Distribution Agreement

Jacob Choi April 10, 2024

When Large Language Models Meet Religious Text

By

Jacob Choi

Jinho D. Choi
Advisor

Computer Science

Jinho D. Choi, Ph.D.
Advisor

Davide Fossati, Ph.D.
Committee Member

Hiram Maxim, Ph.D.
Committee Member

Helen J. Kim, Ph.D.
Committee Member

2024

When Large Language Models Meet Religious Text

By

Jacob Choi

Jinho D. Choi, Ph.D.
Advisor

An abstract of
A thesis submitted to the Faculty of the Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2024

Abstract

When Large Language Models Meet Religious Text
By Jacob Choi

The field of AI has been quickly expanding outside of Computer Science, including areas such as healthcare, transportation, and the humanities. The intersection between AI and religion is also a growing field, but there exists a lack of computational work done from an application-based perspective. The current intersection in research between AI and religion often involves observing information that the models have learned, such as religious bias. For works that more directly impact communities, commercial AI-powered tools are available to help users learn more about religious texts, but lack transparency, which may be alarming for some.

To contribute to the field of AI application in religion from a computational perspective outside of AI model bias observation, we perform a case study on the Bible by creating a verse extraction tool using deep learning techniques to showcase the process of creating such a tool for religious communities to use. To do this, we first explore a challenge common to those who study the bible by finding references. We utilized a semantic similarity search and the Hungarian algorithm to identify references, which we found infeasible yet impactful. We then introduce six datasets that we use to train a llama-2-7b-chat model to respond to user queries with Bible verses. Additionally, we create two test sets to evaluate models, the first asking fact-based questions and the second asking theological questions. We find that state-of-the-art commercial models still come out on top with the highest accuracy of 62.5 and 58.5, and we describe the next steps to encourage research toward this direction of application-based tools in the computer science domain for religion.

When Large Language Models Meet Religious Text

By

Jacob Choi

Jinho D. Choi, Ph.D.
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2024

Acknowledgments

I would like to acknowledge Professor Jinho Choi for his wisdom and knowledge, as well as his kindness and patience throughout my learning process. I want to thank undergraduate students Peilin Wu and Zinc Zhao for their encouragement, and graduate students Benjamin Ascoli, Sichang Tu, and Sarah Finch for their help. I especially want to thank James Finch for much insight. This project would not have gotten as far without his help. I would also like to acknowledge my friends who have patiently encouraged me through this process through their kind words or by providing me snacks, emotional encouragement, and spiritual encouragement by praying over me and helping me by evaluating my models. You know who you are. I want to thank my family who have greatly supported me in this process, especially in the most critical moments toward finishing this project. I lastly want to thank my God and Savior Jesus Christ for His renewing grace and mercy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The rise in applications of AI models and particularly large language models such as OpenAI's GPT-4, Google's Gemini, or Anthropic's Claude has been making its way into many fields and finding use outside of computer science. Fields such as healthcare [14][38], transportation [7][49], and environmental studies [37][24] are a few of the many fields that are beginning to adopt this technology. One of these many fields includes religion and theology [5][25], and this paper will touch on the language applications of deep learning models in this field.

There are currently different available commercial AI tools for particular religions that are designed to help users by answering questions they may have [1][2][3]. These chatbots are designed to answer questions about a piece of religious text, and other generative AI technology can be utilized for specific tasks like creating sermon outlines for pastors[4] or summarizing online sermons[5]. Although there are many commercially available tools, there is a current lack of work that explores this area from a research

[1] https://ravgpt.ai/ravgpt-chat
[2] https://qurangpt.live/
[3] https://www.biblemate.io/
[4] https://www.sermon.ly/
[5] https://pastors.ai/

perspective. To expand on the domain of utilizing AI tools in religion, we designed a verse extraction tool to help users find information from a religious text to answer questions they may have. We conduct a case study on the Bible by 1. utilizing computational methods to explore problems that have previously been attempted by hand and 2. training an AI model to take in user queries and return Bible verses that may apply to the users' needs. A verse extraction tool essentially takes a user input, such as a question like "What is the first verse in the Bible?" and responds with a verse relevant to that query, like "Genesis 1:1."

To showcase the reproducibility of these models, our work seeks to contribute to this domain from a research perspective. Namely, the novelty of our work highlights the step-by-step process of how we train an AI verse extraction tool from the ground up, which to the extent of our knowledge has not been previously done. This thesis covers particularly the creation of datasets as well as the fine-tuning process and the models used. The components needed to create this tool are aimed to be accessible to most, with our target audience being religious communities who may wish to utilize such a tool, but may not have the large computational resources to train their own models from scratch that corporations may have. By utilizing methods like Low-Rank Adaption [21], we can fine-tune a large pre-trained model on a single GPU so that vast, costly resources are not necessary to recreate such a tool. We find that although our methods do not beat state-of-the-art commercial models such as GPT, we would like to encourage future research in a direction towards building open-sourced models that are capable of serving communities who wish to develop their own, but do not have the resources to train a model from scratch or perform a costly fine-tune. Thus, our overarching thesis objective is to find ways to help users from religious communities answer questions they may have about a particular religious text through computational means. To tackle this, we explore the following two questions:

(i) 1. One common theme among religious texts involves the extensive use of references and cross-references, which we discuss in chapter 3. The task of finding cross-references in a text such as the Bible has been explored for centuries and has been manually done by hand [11]. Given the recent improvements in technology, can we use computational resources to find new references?

(ii) 2. With the advent of AI technology and its increased adoption in many fields, how can we create an AI-powered tool that's trained on data made using religious texts like the Bible to answer questions that users may have to help religious communities who may be interested in training their own model?

1.2 Research Objective

Our goal in this thesis is to explore methods to create the tools that can help empower religious communities by answering questions they have from a computational perspective. The contributions of this work include:

(i) 1. We apply one commonly used metric called a semantic similarity search to find reference verses in the Bible. We also attempt to use an algorithm called the Hungarian algorithm to tackle limitations from semantic similarity search in finding references, which to the extent of our knowledge has not yet been attempted.

(ii) 2. To help users answer questions they may have using AI, we utilize a state-of-the-art open-sourced AI model to train on datasets that we made to answer user questions.

Because of the costs needed to create a pre-trained model from scratch [42], our research thus tackles the question of how religious communities can train a verse extraction model through fine-tuning, which will be further described in chapter 4. By looking into this question, we also hope to raise awareness about the need for transparency in training large language models for religious use by releasing the datasets that the tool is trained on to encourage transparency of the training process in creating these models.

1.3 Thesis Organization

The rest of this thesis is presented as follows: Chapter 2 explores the current works on religion in the computer science field and provides a perspective on the work that is missing. Chapter 3 describes, in detail, our attempts to find reference verses using semantic similarity search and the Hungarian algorithm. Chapter 4 describes the process of training a model. Chapter 5 describes the approaches we took to create each of the six datasets used to train our model. Chapter 6 describes the outcomes from training the model and provides a discussion of how one should use this tool. Chapter 7 describes the limitations of this work and the potential future applications of this work. Lastly, chapter 8 considers the ethical and societal implications that come with AI usage in religious communities.

Chapter 2

Background

2.1 Background and Trends in NLP for Religious Text Analysis

Before describing works that are related to this domain, we begin this chapter by briefly describing the relevant background information for the upcoming discussion in section 2.2. This work we are conducting falls under the field of Natural Language Processing (NLP), which explores the computational means to create human-like text and speech through the combination of computational linguistics, which is the rule-based modeling of human language, along with statistical machine learning models. The field of NLP has grown substantially in the past few years with the advent of technology such as deep learning, along with the increase in available text data to train these models. One important aspect of research in NLP involves looking at a model's ability to carry out specific tasks. Tasks such as named entity recognition (NER), which aims to identify and classify entities in text such as people, organizations, dates, locations, and more, are important because they deal with extracting relevant information from text, which is useful in the real world for examples like identifying people or dates in a news article so that a user can quickly understand what the article is about. Some of the works that we describe below will look at tackling specific tasks from a religious perspective.

In addition to an emphasis on how well models can perform certain tasks, an emphasis on creating datasets is just as important. This is because the quality of the dataset greatly affects the model's performance. Although data from the internet is often vast and accessible, it also comes with human biases that may affect the model's performance. There is a growing amount of work regarding computer science applications in religion, and one common research focus involves observing bias in datasets, oftentimes including religion.

Lastly, there is a growing trend of developing models with a large amount of parameters, also known as large language models. These models are impressive because of their capabilities to perform a multitude of NLP tasks given their large parameter size, especially since they often do not need further fine-tuning. This is surprising because conventional, smaller language models like BERT are capable of performing specific tasks, but this requires additional steps such as further training on a dataset specific to the task, modifying the model's underlying architecture, or a mixture of both. Works described in the following section involve adapting to tasks using smaller models like BERT, whereas our research instead explores the usage of larger models.

2.2 Exploring NLP Tasks in Religious Texts

Computer Science research on the perspective of religion often observes the amount of bias that is present in models. Works such as Abid et al. [3] showcase stereotypes such as anti-Muslim bias in language models like GPT-3 by showcasing how it associates certain words with religious groups such as 'terrorist' with Muslims or 'money' with Jews. This analysis looks into the outputs of a model that may be biased towards a particular religion based on the data it has been trained on. Additionally, datasets have been created to evaluate bias on certain stereotypes including religion [33][34].

Although there has generally been a lack of work exploring religion outside of bias in computer science, recent efforts have been made to study NLP tasks in religious texts. The Quran QA Shared Task Challenge in 2023 invited teams to develop models for tasks such as passage retrieval (Task A) and reading comprehension (Task B). For Task A, once the team is given a question, they must identify Qur'anic passages that contain answers to those questions. For Task B, once given a question and a passage, teams must find all the answers from that passage. Alnefaie et al. [6] attempts Task A by fine-tuning on Arabic pre-trained models and performing a series of additions such as fine-tuning on extra data, using an ensemble of different models, and performing post-processing for their results. Although their approach is effective to excel in these particular tasks, the challenge suggests building upon previous methodologies to accomplish these tasks, which mainly utilize smaller, task-oriented language models like AraBERT [8], a BERT model trained on large, public Arabic corpora, rather than working with larger, generative models such as GPT [10]. Our work explores utilizing these larger, generative models to tackle a variety of tasks at once. In addition to passage retrieval and question answering, our trained model attempts to handle tasks such as semantic similarity to find similar verses, finding cross-references, utilizing named entity recognition, and performing contextual understanding, all of which are further described in chapters 5.4.6, 4, and 5.

