on a domain-adapted BERT base model. As an alternative method, we additionally explore the KnowBERT architecture which encodes knowledge directly into a language model. Our baseline is a domain-adapted BERT large model without any knowledge graph assistance. Evaluation metrics we employ include accuracy, precision, recall, F1 score, perplexity, and a factual recall test. We find that in general injecting knowledge into large language models improves accuracy. Between the two frameworks we examined, KnowBERT outperformed RAG-BERT on all metrics except absolute accuracy in the factual recall test. Thus, we show that enhancing language models by embedding knowledge can be a viable solution to the general problem of hallucination. Future work involves combining various architectural choices from RAG, KnowBERT, and other methods for further model improvement and accuracy.

1 Key Information to include

- Mentor: Aditya Agrawal
- External Collaborators: N/A
- Sharing project: N/A
- Contributions: The authors chose to go alphabetical with author ordering. The two team members contributed equally to the final project. Adarsh contributed the implementation of the RAG method and KnowBERT methods and performed the evaluations for the models. Nikash contributed the extraction of the Wiki dataset and the evaluation of the baseline methods. Both members contributed equally to the final report.

2 Introduction

Large Language Models (LLMs) have revolutionized the field of natural language processing (NLP), especially with the addition of transformer architectures (Vaswani et al., 2017) in recent years, such as with BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). These models excel in understanding and generating human-like text, which has opened up applications in numerous areas including machine translation, question answering, and sentiment analysis. However, despite their significant advancements, LLMs face challenges when it comes to accessing and employing precise, factual knowledge. This limitation often leads to the notion of “hallucination,” where models generate plausible-sounding yet factually incorrect or illogical information.

In particular, hallucination arises from the inherent design of LLMs, which relies on patterns learned from large text corpora, where knowledge is often long-tailed and not always easy to recall upon inference. These models can mimic language fluently, but do not have a natural mechanism to verify the factual accuracy of the information they generate (Kandpal et al., 2023). This is an important issue, as the generation of false or misleading information can potentially have serious consequences in applications to fields such as healthcare, law, and education. Addressing this problem is hence essential to ensure the reliability and trustworthiness of these AI systems.

Currently, methods to mitigate hallucination in LLMs can be classified into two categories: prompt engineering and developing novel models (Tonmoy et al., 2024). The former, prompt engineering, involves structuring the prompt to an LLM in particular ways to obtain a more desirable output. It can help provide necessary context or specific instructions to guide a model to the expected answer without hallucination. The latter, developing novel models, simply entails the creation of new model architectures and data representations to improve outputs and reduce hallucinations. Some examples may involve modifying a model’s decoding process to guide generation more toward authentic context, or altering a model’s loss function to better incorporate how close outputs are to ground truth.

While prompt engineering and model development methods have been explored to address hallucination, there still exists some gap in directly comparing and measuring their effectiveness. Existing research often focuses on one method in isolation, making it difficult to determine which approach yields the most significant reductions in hallucination and under what specific circumstances. This is where our project aims to contribute. Specifically, we investigate the popular prompt engineering method of Retrieval-Augmented Generation (RAG), where factual information is retrieved from Knowledge Graphs (KGs) and incorporated into the query to guide the LLM towards a more accurate response. We will also explore a novel model development method, KnowBERT, designed specifically to enhance factual language understanding in LLMs at the embedding layer.

By directly comparing these various approaches alongside a baseline absent of any knowledge enhancement methods, our project aims to shed light on which methods are most effective at mitigating hallucination in LLMs and under what conditions. With this objective, we aim to provide valuable insights for researchers and developers working to improve the reliability and trustworthiness of LLMs.

As for our results, we found that these knowledge-injected large language models can serve as viable solutions to the growing problem of hallucinations. Both methods (prompt engineering and novel architectures) at a high level demonstrated marked improvements in terms of accuracy and semantic understanding compared to the baseline models. Between the two, we would say that the novel model architecture that we implemented, KnowBERT, was more successful that the prompt-engineering representative, Retrieval-Augmented Generation.

3 Related Work

The landscape of research in knowledge enhancement for language models is diverse and evolving, with multiple approaches contributing to the development and improvement of natural language understanding and generation. This section reviews key works that serve as a basis for our work.

