

Social Media Text Analytics

Using Large Language Models

By:

Gurkaran Singh Sran

Supervised By:

Prof. Yuri Lawryshyn

Department of Mechanical and Industrial Engineering

University of Toronto

Abstract

This report pertains to a project undertaken for a client which is an investment company. The client required a solution that would analyze public opinions, comments, and questions about a topic, particularly financial products and/or services from a database of social media posts. This solution would then distill down posts, often thousands in number, into useful insights and summaries about the top benefits people are discussing about the product or service, its limitations, and the uncertainties people have. Although multiple Natural Language Processing techniques could be adopted to develop this solution, the scope of this project involves developing the above-discussed solution using Large Language Models (LLMs). An important characteristic of an LLM is its context window, the maximum number of tokens (words or units of text) it takes as input, influencing its response. However, the context window of LLMs available to date is limited in size and often fails to accommodate most real-world datasets when fed in their entirety. Hence, there arises a need to devise a solution that can take in an entire dataset but feeds only those segments of the dataset to the large language model that are most relevant to the prompt given to the model, and at the same time fit within the context window of the model. This project is primarily focused on developing a solution using a technique known as Retrieval Augmented Generation (RAG) which is detailed in this report.

Acknowledgments

I would like to express my sincere gratitude to everyone at CMTE group for their valuable support throughout the duration of the project. I am deeply indebted to my supervisor, Professor Yuri Lawryshyn whose guidance, support, and expertise have been invaluable throughout the research process. I extend my appreciation to Kyle Burgers, Yash Savla, and Mohamad Radwan for their mentorship and insightful feedback during our weekly meetings.

This research project would not have been possible without the collective support of these individuals, and for that, I am truly grateful.

Table of Contents

Abstract.....	2
Acknowledgments	3
Table of Figures.....	5
1. Introduction.....	6
1.1 The Problem.....	6
1.2 The Objective	7
1.3 Recommendations	9
2. Literature Review	10
3. Methodology.....	16
3.1 Data Pre-Processing and Segregation	16
3.2 Generation of Text-Embeddings	18
3.3 Preparing Training Examples for Fine-Tuning the Embedding Model.....	21
3.4 Fine-Tuning of Embedding Model.....	26
3.5 Validating Selected Embedding Model Using Test Dataset on an Unrelated Topic	29
3.6 Data Pipeline and Implementation.....	30
4. Results and Discussion	36
4.1 Impact of Input Context Size on the Summarization Generated by LLM	36
4.2 Analyzing Summarizations Generated without Fine-Tuning of Embedding Model..	42
4.3 Summarization on Positives of FHSA	44
4.4 Summarization on Negatives of FHSA	46
4.5 Summarization on Uncertainties about FHSA	47
5. Recommendations and Future Work	50
References	52

Table of Figures

Figure 1: Process flow for data pre-processing, transformation, and storage	17
Figure 2: Process flow for preparing dataset-3 and two-dimensional t-SNE plot of embeddings of dataset-3 generated by embedding model-1.....	20
Figure 3: t-SNE plots of embeddings of hypothetical training examples showing the effect of embedding model training on the embeddings of comments in the training examples.	22
Figure 4: Process flow of splitting labeled dataset-3 into training data and validation data.....	23
Figure 5: Hyperparameter Tuning for the fine-tuned embedding model.	28
Figure 6: Two-dimensional t-SNE plot of embeddings of dataset-4 generated by the selected embedding model.....	30
Figure 7: Comparison of retrieval performance on dataset-2 between embedding model-1 and embedding model-2.	32
Figure 8: Data Flow between different components of the solution.	34

1. Introduction

1.1 The Problem

It is common to want to understand public opinion on a particular subject. This includes what people consider beneficial or positive about the subject, what the limitations or disadvantages are, and what questions or uncertainties people have. This information can be valuable for businesses. For instance, if the subject is a financial service offered by multiple financial institutions, knowing the concerns and pain points of customers or potential customers can aid in creating a marketing strategy that addresses these issues. Knowing about the top queries that people have can help develop marketing material that clarifies these issues. It can now turn prospective clients into customers who were hesitant earlier because of lack of knowledge or due to misinformation. Similarly, for a business, knowing about people's thoughts of its products or services can be highly beneficial as this feedback can help the business focus on the areas where it needs to improve the most. Additionally, understanding people's needs by analyzing their opinions on a topic can help in the development of a brand-new product that caters to the most pressing demands. This information about people's views and questions on a topic can have countless applications beyond the ones discussed above.