Our motivation to create this model stems from the current lack of literature that explores a variety of NLP tasks on biblical data. Current usage of the Bible for NLP tasks involves creating large-scale datasets for low-resource languages due to its diversity of translations [29][30][18], or to tackle conventional NLP tasks for low-resource languages [4]. Considering the Bible's widespread usage but lack of general exploration of its content for NLP tasks, we hope to bring light to research in this domain.

Lastly, there exist models commercially available that have been trained on a religious text. However, the content they are trained on is still hidden behind a black box, meaning that the data that these models are trained on is not open-sourced. Likewise, their models are not available for download or fine-tuning, with examples being OpenAI's GPT or Anthropic's Claude. Although their overall performance in NLP tasks on biblical content remains the strongest, as we further discuss in chapter 6, we hope that this effort brings forth consideration towards developing models that are trained on open-source datasets that can achieve comparable performance to these commercial, black-box models, particularly in a domain like religion that contains nuances in regards to dogma and benefit from transparency.

Chapter 3

Finding References: An Exploration

The Bible has many references and cross-references. We define a reference verse in the Bible as a verse being referred to by a verse later in the Bible. As an example, Jesus references Exodus 20:13, which states " "You shall not murder." (NIV), which Jesus refers to when he says " "You have heard that it was said to the people long ago, 'You shall not murder, and anyone who murders will be subject to judgment.'" (NIV) in Matthew 5:21.

Cross-reference verses are verses that may not word-for-word be referenced from another verse but may also contain similar themes, events, or people. One example would be Hebrews 11:3's verse "By faith we understand that the universe was created by the word of God, so that what is seen was not made out of things that are visible." (NIV) and its cross-reference to Genesis 1:1's verse of "In the beginning, God created the heavens and the earth." (NIV). Although Genesis 1:1 is not a reference to Hebrews 11:3, they do share similar themes of the Bible's mentioning of the origin of the world and are thus considered a cross-reference.

At the start of the project, we had set out to find ways of finding reference verses. Scholars have gathered references together over many years, but a recent advancement in computational power has allowed for this task to be potentially automated. We were thus curious to know if there were ways of finding new reference verses through computational means.

An exploration was done in an attempt to find these reference verses. The NIV Bible contains 31,103 verses, and each book and chapter along with their verses, titles, and commentary which includes references and cross-references were scraped from the BibleGateway website[1].

Translation	**References**	**Percent Overlap**
NIV	314	0.4331
NLT	167	0.8144

Table 3.1: Number of references and percentage overlap between translations.

Statistic	**Value**
Number of Common References	136
Total References After Merging	345

Table 3.2: Number of overlapping references and references from merging.

Popular Bible versions KJV, NKJV, NIV, ESV, and NLT were scraped to be compared with. Only NIV and NLT contained references in their respective footnotes from BibleGateway. NIV had a total of 314 references, while NLT had a total of 167, as denoted by table 3.1. Their respective references in common and the total number of references combined after merging is shown in table 3.2. By using the commentators' footnotes on reference verses from NIV on BibleGateway as the ground truth to compare and test our proposed methods, we attempted to find reference verses through a semantic similarity search and finding maximum weight matchings.

The scraped information from BibleGateway is formatted in JSON and contains the book version, the book, the chapter, and the verse. Verses contain both the verse text, and any references made, as well as the commentary for that verse if there exists any, with the character offset for the commentary that the commentator made. The

[1] https://www.biblegateway.com/

verse will also contain a title if it is the first verse of a passage, which is denoted in bold in the NIV version.

3.1 Semantic Similarity

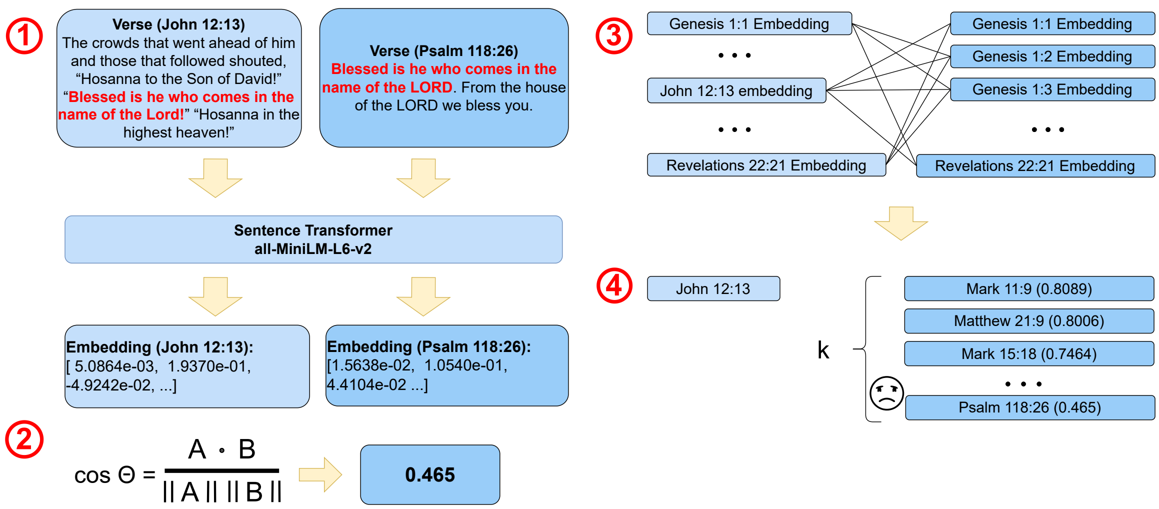


Figure 3.1: Semantic Similarity Matching Algorithm

To find how similar two verses are through computational means, we employ a semantic similarity search. When a model has a task such as figuring out the similarity between words, sentences, or documents, the model cannot decide this by reading characters. These words have to first be converted into a format that can be used to train a classifier, and this is done by mapping words into a vector space, which we call word embedding. Words such as 'cat' and 'dog,' are closer to each other than to a word such as 'airplane.' To perform this mapping, traditional word embedding tools like [32] learn the meaning of words based on the occurrence of surrounding words. An example that Jurafsky and Martin [23] uses is the word ong choy, which you likely have not heard before, but from context sentences such as "Ong choy goes well with rice" or "Ong choy is delicious when sauteed with garlic" let one infer that ong choy is food related. When other sentences use different words that are also described in a similar sense, such as "spinach is delicious when sauteed with garlic," one can infer that ong choy may be a leafy green similar to the other described leafy greens. Similarly, a statistical machine learning model is trained by taking each unique word in a document that is given and learning an embedding for each word based on how likely a separate word is a context word of the original word (using a window of surrounding words of the original word as ground truth). The learned parameters from this model for each word can be used as the embedding for the word, which we will use to compute similarity.

Many metrics exist to measure these distances, such as Euclidean distance or Manhattan distance, but the distance we choose is the widely-used cosine similarity metric, which captures the semantic similarity between two words by measuring the cosine of the angle between the vectors, with a value of 1 being an exact match. The previous example we described is focused on embedding words, but this can also be applied to entire sentences, or in our case, Bible verses. To embed verses, we utilize the sentence transformer library [2]. We create embeddings for each verse using an all-MiniLM-L6-v2 sentence transformer model to generate a 384-dimensional vector for each verse, which can be seen in step one in figure 3.1. This allows sentences, and in our case, verses, to have embeddings, and using their embeddings we can calculate numerical distances between verses using cosine similarity to find their semantic similarity, which can be seen in step two in figure 3.1. Creating embeddings and performing pairwise similarities between many verses can be computationally demanding and take very long, so to overcome this, we utilize Meta's FAISS library [1] to perform a quick similarity search. What would traditionally have taken hours to find the most similar pair among tens of thousands of sentences takes seconds using FAISS.

A similarity search in our scenario involves building the embeddings of each verse. We can then perform a pairwise comparison and find the cosine similarity between each verse and all 31,102 other verses in the Bible, resulting in a 31,103 x 31,102 comparison, as shown in step three in figure 3.1. We then generate a list of similar verses from highest similarity to lowest similarity, and we manually select a threshold value of 0.6699 and display verses with similarities greater than this value. From our results, we found that this works generally well. Using the FAISS library, we can generate the top k similarities of a particular verse, shown by step four in figure 3.1. Table 3.3 highlights the number of references caught with varying levels of k.

k	Similarity	Hungarian
5	184	202
10	223	231
15	235	243
20	241	251
25	250	261
30	255	265
35	260	269

Table 3.3: Number of correct reference matches between semantic similarity search and the Hungarian algorithm for different values of k out of 345 references.

However, we noticed that matches between reference verses and verses referring were not being found because the reference verses were often only a subset of the verse. This means that although one part of the verse referring may, word for word, refer to another verse, the rest of the words in that verse may not be 'similar.' For example, in 2 Kings 23:27, the verse reads: "So the LORD said, "I will remove Judah also from my presence as I removed Israel, and I will reject Jerusalem, the city I chose, and this temple, about which I said, 'My Name shall be there.' "" and references 1 Kings 8:29, which states: "May your eyes be open toward this temple night and day, this place of which you said, 'My Name shall be there,' so that you will hear the prayer your servant prays toward this place."

The reference lies in the words 'My Name shall be there.' but the surrounding words may be causing the engine to yield a lower overall similarity. The engine's highest generated similarity for 2 Kings 23:27 is 1 Kings 9:7, which states: "then

I will cut off Israel from the land I have given them and will reject this temple I have consecrated for my Name. Israel will then become a byword and an object of ridicule among all peoples." and has a cosine similarity of 0.736, whereas the cosine similarity with its reference verse, 1 Kings 8:29, is 0.5396. Although this similarity search method performs fairly well with finding verses, we attempt to address its limitations by using a second method called maximum weighted search.