BERT. Bidirectional Encoder Representations from Transformers, or BERT, is a language model introduced in 2018 by Google AI research (Devlin et al., 2019), popularized for its significant improvement over previous SOTA language models. BERT uses a transformer architecture to understand the context of words in a sentence by looking at both the left and right context, unlike traditional models that read text sequentially. It employs a two-step training process: unsupervised pre-training on a large text corpus to learn general language representations, followed by supervised fine-tuning on specific tasks, making it highly adaptable. BERT-large, in particular, is renowned for its deep, 24-layer architecture that provides a rich understanding of language nuances. We’ve selected BERT-large as our baseline, as its robust contextual embedding capabilities have set SOTA performance benchmarks across a wide range of NLP tasks.

Retrieval-Augmented Generation. One of the most prominent approaches to enhance the knowledge capability of language models is the integration of external knowledge sources, such as Knowledge Graphs (KGs). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) is the most notable method in this domain, as it combines the strengths of retrieval-based and generative models, enabling the model to retrieve relevant documents from a large corpus and generate coherent responses using the retrieved information. Typically, when a query is made to a language model, a RAG system will first search for and retrieve relevant information from a large dataset or knowledge base, and then use that information information to prompt engineer the original query and guide the generation of the model's response. This process is displayed in Figure 1. The use of KGs instead of simpler vector databases for RAG further enriches the contextual understanding by providing structured and factual information.

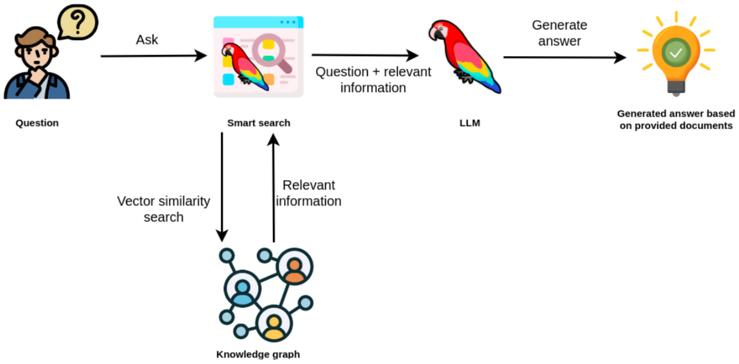

Figure 1: Retrieval-Augmented Generation (RAG)

KnowBERT. The knowledge enhanced BERT model (Peters et al., 2019), or KnowBERT, is an extension of BERT that, unlike RAG, integrates structured knowledge directly into the model, enhancing its ability to handle tasks requiring entity-level understanding. It uses the large-scale knowledge bases Wikipedia to gather this structured knowledge, and involves two-step process: first, it links entities to their corresponding entities in the knowledge bases to retrieve relevant entity embeddings, and second, it integrates the entity embeddings with the BERT architecture using a word-to-entity attention mechanism, as displayed in Figure 2. This integration allows KnowBERT to leverage structured knowledge, improving its performance on tasks involving named entities and factual information. Evaluating KnowBERT in our comparison is hence beneficial because it exemplifies how embedding external knowledge directly into language models can enhance understanding and mitigate issues like hallucination. As a further benefit, the outer layers of KnowBERT and BERT are the same, so KnowBERT can serve as a clean replacement for BERT in most BERT-based models. This is what allows us to train KnowBERT in both the masked language learning and the next sentence prediction models.

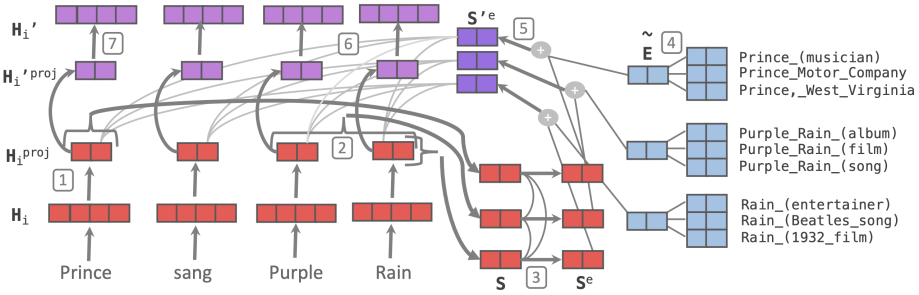

Figure 2: Visualization of the KnowBERT architecture's recontextualization of words with a word-to-entity-span attention mechanism.