One of the biggest sources of the above-discussed information is social media (Hou et al. (2020)) [1]. However, the sheer volume of such information makes it nearly impossible for humans to read it all. Thus, a business looking to tap into the potential of information in social media data needs a social media analytics solution designed for their specific use case. It is no surprise that leading companies like Apple, Google, Microsoft, Facebook, Samsung, and Amazon use social media analytics to improve corporate strategies and customer relations (Ducange et al. (2018)) [2].

Diverse methods of analyzing social media text data have been discussed by Hou et al. (2020) [1] and by Wang et. al (2016) [3]. These methods find their application in the areas of topic analysis, time series analysis, and sentiment analysis. However, these methods cannot be used in an application where it is required to generate summarization after understanding the information available in text data because these methods do not involve generating natural language and learning language representations (Young et al. (2018) [4], Peters et al. (2018) [5]). Such applications come under the domain of Generative Artificial Intelligence (Mlađan Jovanović and Campbell (2022) [6]). This domain has been an area of aggressive research involving the utilization of deep learning models for Natural Language Processing and generating natural language (Young et al. (2018) [4], Goldberg (2016) [7], Kim et al. (2016) [8]). A breakthrough was achieved in this research area when Vaswani et al. (2017) [9] proposed a neural network architecture called the Transformer, as this architecture was highly parallelizable and required significantly less time to train. This architecture then became a fundamental building block for Large Language Models (LLMs) (Devlin et al. (2019) [10], Radford et al. (2018) [11]).

This study is a proof-of-concept project which aims to demonstrate the use of LLMs in social media analytics to summarize public opinions on a topic, based on a text dataset comprised of social media comments.

1.2 The Objective

The objective of this study is to develop a Large Language Model solution that will process a readily available text database of social media posts on a topic, to retrieve the most relevant posts concerning a given prompt about public opinion on the topic, and generate a summarization on the topic as per the prompt as a set of unique points discussed in the posts.

The scope of the prompt mentioned in the above objectives is limited to asking about benefits or advantages related to the topic, limitations or disadvantages, and any questions or uncertainties that are being discussed on the topic in the social media comments within the given dataset.

Key contributions made in this work to meet the above objectives are:

- (i) An appropriate text embedding model is selected and is fine-tuned so that the embeddings generated by the fine-tuned model are such that in the high dimensional space of embedding vectors, the embeddings corresponding to posts concerning positives tend to be closer to each other and far from the posts concerning negatives and vice-versa.
- (ii) A data framework is created using the LlamaIndex Python module in conjunction with a vector database which contains all the social media comments from the dataset along with their text embeddings generated using the fine-tuned embedding model. This framework is called an 'index' that is designed to enable querying on the dataset by an LLM. This index created with a vector database is typically called a vector store index.
- (iii) A workflow is designed on the principles of a technique known as Retrieval Augmented Generation (RAG) which is discussed in detail in the Literature Review (Chapter 2). This workflow enables a semantic search to retrieve the most relevant documents concerning the query prompt which are then given to the LLM as a context.

1.3 Recommendations

Based on the findings obtained from this study, we recommend that for the NLP applications that involve providing an LLM with data from an external data source, Retrieval Augmented Generation (RAG) be used. This is because RAG addresses the problem of a limited context window in LLMs by retrieving only the relevant information from the data source and providing it to the LLM. We also highly recommend fine-tuning the text embedding model used to generate the embeddings of the text documents present in the external database, as fine-tuning significantly improves the quality of retrieval by preventing irrelevant information from being fed as context to the LLM. A lower usage of the context window is suggested, as opposed to using the context window completely, as it results in generated content that is highly aligned with the data given as context and content that has minimal redundancy.