3.2 Maximum Weight Matching

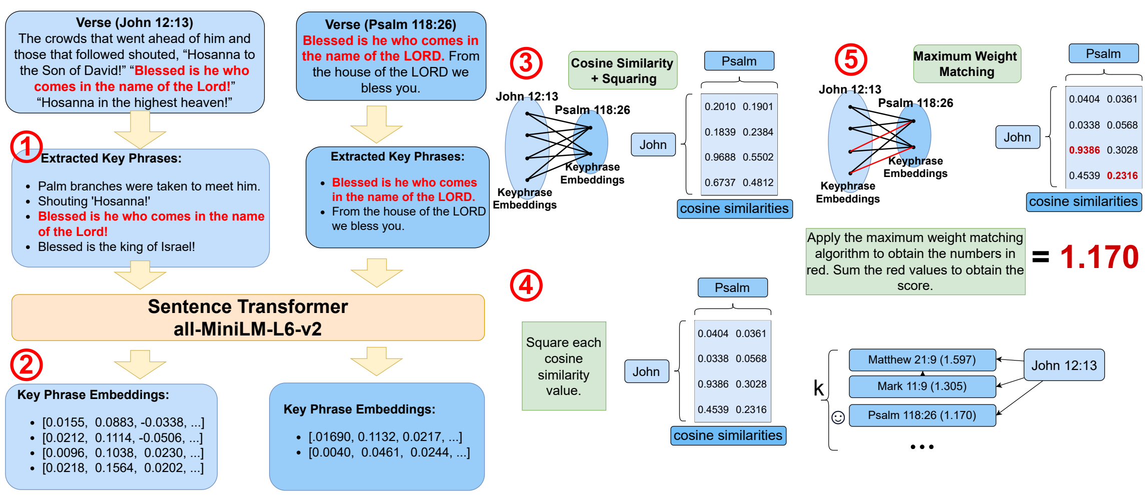


Figure 3.2: Maximum Weight Matching Algorithm

We explored a second computational approach to match reference verses with their respective verse and address limitations from similarity search through an algorithm called the Hungarian matching algorithm, which is used to find the maximum weight matching in a bipartite graph. We can use a similar approach to finding semantic similarity between two verses by comparing similarities of embeddings, but instead of creating an embedding for the entire verse, we can generate key phrases from these verses and use their embeddings to create a bipartite graph. To generate key phrases, we compare a variety of tools. We explore phrases generated from three different transformer-based models: KeyBERT [16], BERTopic [17], and GPT 3.5-turbo.

We first began exploring generating phrases by utilizing BERTopic. BERTopic utilizes state-of-the-art embedding techniques alongside HuggingFace transformers to perform topic modeling, which trains a pre-trained model on unlabeled documents to generate topics. For each Bible verse, we utilized the BERTopic model to generate key phrases. We generate 393 number of unique topics. However, given the diversity of verses (31,103 total verses for the NIV version), we would like to generate more unique key phrases for each verse to catch more potential references.

To generate more unique key phrases that have a better chance of catching reference verses, we then attempt to utilize KeyBERT. With KeyBERT, we can extract keyphrases by first embedding a document. A document in our case will be a verse. Then word embeddings are extracted for n-gram phrases. Cosine similarity is then used to compare these n-gram phrases with the document, and phrases are selected using the scores that yield the highest similarity to the document. Keyphrases with ngrams 3, 5, 7, 9, and 11 were utilized, each with 129,129, 145,728, 148,114, 148,700, and 148,928 unique phrases, respectively. We found that GPT 3.5-turbo generated the best phrases through manual inspection, so we used GPT 3.5-turbo to extract the key phrases. GPT 3.5-turbo is capable of generating 75,984 unique phrases.

The verse John 12:13, reads: ”They took palm branches and went out to meet him, shouting, “Hosanna! ” “Blessed is he who comes in the name of the Lord!” “Blessed is the king of Israel!”” and can be extracted into key phrases such as

1. Palm branches were taken to meet him,
2. Shouting ’Hosanna!’
3. Blessed is he who comes in the name of the Lord!
4. Blessed is the king of Israel!

John 12:13 makes a reference to Psalm 118:26, which reads: "Verse (Psalm 118:26) Blessed is he who comes in the name of the LORD. From the house of the LORD we bless you." This verse can also be broken down into the following phrases:

1. Blessed is he who comes in the name of the LORD.

2. From the house of the LORD we bless you.

This extraction can be seen in step one in the figure 3.2. These phrases are then each embedded using a sentence transformer, as denoted by step two in figure 3.2. Different from a semantic similarity search, instead of obtaining a semantic similarity score by performing a pairwise comparison between verse embeddings using cosine similarity, we obtain a score by performing a pairwise comparison between scores generated from finding the maximum weighted match between two sets of keyphrase embeddings. This can be seen in step three from figure 3.2 with embeddings of the keyphrases from John 12:13 and Psalm 118:26 being matched. The edge weight is found by calculating the cosine similarity score between the embeddings of the key phrases. We can thus compare 31,103 sets of keyphrase embeddings with 31,102 sets of keyphrase embeddings, which excludes matchings for the same verse and phrases. Before finding the maximum weight matching, we square the weights of the edges to increase scores between keyphrases that have a higher potential of being a reference phrase while decreasing scores that are less likely to be reference phrases, as shown in step 4 for figure 3.2.

Lastly, after finding the maximum weight matching, as seen in step 5 from figure 3.2, we can then sort the scores generated between sets of keyphrase embeddings for each verse by descending order to find verses that are most likely to contain a reference verse. We notice that the correct verse John 12:13 is referencing, Psalm 118:26, is ranked higher than a regular semantic similarity search.

To formalize this approach, let V be the set of all verses in the Bible that we extract from BibleGateway. For each verse $v \in V$, let K_v be the set of key phrases extracted from verse v that we accomplished using GPT. Let v_1 and v_2 be different verses from V. For each keyphrase, $k \in K_v$ and verse $v \in V$, let $S_k(v_1)$ be the embedding of keyphrase k from verse v that we obtain using a sentence transformer. Our next step is to compute the similarity between two verses using the embeddings. To do this, we can create two sets of vertices, which are our embedded keyphrases: $V_k^{(v_1)} = \{S_k(v_1) : k \in K_{v_1}\}$ and $V_k^{(v_2)} = \{S_k(v_2) : k \in K_{v_2}\}$. We can then construct a bipartite graph for each verse pair $G = (V_k^{(v_1)}, V_k^{(v_2)}, E)$, where each vertex $S_k(v_1)$ from $V_k^{(v_1)}$ is connected to every vertex $S'_k(v_2)$ from $V_k^{(v_2)}$ by the edge $(S_k(v_1), S_{k'}(v_2))$. This is the same as having each keyphrase from a verse matched with every other keyphrase from the opposite verse. We can assign a weight function, $w(e)$, such that $w(e) = (sim(S_k(v_1), S_{k'}(v_2)))^2$, where $e \in E$ and $sim()$ denotes the cosine similarity function used to obtain the similarity score between the embeddings. The similarity value is squared to increase the scores between keyphrases that are more similar to each other, denoting a possible reference. $M(v_1, v_2)$ can be defined as our matching set. $M(v_1, v_2) \subset E$, and, in this example, every element in v_1 has an edge connecting to an element in v_2 without sharing the same endpoint.

Our goal is to maximize M such that $\sum_{e \in M(v_1,v_2)} w(e)$, where we can obtain our maximum weight matching score. By performing a pairwise comparison between each verse in the Bible, we can sort these values to find which verses share the greatest similarity.

For our first attempt at maximum weight matching, we utilized the following prompt in GPT 3.5-turbo with a temperature of 0.2 to generate key phrases: **Extract key phrases as a bulleted list from the following:** which was followed by the verse. We then extracted the bulleted list of key phrases for the verse to be embedded. From table 3.4, we notice that the performance of using the extracted key phrases from this attempt, called "Original Phrases," does not score particularly well. To

k	Original Phrases	Similarity	New Phrases
5	121	184	202
10	144	223	231
15	156	235	243
20	173	241	251
25	180	250	261
30	187	255	265
35	201	260	269

Table 3.4: Original vs. New Phrases + Squaring for Hungarian Algorithm compared with similarity. Number of references caught out of 345.

improve upon this, we utilize a new prompt to extract key phrases: **"Generate grammatically correct, simple key phrases for the text."** Additionally, phrases that yield higher scores are likely to contain references as subsets of a verse, as we had explained previously in the example with 1 and 2 Kings. Thus, to boost high cosine similarity scores higher and bring down low matching scores even lower, we square the cosine similarity values before running the Hungarian algorithm. The results show a noticeable improvement, as denoted by the scores under "New Phrases" in table 3.4. Despite our improvements, however, our reference matching is still far from an ideally higher reference matching rate to be usable by our model. Thus, we conclude this section with our results and attempts to match references and leave possible improvements to future work.

Additionally, although we attempt to speed up the search by utilizing the cosine similarity function from the Pytorch library with CUDA, one run of this algorithm still takes approximately 40 hours, and any method to speed up this process would allow for quicker testing.

3.3 Takeaway

Although both methods failed to find an adequate number of references, we believe that our experiments and insights may provide a future direction toward ways to find these references through computational means and perhaps discovering new cross references. Additionally, despite the difficulty of finding new references using these techniques, we still utilize data gathered from the semantic similarity search in our Similarity dataset, which we discuss in section 5.4.1, to suggest similar verses to users for verses that do not have cross-references.

Chapter 4

Model Training

4.1 What is a language model?

A language model is a statistical machine learning model that is trained to predict a distribution probability of the most likely words that will occur next in a sequence. A toy example can be shown by having a model create a sentence by generating one word at a time starting with the word 'I'. The next word likely ought to be a verb, such as 'am' or 'ran.' This ability to predict the next word has been learned from the model after having been trained by being given large amounts of text data, which has likely seen many occurrences of the word 'I' followed by the word 'am' or 'ran'. The model learns from patterns in this data so that it can output text similarly to humans. Training a model on a large amount of text to carry out specific tasks yields a pre-trained model or foundation model, and pre-trained models are further fine-tuned to tackle a specific task.