By drawing on the strengths and addressing the limitations of existing methods such as RAG and KnowBERT, and establishing a solid baseline with BERT-large, our work aims to illuminate and advance the understanding and application of knowledge-enhanced language models. The systematic comparison of hallucination mitigation methods and the practical re-implementation of Knowledge Graph-based RAG are contributions that we believe will drive and encourage further innovations in this significant area.

4 Methodology

4.1 RAG Approach

Our first approach involves implementing RAG with KGs, for which we've coded our own RAG system. We initially planned on using existing KGs, but part of our considerations was the efficiency of the model. Thus, we determined that most existing KGs are either too large to make a tractable query or do not have useful API for the model to use quickly, and decided to construct our own KG. This KG construction process is outlined as follows:

1. **Data Collection:** We scrape data from a curated set of scientific Wikipedia articles using the Wikipedia Python library.
2. **Entity Extraction:** Using Spacy's ''en_core_web_sm'' tool, we extract subjects, verbs, and objects from the collected Wikipedia data.
3. **Graph Database:** The extracted entities and relationships are imported into NetworkX's DiGraph libraries (Hagberg et al.). We utilize NetworkX's capabilities to manage and store the graph data efficiently. The Knowledge Graph is then saved in a JSON file format to be later utilized by the model.

Our constructed KG using NetworkX and the top science Wikipedia articles is shown in Figure 3.

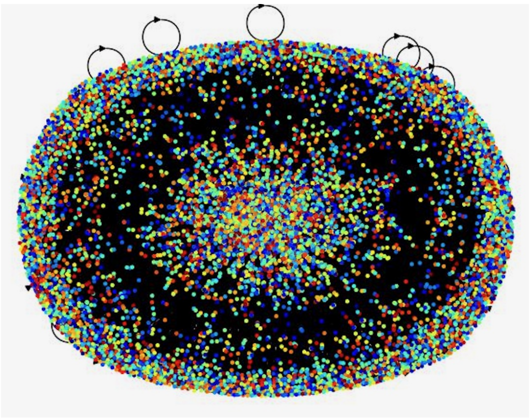

Figure 3: Visualization of Constructed Knowledge Graph

Then, for given user prompts, we implement RAG as follows:

1. **Tokenization:** The prompt is tokenized to extract keywords.
2. **Subject-Verb-Object (SVO) Triplets:** Using Spacy's NLP, the primary subject is extracted from the prompt. An exhaustive search of the knowledge graphs links is performed creating a number of SVOs.
3. **Response Generation:** The SVOs are appended to the original prompt as context, which is then fed into a language model to generate an answer. This takes advantage of BERT's bidirectional attention and transformer architecture to comprehend and apply simple appends to the query.

For the language model, we use the BERT-base-uncased model, sourced from the Hugging Face transformers library, to generate responses. The process integrates the structured knowledge from the graph into the prompt, enhancing the model's response accuracy.

4.2 Knowledge Embedding Approach

In addition to RAG, we incorporate the KnowBERT architecture as an alternative method that instead directly embeds structured knowledge into BERT. We utilized existing code to load the model (Peters et al., 2019) but wrote our own code using PyTorch (Paszke et al., 2019) to finetune and evaluate it with our Wikipedia data. As for the model construction, after extracting the text from the Wikipedia articles, we formatted the text into files where each line contained two full sentences either randomly paired together or two sentences with one following the other. This created the setup for pretaining KnowBERT for the next sentence prediction problem described later. (Peters et al., 2019)'s implementation contained a number of helpful scripts to accomplish this task. Using the AIDA-CoNLL dataset (Hoffart et al., 2011) as instructed by the KnowBERT authors, we pretrained the entity linkers. Finally, using our supervised Wikipedia Data and the now-trained entity linkers, we fine-tuned BERT into KnowBERT, specifically for answering science-based questions.

4.3 Baseline

Our baseline for comparison is a BERT-large model that does not utilize any knowledge graph assistance, but uses domain adaptation on our data. We further pretrain this model on the Wikipedia articles we pulled previously to construct knowledge graphs, and evaluate using our score metrics. This baseline serves to establish a performance benchmark, allowing us to evaluate whether the smaller BERT-base model, augmented with knowledge graphs through RAG or KnowBERT, can outperform the larger BERT model. This comparison is crucial to determine the effectiveness and efficiency of integrating knowledge graphs into smaller models, potentially offering a more resource-efficient solution without compromising performance.

5 Experiments

5.1 Data

Our primary data consists of a collection of Wikipedia articles, which were the top 85 results from the natural science and physical science subcategories. We use these articles to generate our RAG knowledge graph, as well as to evaluate our models with BERT next sentence prediction and a specially designed factual recall test. A selection of some of these articles are displayed in Table 1.

"Astronomy"	"Glossary of astronomy"	"Outline of astronomy"
"Portal:Astronomy"	"Advanced Scientific Data Format"	"Alignments of random points"
"Aperture Photometry Tool"	"Astroinformatics"	"Astrology and astronomy"
"Astrology and science"	"Astronomer"	"Astronomical coordinate systems"
"Astronomy and spirituality"	"Astrophysical fluid dynamics"	"Baade-Wesselink method"
"Barcelona astrolabe"	"Blanketing effect"	"Burning plasma"
"Coincidence method"	"Constellation"	"Cosmic wind"

Table 1: Assortment of some of the science Wikipedia articles used in our dataset.

5.2 Evaluation Method

Next Sentence Prediction Test: To comprehensively evaluate the actual knowledge comprehension performance of our models, we employ accuracy, precision, recall, and F1 score metrics using the Scikit-learn (Pedregosa et al., 2018) implementation. With each of our models, we use our evaluation data and compute these metrics for the next sentence prediction task, where a model predicts the sentence most likely to follow the given one. In addition, we compute model perplexity, which measures how well or how confidently a language model makes its predictions by quantifying the average uncertainty or surprise of the model in predicting the next sentence. The metrics we employ are defined as follows:

$$\textit{Accuracy} = \frac{\textit{true positives + true negatives}}{\textit{true positives + true negatives + false positives + false negatives}}$$

$$\textit{Precision} = \frac{\textit{true positives}}{\textit{true positives + false positives}} \qquad \textit{Recall} = \frac{\textit{true positives}}{\textit{true positives + false negatives}}$$

$$\textit{F1} = 2 \times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision + Recall}} \qquad \textit{Perplexity} = \exp\left(-\frac{1}{t}\sum_{i}^{t} \log p_\theta(x_i|x_{<i})\right)$$

Factual Recall Test: In addition to the next sentence prediction evaluation metrics. We have specially designed a factual recall test to compare the various model performances at parsing and relaying ground truths. This test takes advantage of the fact that both models are BERT-based; BERT is fundamentally a Masked Language Modelling. The factual recall test is as follows:

1. Using GPT-4, we generated 100 fill-in-the-blank questions and answers based on our 30 Wikipedia articles
2. Models were asked to fill in the **[MASK]** with their predictions, but most importantly the probability they assigned to each word in their entire vocabulary was examined.
3. The rank in which the ground-truth appears in the probability-vocab vector is used to calculate the following metrics.
4. We evaluate absolute accuracy, mean reciprocal rank, and average cosine similarity.

Absolute Accuracy measures the percentage of queries where the ground truth word is exactly returned. Mean Reciprocal Rank (MRR) measures the average of the reciprocal ranks of the ground truth word in the probability distribution for each query. Average Cosine Similarity measures the average of the cosine similarities between the returned word vector and the ground truth word vector for all queries. These three metrics are defined below for evaluation on N queries where $\mathbf{r_i}$ and $\mathbf{g_i}$ are the predicted word and ground truth word for query i, respectively:

$$\textit{Absolute Accuracy} = \frac{\textit{\# of correct predictions}}{N} \qquad \textit{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\textit{rank}_{g_i}}$$

$$\textit{Avg. Cosine Similarity} = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{p}_i \cdot \mathbf{g}_i}{\|\mathbf{p}_i\|\|\mathbf{g}_i\|}$$

5.3 Experimental Details

For our RAG approach, we utilized NetworkX graph database system to import nodes and relations we scraped from our Wikipedia data as previously described. This process of generating the knowledge graphs took approximately 3 hours for our set of articles. For the BERT-large baseline, we pretrained further on our Wikipedia data with a consisent learning rate of 5e-5 for a duration of 3 epochs, which required around an hour on a single NVIDIA A100 Tensor Core GPU. For the KnowBERT model, in total training was about 6 hours with 1 NVIDIA T4 for 1 epoch of 10000 steps and the same learning rate. Importantly, however, 24 GB RAM was necessary to perform the training.

5.4 Results

Our quantitative results for the next sentence prediction and factual recall tasks are presented in Tables 2 and 3, and are analyzed in Section 6.