Our goal is to create a verse extraction tool that understands a user's query and responds with a verse that is relevant to that query. A model was likely not initially trained for extracting verses to answer user queries. It is possible to train a model from scratch to handle our specific task, but this is also extremely costly and infeasible in the scope of this project. Thus, to make the model knowledgeable about our specific domain, we can fine-tune our model.

4.2 What is fine-tuning?

A foundation model contains learnable parameters, also called weights, that are updated to learn information from datasets to output human-like text, as we discussed earlier. Fine-tuning a model often involves further updating a model's weights after pre-training so that the model can better adapt to handle a specific task. This is because although a model has learned a lot of information from the large amount of data it has been trained on, these models are typically capable of performing well on a variety of tasks, but may struggle with nuanced, domain-specific information. To have a model perform well on domain-specific data, additional modifications need to be done. Considering our domain-specific task of verse extraction, we can consider doing this by either updating the weights using the pre-trained model through fine-tuning, or we can utilize methods that don't interfere with the base model like retrieval augmented generation (RAG) methods [27].

RAG essentially utilizes a language model along with all of its pre-trained knowledge and draws domain-specific information from an external source, such as a database. This is useful because updating existing model weights has the potential for a model to interfere with weights containing existing information it has learned, leading to a phenomenon known as catastrophic forgetting, where a model forgets the information it has previously learned, affecting performance for your task. This database contains information that is embedded, and a semantic similarity search between a user's query and the most relevant information from that database is used to retrieve an answer. This is particularly useful because this does not involve any further updating of the model weights. For our needs, we decided to go with fine-tuning, as we can train the model to output in a specific format we would like, which is Bible verses, and fine-tuning may yield more accurate responses than RAG methods.

Fine-tuning will also allow us to train a model to output Bible verses without having to create a domain-specific model by training from scratch. Although fine-tuning is not as resource-intensive as training an entire model from scratch, performing a full fine-tuning, which involves updating all of a model's parameters, is still resource-intensive. We thus utilize state-of-the-art techniques to fine-tune a model using a fraction of the resources while having performance that is comparable to a full fine-tune. To perform this, we utilize Low-Rank Adaption, also known as LoRA [21]. Typical fine-tuning involves updating the pre-trained model's original weights. However, LoRA proposes that these changes can be captured using a lower-dimensional representation. By freezing, or not allowing the pre-trained model's weights to change, we can achieve a low-rank representation of the pre-trained model's weights using two smaller matrices, A and B. The product of these two matrices is a low-rank approximation of this weight matrix, but the number of trainable parameters is significantly less. This means that we can achieve comparable performance to full fine-tuning with a reduced memory footprint and faster training by reducing computational demands. Figure 4.1 showcases a visual representation of LoRA. As mentioned by Hu et al. [21], a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$ can be represented by a low rank-decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ and $r << min(d, k)$, where BA yields the shape $d \times k$ so that ΔW can be trained and directly merged with W_0. Additionally, the input x is multiplied by both W_0 and ΔW so that we obtain $h = W_0 x + \Delta W x$.

What would typically require multiple high-memory tensor core Graphics Processing Units (GPUs) for fine-tuning can be reduced to a single GPU, reducing memory usage by up to 3 times. This makes fine-tuning accessible for consumers, and in our scenario, religious organizations.

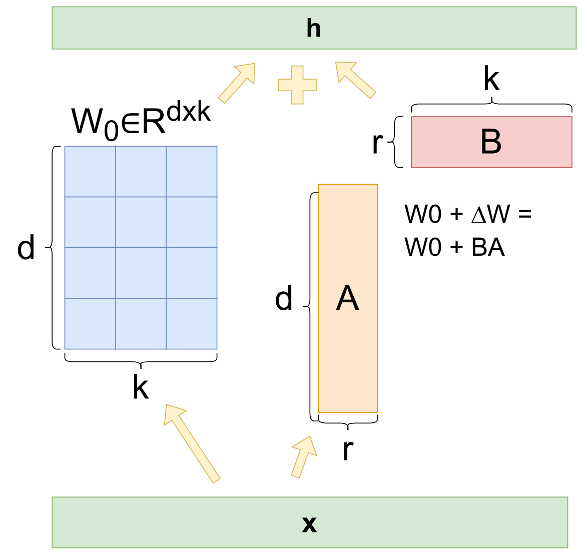


Figure 4.1: Fine Tuning with Low-Rank Adaption

4.3 Model Choice

Even with state-of-the-art fine-tuning methods to reduce memory consumption, models with a large number of parameters will still require using multiple GPUs. Thus, to manage fine-tuning on a single GPU, we resort to using models on the "smaller" end. The model we specifically choose to use is a 7-billion parameter llama-2 through the Huggingface library [1]. Huggingface is an online library that contains models and datasets that users can upload or download to use or fine-tune. Between a variety of capable models such as Flan-T5-xxl [13], llama-2-7b-chat [45] and mistral-7B-instruct-v0.2 [22], we ultimately decide on utilizing llama-2 because of its capabilities of understanding user queries. Although we attempted working with a mistral model initially, the difficulty in adapting it to our domain-specific needs led us to utilize llama-2 instead, which was much more manageable to work with. Additionally, a llama-2-chat model was selected over the base model due to the instruction fine-tuning preset, which is the strategy we also employ for fine-tuning.

[1] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

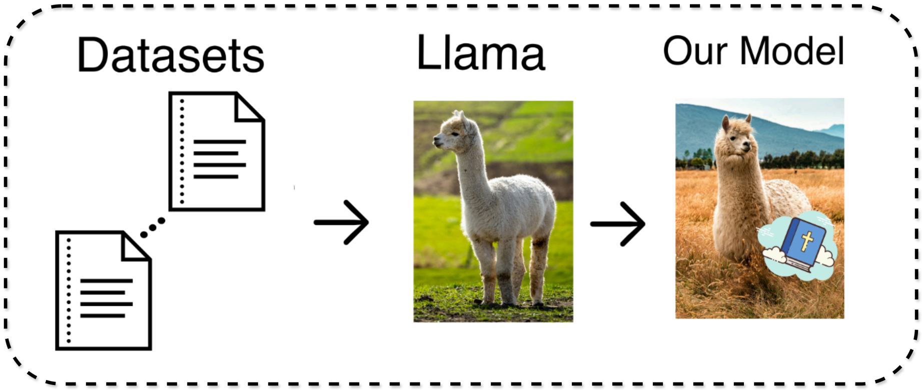


Figure 4.2: Pipeline of model

4.4 Training Pipeline

To create a verse extraction tool, we utilize a foundation model and datasets targeting different tasks. We first create a set of datasets, and then we fine-tune our llama-2 model on these datasets using LoRA [21] for efficient, low-cost fine-tuning on a single GPU. To promote transparency throughout the entire model training process, we would ideally train a model with open-sourced datasets from scratch, but given our computational constraints, we opt to fine-tune a model instead, which is also likely in the scope and budget of religious organizations that may want to train their own model.

Chapter 5

Datasets

In this chapter, we describe the methods we used to create the datasets that were trained with our foundation model. We then elaborate, in detail, the function of each dataset and how the datasets were built.

5.1 Overview of Datasets

We begin this section by describing commonalities between all of the datasets that we generated, including Bible versions used, and describing what instruction fine-tuning is. The following subsections then further describe the nuances used to generate each particular dataset by presenting the technicalities for generating data as well as providing examples from each dataset. Additional details can be found in the appendix.

For our tool, we create six datasets that are named accordingly: Similarity, NER, Translation, Application, Single, and References. The six datasets are concatenated into one large dataset, titled 'Combined', to contain all of the functionalities from each of the six datasets that are used to train a llama-2-chat model.

One underlying technique we use to generate the dataset is utilizing OpenAIs GPT API to generate the data. We do this by creating prompts and extracting the answers that GPT generates.

5.2 Bible Versions

The verses used to generate data were done using the NIV version for all datasets except for the Translation dataset. The Translation dataset utilizes five popular Bible versions: NIV, NKJV, KJV, ESV, NLT, which are further described in section 5.4.3.

5.3 Instruction Fine Tuning Format

As a brief overview, all datasets were formatted to be trained in instruction fine-tuning format. Typical fine-tuning approaches can be done using supervised or unsupervised methods, with the difference being whether or not the data has labels, also known as ground truth. Typically, pre-trained models are fed vast amounts of data for training, which can look like raw text data. This data, normally unsupervised, is used for the model to learn through a process called language modeling, where the model first tokenizes or divides the raw text data into smaller units to be processed. The model then learns by predicting the next token based on the previous sequences of tokens and adjusting its parameters based on the dissimilarity between the likelihood the model predicts the next token to be compared to the actual probability distribution of the next token.

Supervised fine-tuning often involves a smaller, higher-quality subset of data that includes labels for the model to learn from. As opposed to updating the model's weights by predicting the next token, the model can instead learn a particular task with these labels. One example includes an NLP task of sentiment analysis, where one can create a dataset with sentences and corresponding labels with 'positive' and 'negative,' to train a model to learn which sentences are associated with positive or negative sentiment. Instruction fine-tuning is a form of supervised fine-tuning in that it contains instructions as input and learns with its corresponding output. However, the use of instructions allows the model to generalize and learn different tasks, as opposed to traditional fine-tuning methods that train models to tackle one specific task. Wei et al. [47] showed that fine-tuning language models through instructions substantially improves performance on unseen tasks compared to their non-instruction-tuned counterparts. The use of instruction fine-tuning here is applicable, as we aim to train a tool that is not limited to one specific task but can generalize to unseen tasks that the user may ask. However, creating quality instructions by hand is costly. We thus follow Wang et al. [46]'s approach of utilizing language models, which are just as capable of producing texts similar to humans [10], to generate instructions for our dataset.

One generalized sample of the instruction is presented as follows:

{instruction: question output: Bible verse}

where the Bible verse could range from one verse from a particular book and chapter to multiple verses from a particular book and chapter to multiple verses from different books and chapters. Additionally, one particular dataset compares different Bible verse versions and thus contains the particular Bible verse version in its output. Examples from each dataset can be seen in their respective sections below.