Model	Accuracy	Precision	Recall	F1-Score	Perplexity
BERT-large	0.55	0.72	0.04	0.07	268.74
BERT-large Adapted (Baseline)	0.60	0.65	0.32	0.43	2.02
RAG BERT-base	0.61	0.66	0.33	0.44	2.00
KnowBERT	0.88	0.88	0.89	0.89	1.50

Table 2: Comparison of Next Sentence Prediction Model Performance

Model	Absolute Accuracy	MRR	Avg. Cosine Sim
BERT-large	0.39	0.0023	0.85
BERT-large Adapted (Baseline)	0.39	0.0023	0.85
RAG BERT-base	0.43	0.0013	0.87
KnowBERT	0.37	0.79	0.87

Table 3: Comparison of Factual Recall Model Performance

6 Analysis and Discussion

For next sentence prediction, fine-tuning BERT-large on our Wikipedia data resulted in improved scores for accuracy, recall, and F1 score, which was expected since the model is able to better learn domain-specific patterns and nuances in the data even with minimal fine-tuning. RAG, however, was almost identical in performance to adapted BERT, with just very slight improvements in scores. KnowBERT, on the other hand, does significantly better with all scores, with an impressive 47% improvement in accuracy over the baseline. This disparity between the two knowledge enhancement methods was relatively unanticipated, but may be attributed to a couple different reasons. KnowBERT embeds knowledge directly into the model during pre-training, allowing it to understand semantics and meanings more deeply, whereas RAG retrieves knowledge on-the-fly during inference, resulting in a less integrated understanding. This deep integration in KnowBERT should lead to more substantial performance improvements because the model can truly understand the semantics and meaning of the language. Additionally, RAG's retrieval mechanism risks introducing noise or irrelevant context, as the retrieved information might not always perfectly align with the task at hand, leading to only slight improvements.

The perplexity results from this task also offer a compelling insight into model knowledge enhancement. As perplexity is a measure of how well a probability distribution or model predicts a sample, it is exceedingly high for BERT-large at 268.74, indicating significant uncertainty and inefficiency in its predictions. Fine-tuning on the Wikipedia dataset, however, reduces perplexity dramatically to 2.02, showcasing how task-specific training can enhance model confidence and predictive coherence. RAG again had similar results to BERT adapted, with just a 0.02 improvement in perplexity. KnowBERT, though, achieves an even lower perplexity of 1.5, again suggesting that embedding the structured knowledge can further reduce prediction uncertainty and improve the model's overall linguistic and contextual understanding.

For factual recall, the absolute accuracy, MRR, and average cosine similarity metrics were exactly the same between BERT-large and BERT-large Adapted. This tells us that fine-tuning on this smaller dataset does not modify the underlying model weights enough to alter recall outputs. In contrast, KnowBERT exhibits slightly lower absolute accuracy, but significantly higher MRR, indicating that while it misses the exact match on a few more occasions (absolute accuracy), it ranks the correct answer much higher or with much more confidence on average (MRR). This supports the idea that directly embedded knowledge performs better in ranking relevant information, but when it errs, it does so more confidently, resulting in larger deviations from the ground truth.

RAG's performance, characterized by the highest absolute accuracy but lowest MRR, indicates a different set of challenges, opposing that of KnowBERT's. The high absolute accuracy suggests that

RAG's retrieval-augmented approach is effective at attaining the information necessary to identifying correct answers frequently. However, the low MRR points to potential issues with hallucination, where the model might confidently propose incorrect answers when it does fail to get such information, due to its reliance on external context without any actual semantic understanding. The results consistently emphasize the strengths and limitations of the two different model adaptations, particularly that knowledge embedding makes for a model that can genuinely understand and interpret the semantics of given text, but cannot always retrieve the additional contextual information that RAG can.

Some qualitative test results are displayed in Table 4 alongside the following prompts:

- Prompt 12: The Aperture Photometry Tool is used in the field of **[MASK]** astronomy.
- Prompt 94: Burning plasma is studied in the context of nuclear fusion in stars and **[MASK]**.
- Prompt 100: DGSAT I is an example of a faint ultra-diffuse **[MASK]** galaxy.