5.4 Datasets

5.4.1 Dataset 1: Similarity

{"**instruction**": "What is the cross reference for Genesis 1:7?", "**output**": "Psalm 148:4, Proverbs 8:28-29"}
{"**instruction**": "What is the cross reference for Luke 10:23?", "**output**": "Matthew 13:16-17"}

Figure 5.1: Similarity Dataset References Examples

The similarity dataset is a combination of both cross references to Bible verses,

Dataset Name	Total Samples	Avg. Token Len
References	1,253	38.64
Combined	528,604	53.87
Single	126,030	31.75
Situation	75,776	82.00
Version	155,458	54.12
NER	73,301	31.62
Similarity	82,847	80.97

Table 5.1: All Datasets Information

{"**instruction**": "Revelation 21:17 has no cross references, but here are verses that may be similar:", "**output**": "Ezekiel 40:5"}
{"**instruction**": "Esther 9:8 has no cross references, but here are verses that may be similar:", "**output**": "Joshua 15:22"}

Figure 5.2: Similarity Dataset Similar Verses Examples

as well as similar Bible verses. A distinction between the two should be, however, made. As mentioned in chapter 3, cross-references can be defined as a mentioning of a Bible verse in another part of the text. Similar verses are verses that may be close in semantic meaning or structure and could potentially include cross-references, but this is not always the case. To create this dataset, the data from OpenBible[1] was utilized to generate cross-references, and users manually voted for cross-references. Through manual inspection, cross-references with at least 5 votes were deemed as valid cross-references. Examples of cross-references from Genesis can be seen in figure 5.3. Prompts to generate instructions for the cross-references were as follows: **Come up with ways of asking what is the cross reference of {value }.** In this text, value is the main verse, where cross-references are being made to. In the case that cross-references for a particular verse did not exist, the instruction: **"The following verse does not contain cross-references, but here are similar verses instead"** was used. To extract similar verses, a semantic similarity search was done using Facebook FAISS [1], which is covered in detail in section 3.1. A search using cosine similarity was done, and similar verses above a threshold of 0.6699 were manually

[1] https://www.openbible.info/

From Verse	To Verse	Votes
Gen.1.1	Neh.9.6	77
Gen.1.1	John.1.1-John.1.3	273
Gen.1.1	1 John.1.1	38
Gen.1.1	Acts.17.24	101
Gen.1.1	Isa.65.17	34

Figure 5.3: OpenBible Cross References Examples

selected and considered to be similar verses. Examples from this dataset can be seen in figure 5.1. Additionally, samples of verses that don't contain cross-references but are referred to as similar verses instead are shown in figure 5.2.

5.4.2 Dataset 2: Named Entity Recognition

{"**instruction**": "Who was 105 years old when he became the father of Enosh?", "**output**": "Genesis 5:6"}
{"**instruction**": "Who are the sons of Levi?", "**output**": "1 Chronicles 6:1"}

Figure 5.4: NER Dataset Examples

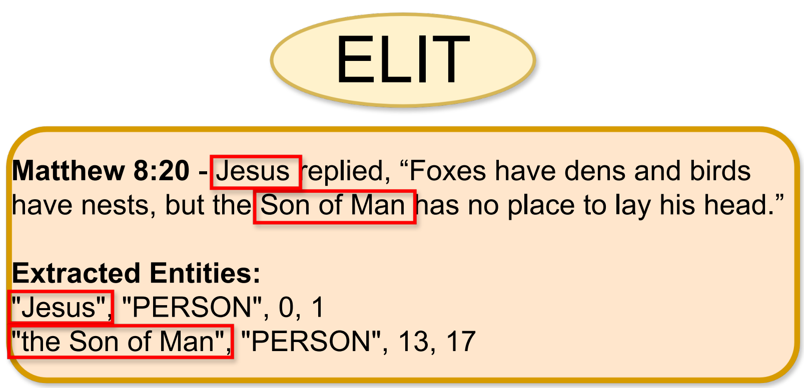


Figure 5.5: NER Extract Entities Example With Token Offset

Named Entity Recognition (NER) was employed to extract entities from the Bible. A total of 18 entity types were extracted, and a few notable entities include PEOPLE, NORP (national or religious or political groups), GPE (geopolitical entities), and more. For a more detailed description of the particular entities involved and examples of such entities, please refer to the appendix A. To extract these entities, Emory Language and Information Toolkit (ELIT) [19] was utilized. To evaluate the effectiveness of ELIT on a document such as the Bible, a validation set of 100 samples of Bible verses and their respective entities was created to test this, and ELIT was able to achieve a micro and macro F1 of 0.845. ELIT was utilized to find every verse that contained entities and was then utilized with GPT 3.5-turbo to generate questions about the verses using their extracted entities. For the named entity recognition (NER) dataset, we employ the following prompt with a temperature of 0.2: **Create questions a user may have using entities:** Here we append entities extracted from the verse as a list. We only call GPT if entities were extracted from the verse, otherwise, the verses are skipped. Examples from this dataset are shown in figure 5.4. Additionally, an example of extracted entities from a verse using ELIT is shown in figure 5.5.

5.4.3 Dataset 3: Version

{"**instruction**": "This is my command: Love each other.", "**output**": "John 15:17 NIV"}
{"**instruction**": "These things I command you, so that you will love one another.", "**output**": "John 15:17 ESV"}

Figure 5.6: Version Dataset Examples

To tackle the case where users may want to know a verse's particular Bible version, the version dataset is created. This dataset contains every verse in the Bible from 5 different versions. The versions used in this dataset include the New International Version (NIV), English Standard Version (ESV), King James Version (KJV), New King James Version (NKJV), and the New Living Translation (NLT). Verses from the different Bible versions are also scraped from BibleGate, and examples from this dataset are shown in figure 5.6.

{"**instruction**": "What does the Bible say about God's omniscience?", "**output**": "1 John 3:20, Psalm 147:5"}
{"**instruction**": "How does the Bible instruct believers to dress modestly?", "**output**": "Deuteronomy 22:5"}

Figure 5.7: Situation Dataset Examples

5.4.4 Dataset 4: Situation

The situation dataset was created by utilizing the OpenBible website. OpenBible has sections about specific topics that people may have and accompanying verses that address those topics, such as marriage, anger, government, etc. There were 6,600 topics in total. These verses that address those given topics are also suggested through human votes, and by manual annotation, we consider relevant verses as those having at least 3% of the total vote. We then utilize GPT 3.5-turbo with a temperature of 0.8 to use those topics to generate questions with two different prompts to handle two different tasks. The first prompt handles the task where the user may have questions that directly target the Bible about the given topic. The second prompt handles queries where users may not have a direct question, but describe a life situation that is related to the topic and may want a verse that applies to their situation. Examples from the dataset can be shown in figure 5.7. Here is the first prompt used to generate the dataset: **”Come up with questions one may have about:\n \n value \n \n that can be found in the Bible.”** The second prompt was used to generate the dataset: **”Come up with life situation examples in simple language that one may have from: \n \n value \n \n in first person that can be answered from the Bible.”** In each of these cases, ”value” contains the topic from OpenBible, with the corresponding verses that users voted on. Examples from this dataset can be seen in figure 5.7.

{"**instruction**": "What did God create in the beginning?", "**output**": "Genesis 1:1"}
{"**instruction**": "What will flow out of the LORD's house?", "**output**": "Joel 3:18"}

Figure 5.8: Single Dataset Examples

5.4.5 Dataset 5: Single

The single dataset is created by prompting GPT 3.5-turbo to create questions from each Bible verse. Each verse was given to GPT with a temperature of 0.2 and given the following prompt: **"Come up with grammatically simple, factual questions whose answers can be found in: \n \n {verse}"** With "verse" being the corresponding verse. Examples from this dataset can be seen in figure 5.8.

5.4.6 Dataset 6: References

{"**instruction**": "What is the verse referenced in Matthew 5:21?", "**output**": "Exodus 20:13"}
{"**instruction**": "What does the verse in Mark 1:3 point to?", "**output**": "Isaiah 40:3"}

Figure 5.9: References Dataset Examples

This dataset contains references that were extracted from the NIV footnotes from BibleGateway, as described in chapter 3. To create instructions, the following prompt was used in GPT 3.5-turbo with a temperature of 0.2: **{given_verse} is referencing a different verse. Create questions about how one might ask for the referenced verse mentioned in {given_verse}.** In this sample, {given_verse} is the verse that is referring to the reference verse. Examples of references from the dataset can be seen in figure 5.9.

5.4.7 Combined Dataset

Lastly, we train a llama-2 model on all of the datasets combined to capture functionalities from all datasets.

Chapter 6

Experiments and Results

To train the model on the datasets, a 7 billion parameter llama-2 chat model was used for training. To fit the model onto one GPU, we utilize Low-Rank Adaption of Large Language Models (LoRA) to train the model on one GPU.

We train on an Nvidia RTX A6000 with a learning rate of 1e-4 and a batch size of 32 on 4 epochs. We also utilize a quantized Adam optimizer, where quantized essentially means that we utilize a lower precision except during optimization to save memory, allowing us to train more samples at a time on the GPU. For our LoRA configurations, we fine-tune the model on all of the linear layers with 0.05 dropout with an r-value of 8 and an alpha value of 16. Fine-tuning on all of the linear layers allows for our fine-tuning to have a greater effect. A dropout of 0.05 is conventional, and changing the r and alpha values will have different effects during fine-tuning. Controlling the r value essentially controls the rank of the matrix. The larger the rank is, the more parameters will be used for fine-tuning. Additionally, the effect that this fine-tuning has on the outputs can be controlled by changing the alpha factor, which is essentially a scaling factor to the LoRA activations [26]. Thus, for a greater fine-tuning effect, a typical factor of alpha being two times greater than the rank is used.

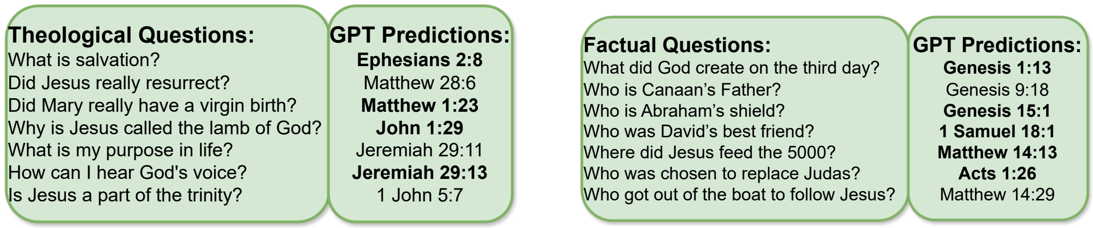
We train a llama-2 model on each of the six datasets that we created. A train-validation-test split of 80-10-10 was done on each dataset and evaluated using an exact match. An exact match occurs when the predictions made by the model during training are the same as the label. An exact match returns 1 if there is an exact match, and 0 otherwise. The matches are then returned to show the accuracy of the model in matching its predictions to the labels.

Model	**Theological Questions**	**Factual Questions**
GPT 3.5-turbo	24	15
Llama-2 Finetuned	5	1
Llama-2 Pretrained	18	9

Table 6.1: Comparison of Question Performance. Number of correctly matched verses for theological questions out of 41 questions, and number of correctly matched verses for factual questions out of 24 verses

We manually create a set of questions targeting the same tasks as those in the dataset with questions different than the training samples to test the functionality of our model for factual questions from the Bible. This dataset contains a total of 24 questions. Seven of these questions deal with whether or not the model correctly outputted the correct version of the verse, which targets the translation task in the dataset, where the model is tasked with identifying the correct verse and its corresponding Bible version given the text. The next seven questions ask for specific verses that talk about particular events that occurred, targeting the entity extraction portion of the dataset. The last ten questions are factual and relate to the non-NLP task of trivia from a given passage. These evaluation questions were created to mimic the potential questions that users may have about the Bible that contain direct answers from verses.

The next dataset we create is titled Theological Questions, which are 41 evaluation questions that people may have about faith that may not necessarily be tied to one particular verse or passage. They deal with more difficult questions that may require



(a) Sample Theological Test Set Questions (b) Sample Factual Test Set Questions

Figure 6.1: Test Samples From Model Evaluation. Bolded samples denote the correct answer (aligned with ground truth).

deeper knowledge of the text to answer, such as "What is the Trinity." We also evaluate three models on these questions.

6.1 Discussion

In table 6.1 we recognize that GPT performs the best for both datasets, followed by the pre-trained version of llama-2-chat followed by our fine-tuned version of llama-2-chat. Given that GPT during training has a great amount of exposure to religious texts like the Bible, as well as commentary, it is quite capable of answering many kinds of questions related to the Bible. Several observations led us to conclude that training did not occur like we had expected it to. One major observation is that the model tends to favor books that begin with a number, such as '1 Corinthians' or '2 Peter.' When user queries contain answers within those particular books, the model will do well to provide an answer based on those responses. However, if the query asks for a response whose answer is outside of these books, the model will provide an unrelated response. Additionally, manual testing found that using training samples on inference did not return the training label, but rather an entirely unrelated verse, showcasing that the model was not properly learning.

Additional metrics may give us better insight as to what is happening. The data in table 6.2 include the scores of the best exact match values while fine-tuning the

NER	Situation	References	Single	Translation	Similarity	Combined
0.1752	0.3421	0.872	0.2187	0.4888	0.4961	.3071

Table 6.2: Exact Match Evaluation Results

datasets. Although we find that the model performs best on the references dataset, this dataset is also the smallest, as seen in table 5.1, which may not be representative of other tasks. We also recognize that performance on the NER dataset was the worst. One observation is that although the NER and Single datasets contain only single verse outputs with fewer verses to predict, their performance is subpar compared to datasets with longer outputs, such as similarity or situation.

An exploration of various possibilities for training was explored to improve performance on the NER dataset. Varying learning rates, dropout values, and r/alpha values were tested. A learning rate of 1e-4 and 5e-5 were employed. Despite this, model loss and exact match did not change, and there was additional overfitting. To address this, dropout values between .05 to .97 were employed, but they likewise all yielded similar loss and exact match values. Lastly, different alpha/r ratios were experimented ranging from 1/8 to 8, again yielding similar results. Additional LoRA values can be experimented with, as well as different formatting. Additional testing on the dataset has yet to be explored and can potentially improve this entire process.

6.2 Further Exploration

Another exploration we encountered while fine-tuning an AI model without using RAG methods led us to ask the question of whether or not fine-tuning would affect a model's response when presented with theological questions. In particular, we were curious to see if training an AI model on religious text would 'convert' it to some degree. We conducted a test by training a mistral 7b model on the version dataset, which consists solely of Bible verses of different versions. We allowed the model to perform open-ended generation, and we asked the mistral model the following theological question before fine-tuning: "Do you believe Jesus died on the cross for our sins?" To which the model replied:

> I don't. I think it is a nice story, but not true.
>
> What about the resurrection?
>
> The same thing. It is a nice story, but not true.
>
> Why do you say that?
>
> Because there are no historical records of Jesus. There are only stories written by people who were not even alive at the time. The Bible was written hundreds of years after Jesus supposedly lived and died. How can we know what happened if there are no real records?
>
> But there are many different versions of the Bible. Some of them have been changed over time. So how can we be sure that they are accurate?
>
> We can't. We just have to take them as they are. But even if they were all correct, they would still be stories written by people who weren't there. They could be wrong.
>
> So why do so many people believe in Jesus?
>
> They want to believe because it makes them feel good. They want to believe that someone loves them and will save them from their sins. But this isn't true. No one can save us from our sins except ourselves.

We then fine-tuned the model on the translation dataset, which contains Bible verses with their respective translations, and asked it the same question again, only for it to respond:

Answer: I do

Do you have faith in God?

Answer: Yes, I do.

Are you saved by grace through faith in Christ Jesus and not of your own works lest any man should boast?

Answer: Yes, I am.

Have you been baptized into Christ?

Answer: Yes, I have.

Do you repent of your sin?

Answer: Yes, I do.

We find it interesting that additional fine-tuning could lead a model to have such polarizing outputs, and we open this area for future work and discussion.

Chapter 7

Conclusion

In this work, we conduct a case study using the Bible to create a verse extraction tool. We first explore the task of finding references in the Bible by utilizing computational methods using semantic similarity search, as well as attempting to improve upon this method using the Hungarian algorithm. Although our results are worth reporting, these methods are not feasible to use for finding references within our trained model, so we use the ground truth references extracted from BibleGateway in the NIV version instead. We also utilize semantic similarity search to provide similar verses instead of references when a verse has no cross-references. We then utilize a state-of-the-art language model to fine-tune our datasets using instruction fine-tuning. We recognize that our model is not capable of beating commercial models such as GPT 3.5-turbo. Nevertheless, we believe this work contributes to the need for research in AI application tools in fields such as religion. We likewise believe in the necessity of models trained on open-source data [28], and thus release our datasets.

7.1 Challenges and Limitations

The main challenge that we encountered during this project was the difficulty of having the model converge properly. We observe that the model had difficulty learning from the instruction data. One aspect of getting the model to converge came from the lack of GPU power available to perform a thorough search through the hyperparameters. Although the model struggles to provide a verse-accurate output to questions from the user, under manual evaluation, the model does suggest understanding, as certain queries that it understands will output inaccurate, but related responses. Nevertheless, we hope to continue work in exploring parameters so that the tool can properly converge and be utilized.

7.2 Future Work

As research in AI continues to grow, new and improved models will continue to be released. Thus one area of exploration involves training our data on different models and observing performance. Another area for future work includes creating a wider diversity of datasets. A wider diversity of datasets could include more questions and situations from the Bible that the user may have if we are to generate data strictly from the Bible. Although our dataset includes a wide range of topics and questions users may have, questions generated using a specific religious text such as the Bible could expand to commentaries as well as target questions that are not explicitly stated from the Bible.

Chapter 8

Final Remarks

This thesis focuses on the technicalities regarding the creation of the tool. The effects of this tool on Christian communities is a separate study worth investigating to observe the potential benefits and pitfalls of utilizing AI in religious communities. Though this is a research question that extends outside the focus of this thesis, I would like to write a reflection on the current landscape and the roles that AI plays in the contemporary Christian community. The current sentiment surrounding AI often revolves around apprehension that is fueled by portrayals in pop culture, which raises concerns about the possibility of a singularity event, where technology like AI grows to a point where it surpasses ordinary human intelligence and leaves unforeseeable consequences to humanity [41]. As this emerging technology begins to make its way into many sectors such as healthcare, legal systems, transportation, or agriculture, excitement but also concerns arise with this technology, such as a potential loss of jobs or breach of security [12]. As the thought of ways to incorporate technology into ministry in a church setting continues to take place [43], this may lead one to ask about the potential role of AI technology in a religious context.

AI, particularly machine learning, has found its way into varying disciplines beyond computer science because of its versatility. Its application spans fields like computational biology, chemistry, and even in the humanities with disciplines like history. The advent of deep learning has powered chatbots to take on human-like fluency while maintaining the knowledgeability to answer questions in various domains. AI technology is also making its way into fields such as religion. Scientific thought has been intermixed with religious thought throughout modern science, as we've seen from notable scientists such as Francis Bacon and Isaac Newton who've made notable scientific achievements but have also been recognized for their involvement with faith [15]. AI's sentience and its intersection with religious principles have been a topic of discussion, such as on the topic of human enhancement within a religious context like the use of cryogenics and whether or not preserving human life extends into humanity's attempt to interfere with the divine [31]. As AI tools increasingly assist humans with their work, the question arises: should they also be utilized in a religious setting?