Prompt #	**Ground Truth**	**BERT**	**BERT Adapted**	**RAG-BERT**	**KnowBERT**
12	"optical"	"radio"	"radio"	"infrared"	"optical"
94	"reactors"	"planets"	"planets"	"planets"	"galaxies"
100	"dwarf"	"spiral"	"spiral"	"spiral"	"dwarf"

Table 4: Qualitative Examples

For prompt 94, it appears that BERT, BERT adapted, and RAG all predict the masked word as "planets" due to its relation to "stars." However, KnowBERT interestingly responds with the word "galaxies," which is more closely related to the notions of burning plasma and nuclear fusion in addition to stars than "planets," as galaxies contain all of these elements. Even more interestingly, prompts 12 and 100 showcase KnowBERT's stellar factual recall abilities. The terms "optical astronomy" and "dwarf galaxy" are not terms with common co-occurence in the English language; however, they are the only right answers in this scenario. KnowBERT having this information built into its encoding layers is able to pull this information and generate the right answers. The other BERT models have to rely on the words most commonly associated with the following word. Thus, we see "radio astronomy" and "spiral galaxy"; both of which are more likely than their KnowBERT response counterparts.

While purely qualitative, this form of testing builds further evidence that embedding knowledge into the model allows semantic information to be better extracted and utilized during generation. Even though all the models tended to get the result wrong, it is clear to us as humans, that KnowBERT's responses were more accurate than the others, and at the end of the day, it is that human intuition that we hope to emulate through models.

7 Conclusion

Our project results and findings consistently highlight the strengths and limitations of different model adaptations. They indicate that KnowBERT generally outperforms RAG and baselines, which can be attributed to KnowBERT's method of embedding knowledge directly within the model, which allows for a deeper understanding and interpretation of text semantics. On the other hand, RAG's ability to retrieve additional contextual information on-the-fly offers flexibility and access to a wider range of information, though it comes with the risk of introducing potential noise. Our achievements from this work include constructing our own knowledge graph full of scientific information, re-implementing RAG from scratch, and conducting a comprehensive range of evaluation metrics that showcase the capability to enhance model performance through various innovative approaches.

The primary limitations of our work include not testing other existing methods for knowledge enhancement and not exploring the full range of use cases for knowledge graphs, such as improving synonym detection and other linguistic tasks beyond factual reasoning. Future work could involve testing additional prompt engineering techniques and developing novel models for knowledge enhancement. For example, in our research we encountered the model SenseBERT (Leviant et al., 2019), which uses a knowledge graph not to just inject factual information but rather to disambiguate between word senses. Additionally, future research into combining the strengths of multiple architectures, such as integrating retrieval mechanisms with deeply embedded knowledge, may further improve model performance and accuracy, opening new avenues for advancing NLP capabilities.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. Exploring network structure, dynamics, and function using networkx.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Furstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 782–792. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. *arXiv preprint arXiv:2211.08411*, 2. Submitted on 15 Nov 2022.

Ido Leviant, Roman Sabo, and Roi Reichart. 2019. Sensebert: Driving some sense into bert. 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)*, pages 3973–3978.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2018. Scikit-learn: Machine learning in python.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

8 Ethics Statement

One significant ethical challenge that comes with this project involves the potential for these enhanced language models to reinforce existing biases. Since the project relies on a curated set of Wikipedia articles, any existing biases in these articles, whether related to gender, race, socioeconomic status, or other, could be inadvertently amplified by the model. This issue is critical as knowledge-enhanced models are likely to be perceived as more authoritative, leading users to trust their outputs without questioning potential biases. This can contribute to the perpetuation of stereotypes and unequal treatment of various groups. To mitigate this risk of reinforcing societal biases, we can employ a couple of different strategies. First, diversifying the dataset by incorporating articles from various perspectives, cultures, and languages (though still credible) can help ensure a more balanced representation of knowledge, thus reducing biases that come with using a limited set of data. Second, incorporating bias detection and mitigation algorithms into the model's training process or generation process can help identify and correct biased outputs.

Another ethical challenge is that there could be an over-reliance on these models' accuracies. Given the project's goal of creating a model that is highly accurate, there is a risk that users may over-rely on their outputs without cross-verifying with original sources or additional evidence. This can be especially problematic in academic, professional, and policy-making contexts where rigorous validation of information is necessary, and it could potentially contribute to the spread of misinformation. To mitigate this risk of over-reliance, we can make sure to continuously expand and update the knowledge base used for model training. Another strategy is to integrate functionality that cross-references the model's outputs with multiple authoritative sources, and if discrepancies are detected, the model can flag the output for further review. Finally, having transparency in data sourcing and providing users with information on the origin of the model's knowledge can help users critically evaluate outputs to make sure they are not trusting them more than they should be.