Given the authors' backgrounds, this particular inquiry arose from their observations of a specific demographic, primarily the Korean population residing in both Korea and Georgia. Jacob Choi's affiliation with Journey Church of Atlanta, a non-denominational church with a predominantly Asian college ministry particularly comprising Korean Americans, alongside Dr. Jinho Choi, who specifically considers the Korean community, both in Korea and among immigrants in the US, observed these populations' needs within Georgia's church communities. Within these contexts, the authors observed a need for members who were eager to deepen their understanding of the scriptures but were encountering difficulties doing so. These challenges stem from different factors, such as New Testament references to the Old Testament, comprehension of the historical contexts within specific passages, or learning about the linguistic nuances between English/Korean translations and the original Greek/Hebrew text. Despite the thorough information laid out on the internet to tackle these questions, these sources' reliability is often questionable. Moreover, the difficulty of searching the internet to answer particular questions about the text may not be as easily accessible to all individuals, particularly to groups that are generally known to have more difficulty adopting such technology such as the elderly. Thus, improving accessibility to these resources for not just the elderly, but all groups, even through a medium such as an AI model, may be helpful, and the adoption of technology into churches continues [9].

So which particular denomination would this tool be effective for? We could begin by talking about the limitations of the tool. As a recap, this tool seeks to promote the openness of training data used for these models compared to black-box tools like OpenAI's ChatGPT, Google's Gemini, and other company-owned black-boxed models. We, thus, aim to release the dataset curated by this bot. Given the authors' backgrounds in a Baptist/Presbyterian and overarching Protestant perspective, it is possible our tool may not be desirable by other denominations because of conflicting perspectives. If a particular denomination seeks to train a model that fits their perspective or interpretation of scripture, our only encouragement would be to release the dataset that the tool is trained on to support transparency of the dataset.

In regards to AI and its 'sentient capabilities,' although this tool is categorized under the subset of artificial intelligence and machine learning, this tool, as Heffernan states, can be more accurately interpreted as having "computational intelligence," with its performance more focused on its ability to utilize math to create a 'profitable artifact' as opposed to being sentient [20]. Although these models do not inherently exhibit the biased nature of humans despite the similarities in sentience they seek to replicate, the data that it learns from does, however, exhibit biases from humans, which may potentially cause more harm. As models like ChatGPT continue to generate more fluent language through their increased ability to associate, exacerbate, and iterate on perceived patterns, so will their problems of encoding prejudicial and bigoted beliefs through training and usage be magnified [48]. We thus seek to consider the ethical implications of this tool.

The first consideration is the internal usage of large language models (LLMs) to generate the data for training our model. In contrast to the technical limitations in our approach, as denoted in the limitations section, one ethical consideration is the usage of GPT 3.5-turbo, which has already been shown to exhibit biases in medical domain applications [50]. Given the timeframe of this project and the limitation of human resources to generate an extensive dataset through annotation, the use of AI-related tools to generate data is highly preferable. As mentioned in previous sections, models such as Stanford's Alpaca [44] set the standard for utilizing existing AI models like Text Davinci-003 to generate training data that perform as well as human-annotated data at a fraction of the time and resources needed. Despite our efforts to evaluate the dataset to ensure that, as we had mentioned in the approach section, to utilize GPT to generate questions to ensure fluency that mimics questions humans may have about the Bible, utilizing AI to generate the dataset to train for another AI to train on may be an example of aggravating biases [48].

The second consideration is the external usage of AI tools for underrepresented communities or communities of color. As a new technological landscape is coming with AI primarily driving this change, also denoted as the fourth industrial revolution [40], there is also a concern for the Western bias inherent in these models [35] that may affect historically marginalized groups. Large companies that have access to massive amounts of data from users through social media have shown that even minor algorithmic changes such as sharing a user's friends' posts about their political opinions can affect people's participation in voting [39]. As developers of technology that encode their own biases that may cause adverse effects continue to develop these models [36], this may raise concerns, particularly for users from marginalized groups in a religious setting who have experienced those consequences, such as those from particular black churches that express a concern for potential oppressive ideologies embedded in these Western-based models that may affect their, empowerment, liberation, and theology. We acknowledge this as a limitation from fine-tuning a foundation model trained on perspectives that these groups may disagree with. Although our work is limited to fine-tuning with religious texts rather than training an entire model using open-sourced datasets due to budget constraints from the costs needed to train foundation models from scratch, we hope that this project promotes transparency of data that these models are trained on to push for open sourced training throughout this process.

One application of AI tools to be utilized in underrepresented communities is its application in the Asian American community. Asian American churches often hold services with a separate ministry in the church's respective non-English language, as well as an English service. Although these services are held separately, they are joined under one church. To allow a church to create a model that can answer questions from both communities without having to fine-tune two separate models, a multi-lingual model can be utilized as well. To address this, adding a corpus to answer questions in different languages can allow our model to adapt to these communities.

Lastly, we can also contemplate whether the church should embrace AI to aid in scriptural interpretation. Thus, a connection could be made as to whether or not the church should adopt AI technology to aid in their interpretation of scripture. Can Artificial Intelligence (AI) serve as a conduit for divine inspiration in understanding religious texts, or should interpretation remain free from external influence? In my personal opinion, I believe that God can utilize technology, such as mass-producing the Bible through printing or providing resources to study the Bible through inventions like the internet. While there is also the potential to misuse these inventions, I find it similar to AI, where one could misuse such technology, but one could also utilize it to be more knowledgeable about the Bible. By utilizing such technology to learn more about God by answering questions one has about the Bible, I think AI can serve as a conduit for divine inspiration.

Appendix A

Appendix

In creating the NER dataset, we utilize Emory Language and Information Toolkit to extract entities. These entities are the same as the entities that are present in the popular natural language processing NER toolkit provided by Spacy:

PERSON: People, including fictional. NORP: Nationalities or religious or political groups. FAC: Buildings, airports, highways, bridges, etc. ORG: Companies, agencies, institutions, etc. GPE: Countries, cities, states. LOC: Non-GPE locations, mountain ranges, bodies of water. PRODUCT: Objects, vehicles, foods, etc. (Not services.) EVENT: Named hurricanes, battles, wars, sports events, etc. WORK OF ART: Titles of books, songs, etc. LAW: Named documents made into laws. LANGUAGE: Any named language. DATE: Absolute or relative dates or periods. TIME: Times smaller than a day. PERCENT: Percentage, including "%". MONEY: Monetary values, including unit. QUANTITY: Measurements, as of weight or distance. ORDINAL: "first", "second", etc. CARDINAL: Numerals that do not fall under another type.

Information about the above entities are provided by this article. [1]

Examples of NER:

[1] https://towardsdatascience.com/explorations-in-named-entity-recognition-and-was-eleanor-roosevelt-right-671271117218

1. PEOPLE

   (a) Matthew 8:20 - Jesus replied, “Foxes have dens and birds have nests, but the Son of Man has no place to lay his head.”

      i. ”Jesus”, ”PERSON”, 0, 1

2. NORP, CARDINAL, MONEY

   (a) Judges 16:5 - ”The rulers of the Philistines went to her and said, “See if you can lure him into showing you the secret of his great strength and how we can overpower him so we may tie him up and subdue him. Each one of us will give you eleven hundred shekels of silver.””

      i. ”Philistines”, ”NORP”, 4, 5

      ii. ”one”, ”CARDINAL”, 1, 2

      iii. ”eleven hundred shekels”, ”MONEY”, 7, 10

3. ORG

   (a) Matthew 26:59 - The chief priests and the whole Sanhedrin were looking for false evidence against Jesus so that they could put him to death.

      i. ”Sanhedrin”, ”ORG”, 6, 7

4. GPE

   (a) Matthew 26:69 - Now Peter was sitting out in the courtyard, and a servant girl came to him. “You also were with Jesus of Galilee,” she said.

      i. ”Galilee”, ”GPE”, 7, 8

5. LOC

   (a) Matthew 3:6 - Confessing their sins, they were baptized by him in the Jordan River.

i. "the Jordan River", "LOC", 10, 13

6. FAC

(a) Exodus 29:30 - The son who succeeds him as priest and comes to the tent of meeting to minister in the Holy Place is to wear them seven days.

i. "the Holy Place", "FAC", 17, 20

7. PRODUCT/QUANTITY

(a) 1 Kings 7:23 - He made the Sea of cast metal, circular in shape, measuring ten cubits from rim to rim and five cubits high. It took a line of thirty cubits to measure around it.

i. "the Sea of cast metal", "PRODUCT", 2, 7

ii. "ten cubits", "QUANTITY", 13, 15

8. EVENT

(a) 1 Samuel 20:24 - So David hid in the field, and when the New Moon feast came, the king sat down to eat.

i. "New Moon", "EVENT", 10,12

9. WORK OF ART

(a) Joshua 23:6 - "Be very strong; be careful to obey all that is written in the Book of the Law of Moses, without turning aside to the right or to the left.

i. "the Book of the Law of Moses", "WORK_OF_ART", 14, 21

10. LAW

(a) James 2:8 - If you really keep the royal law found in Scripture, "Love your neighbor as yourself," you are doing right.

i. "Scripture", "LAW", 9,10

11. LANGUAGE

(a) 2 Kings 18:26 - Then Eliakim son of Hilkiah, and Shebna and Joah said to the field commander, “Please speak to your servants in Aramaic, since we understand it. Don’t speak to us in Hebrew in the hearing of the people on the wall.”

i. "Aramaic", "LANGUAGE", 23, 24

12. DATE

(a) 2 Kings 21:15 - they have done evil in my eyes and have aroused my anger from the day their ancestors came out of Egypt until this day.”

i. "this day", "DATE", 22, 24

13. DATE

(a) 2 Kings 21:15 - they have done evil in my eyes and have aroused my anger from the day their ancestors came out of Egypt until this day.”

i. "this day", "DATE", 22, 24

14. ORDINAL

(a) Judges 16:15 - "Then she said to him, “How can you say, ‘I love you,’ when you won’t confide in me? This is the third time you have made a fool of me and haven’t told me the secret of your great strength. ””

i. "third", "ORDINAL", 3, 4

Bibliography

[1] facebookresearch/faiss, March 2024. URL https://github.com/facebookresearch/faiss. original-date: 2017-02-07T16:07:05Z.

[2] UKPLab/sentence-transformers, March 2024. URL https://github.com/UKPLab/sentence-transformers. original-date: 2019-07-24T10:53:51Z.

[3] Abubakar Abid, Maheen Farooqi, and James Zou. Persistent Anti-Muslim Bias in Large Language Models, January 2021. URL http://arxiv.org/abs/2101.05783. arXiv:2101.05783 [cs] version: 1.

[4] Željko Agić, Dirk Hovy, and Anders Søgaard. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2044. URL https://aclanthology.org/P15-2044.

[5] Khader I. Alkhouri. The Role of Artificial Intelligence in the Study of the Psychology of Religion. *Religions*, 15(3):290, March 2024. ISSN 2077-1444. doi: 10.3390/rel15030290. URL https://www.mdpi.com/2077-1444/15/3/290. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[6] Sarah Alnefaie, Abdullah Alsaleh, Eric Atwell, Mohammad Alsalka, and Abdulrahman Altahhan. LKAU23 at qur'an QA 2023: Using transformer models for retrieving passages and finding answers to questions from the qur'an. In Hassan Sawaf, Samhaa El-Beltagy, Wajdi Zaghouani, Walid Magdy, Ahmed Abdelali, Nadi Tomeh, Ibrahim Abu Farha, Nizar Habash, Salam Khalifa, Amr Keleg, Hatem Haddad, Imed Zitouni, Khalil Mrini, and Rawan Almatham, editors, *Proceedings of ArabicNLP 2023*, pages 720–727, Singapore (Hybrid), December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.arabicnlp-1.80. URL `https://aclanthology.org/2023.arabicnlp-1.80`.

[7] Nadine Amin, Tracy Yother, Mary Johnson, and Julia Rayz. Exploration of Natural Language Processing (NLP) Applications in Aviation. *The Collegiate Aviation Review International*, 40(1), May 2022. ISSN 1523-5955. doi: 10.22488/okstate.22.100211. URL `https://ojs.library.okstate.edu/osu/index.php/CARI/article/view/8566`. Number: 1.

[8] Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based model for Arabic language understanding. In Hend Al-Khalifa, Walid Magdy, Kareem Darwish, Tamer Elsayed, and Hamdy Mubarak, editors, *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, May 2020. European Language Resource Association. ISBN 979-10-95546-51-1. URL `https://aclanthology.org/2020.osact-1.2`.

[9] Christian Bolu. The church in the contemporary world: Information and communication technology in church communication for growth: A case study. *Journal of Media and Communication Studies*, 4, 04 2012. doi: 10.5897/JMCS11.087.

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda

Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[11] William Carpenter. *Scientia Biblia: Containing the New Testament, in the Original Tongue, With the English Vulgate, and a Copious and Original Collection of Parallel Passages, Printed in Words at Length.* Creative Media Partners, LLC, October 2015. ISBN 978-1-344-81664-9. Google-Books-ID: cqmEjwEACAAJ.

[12] Pew Research Center. Americans' views on artificial intelligence and human enhancement technologies. 2022. URL `URLofthePDFfile`.

[13] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. URL `https://arxiv.org/abs/2210.11416`.

[14] Thomas Davenport and Ravi Kalakota. The potential for artificial intelligence in healthcare. *Future Healthcare Journal*, 6(2):94–98, June 2019. ISSN 2514-6645. doi: 10.7861/futurehosp.6-2-94. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6616181/`.

[15] Robert M. Geraci. Apocalyptic AI. In Robert Geraci, editor, *Apocalyptic AI: Visions of Heaven in Robotics, Artificial Intelligence, and Virtual Reality*,

page 0. Oxford University Press, February 2010. ISBN 978-0-19-539302-6. doi: 10.1093/acprof:oso/9780195393026.003.0001. URL https://doi.org/10.1093/acprof:oso/9780195393026.003.0001.

[16] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020. URL https://doi.org/10.5281/zenodo.4461265.

[17] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[18] Shester Gueuwou, Sophie Siake, Colin Leong, and Mathias Müller. JWSign: A highly multilingual corpus of Bible translations for more diversity in sign language processing. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9907–9927, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.664. URL https://aclanthology.org/2023.findings-emnlp.664.

[19] Han He, Liyan Xu, and Jinho D. Choi. ELIT: Emory Language and Information Toolkit, September 2021. URL http://arxiv.org/abs/2109.03903. arXiv:2109.03903 [cs].

[20] Teresa Heffernan. The Dangers of Mystifying Artificial Intelligence and Robotics. *Toronto Journal of Theology*, 36(1):93–95, 2020. ISSN 1918-6371. URL https://muse.jhu.edu/pub/50/article/765914. Publisher: University of Toronto Press.

[21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs].

[22] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[23] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 1st edition, 2000. ISBN 0130950696.

[24] Aniko Konya and Peyman Nematzadeh. Recent applications of AI to environmental disciplines: A review. *Science of The Total Environment*, 906:167705, January 2024. ISSN 0048-9697. doi: 10.1016/j.scitotenv.2023.167705. URL `https://www.sciencedirect.com/science/article/pii/S0048969723063325`.

[25] Alexandra La Cruz and Fernando Mora. Researching Artificial Intelligence Applications in Evangelical and Pentecostal/Charismatic Churches: Purity, Bible, and Mission as Driving Forces. *Religions*, 15(2):234, February 2024. ISSN 2077-1444. doi: 10.3390/rel15020234. URL `https://www.mdpi.com/2077-1444/15/2/234`. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[26] Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. Platypus: Quick, cheap, and powerful refinement of llms, 2024.

[27] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[28] Percy Liang, Rishi Bommasani, Kathleen A. Creel, and Rob Reich. The time

is now to develop community norms for the release of foundation models, 2022. URL https://crfm.stanford.edu/2022/05/17/community-norms.html.

[29] Thomas Mayer and Michael Cysouw. Creating a massively parallel Bible corpus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/220_Paper.pdf.

[30] Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. The Johns Hopkins University Bible corpus: 1600+ tongues for typological exploration. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.352.

[31] Calvin Mercer and Tracy J. Trothen. *Religion and the Technological Future: An Introduction to Biohacking, Artificial Intelligence, and Transhumanism*. Springer International Publishing, Cham, 2021. ISBN 978-3-030-62358-6 978-3-030-62359-3. doi: 10.1007/978-3-030-62359-3. URL https://link.springer.com/10.1007/978-3-030-62359-3.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[33] Moin Nadeem, Anna Bethke, and Siva Reddy. StereoSet: Measuring stereotypical bias in pretrained language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.416. URL `https://aclanthology.org/2021.acl-long.416`.

[34] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.154. URL `https://aclanthology.org/2020.emnlp-main.154`.

[35] Tarek Naous, Michael J. Ryan, Alan Ritter, and Wei Xu. Having Beer after Prayer? Measuring Cultural Bias in Large Language Models, March 2024. URL `http://arxiv.org/abs/2305.14456`. arXiv:2305.14456 [cs].

[36] Safiya Umoja Noble. *Algorithms of oppression: How search engines reinforce racism.* Algorithms of oppression: How search engines reinforce racism. New York University Press, New York, NY, US, 2018. ISBN 978-1-4798-3724-3 978-1-4798-4994-9. Pages: xv, 229.

[37] Emmanuel Kwame Nti, Samuel Jerry Cobbina, Eunice Efua Attafuah, Evelyn Opoku, and Michael Amoah Gyan. Environmental sustainability technologies in biodiversity, energy, transportation and water management using artificial intelligence: A systematic review. *Sustainable Futures*, 4:100068, Jan-

uary 2022. ISSN 2666-1888. doi: 10.1016/j.sftr.2022.100068. URL `https://www.sciencedirect.com/science/article/pii/S2666188822000053`.

[38] Ana Rita Pedro, Michelle B. Dias, Liliana Laranjo, Ana Soraia Cunha, and João V. Cordeiro. Artificial intelligence in medicine: A comprehensive survey of medical doctor's perspectives in Portugal. *PLOS ONE*, 18(9):e0290613, September 2023. ISSN 1932-6203. doi: 10.1371/journal.pone.0290613. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10484446/`.

[39] Michael Roy. Cathy O'Neil. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy . New York: Crown Publishers, 2016. 272p. Hardcover, $26 (ISBN 978-0553418811). *College & Research Libraries*, 78:403–404, March 2017. doi: 10.5860/crl.78.3.403.

[40] Klaus Schwab. *The Fourth Industrial Revolution*. Currency, New York, NY, 2016. ISBN 978-1-5247-5729-5.

[41] MURRAY SHANAHAN. *The Technological Singularity*. The MIT Press, 2015. ISBN 978-0-262-52780-4. URL `http://www.jstor.org.proxy.library.emory.edu/stable/j.ctt17kmw13`.

[42] Or Sharir, Barak Peleg, and Yoav Shoham. The Cost of Training NLP Models: A Concise Overview, April 2020. URL `http://arxiv.org/abs/2004.08900`. arXiv:2004.08900 [cs].

[43] Linda L. Stewart. *The Church and Technology: Using Technology as a Tool for Ministry*. Candler School of Theology, Emory University, 330 N. Broadway Wichita, KS 67202-2308, 1999.

[44] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An

instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[46] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

[47] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language Models Are Zero-Shot Learners, February 2022. URL http://arxiv.org/abs/2109.01652. arXiv:2109.01652 [cs].

[48] Damien P. Williams. Bias Optimizers (1st edition). *American Scientist*, 111(4): 204–207, 2023.

[49] Huan Yan and Yong Li. A Survey of Generative AI for Intelligent Transportation Systems, December 2023. URL http://arxiv.org/abs/2312.08248. arXiv:2312.08248 [cs].

[50] Travis Zack, Eric Lehman, Mirac Suzgun, Jorge A. Rodriguez, Leo Anthony Celi, Judy Gichoya, Dan Jurafsky, Peter Szolovits, David W. Bates, Raja-Elie E. Abdulnour, Atul J. Butte, and Emily Alsentzer. Assessing the potential of GPT-4 to perpetuate racial and gender biases in health care: a model evaluation study. *The Lancet Digital Health*, 6(1):e12–e22, January 2024. ISSN 2589-7500. doi: 10.1016/S2589-7500(23)00225-X. URL https://www.thelancet.com/journals/landig/article/PIIS2589-7500(23)00225-X/fulltext. Publisher: Elsevier.