The rest of this report is organized as follows. Chapter 2 incorporates a review of literature specifically focusing on the techniques and methods we have used in this study. Chapter 3 covers the workflow and methodology to achieve the objective set in Chapter 1. Chapter 4 presents the findings obtained by applying the proposed methodology and discusses those observations in detail. Lastly, in Chapter 5, we list key recommendations based on this study.

2. Literature Review

This section contains a comprehensive literature review of key components that form the building blocks of the LLM solution developed in this study. We begin with text embeddings which have become fundamental in many modern NLP systems and examine methodologies that enable fine-tuning these embeddings for domain-specific tasks. We then investigate Vector Databases and Vector Store Index which act as an external information storage. Following, we review Retrieval Augmented Generation (RAG), a paradigm that combines generation tasks with the power of information retrieval. Finally, we will explore studies that investigate how input context length impacts the performance of large language models.

Text Embeddings and Fine-tuning of Embedding Models

Text embeddings are numerical representations of pieces of text (sentences, paragraphs, or documents) as fixed-length real-valued vectors. Text embeddings are important because many NLP algorithms, including semantic search, require the input text documents to be represented as a fixed-length feature vector. Le et al. (2014) [12] emphasized that commonly used fixed-length features for text, such as bag-of-words, have two major weaknesses: They do not consider the order in which words are placed in a document, and they ignore the semantics of the words. Therefore, we need a model that gives similar vector representations of two sentences carrying the same meaning even if they consist of entirely different words. One of the breakthroughs in this field is the development of transformer-based models, such as BERT (Devlin et al. (2019) [10]) and GPT (Radford et al. (2018) [11]). Both BERT and GPT are language models that have been pre-trained in an unsupervised manner using a large corpus of text data, to learn language representations that enable them to understand language (Devlin et

al. (2019) [10], Radford et al. (2018) [11]). It is important to emphasize here that learning language representations is what enables these models to convert a sequence of words (a piece of text) into its corresponding numerical representation, i.e., text embedding, which is a prominent part of both models' architecture and is often termed as the encoding step. Therefore, once pre-trained, these models can generate similar embedding vectors for the sentences or paragraphs that convey similar meanings which is a fundamental requirement for our solution. Although any of the above-discussed models can be used to generate text embeddings for our solution, we move forward with a BERT-based embedding model called 'BAAI/bge-large-en-v1.5' because it is available as open-source.¹

Furthermore, Devlin et al. (2019) [10] very clearly highlighted the importance and process of fine-tuning embedding models for downstream tasks or domain-specific tasks. They repeatedly stress that unsupervised pre-training is what enables the language model to understand language, but supervised fine-tuning must be done to improve the model's performance on the tasks involving specific use cases. For our solution, we use a fine-tuned embedding model for the use case when we need to cluster social media comments based on whether the opinion in the comments is positive or negative. Additionally, we use an embedding model without fine-tuning for the retrieval of comments involving uncertainties (placed in a separate database) since there is no need to cluster these comments as per any criteria.

Vector Databases and Vector Store Index

A vector database is a type of database that is designed to efficiently store and retrieve vector data. This data is a collection of high-dimensional real-valued vectors which are mathematical

¹ <https://huggingface.co/BAAI/bge-large-en-v1.5/tree/main>

representations of characteristics or attributes. The dimension of all vectors in a vector database is the same but can range from tens to thousands. This dimension usually depends on the type and complexity of the problem at hand. The vectors inside a vector database are typically generated by some transformation or by using some embedding function on the raw data which can be text, audio, video, images, etc. The embedding functions can be machine learning models such as text embedding models, as discussed previously, or they can be any other feature extraction model. Han et al. (2023) [13] have done a comprehensive survey on vector databases and have discussed their function, strengths, and challenges in detail. Vector databases are particularly suited for tasks involving similarity search and retrieval because they use similarity metrics based on vector distances such as Euclidean distance, cosine similarity, correlation coefficient (Sondur et al. (2016) [14]). Thus, it is possible to find the topmost similar or relevant data point in a vector database concerning another data point or query which is a reason why these databases find their application in domains such as recommendation systems, Natural Language Processing (NLP), audio and image processing systems.

Finding the nearest neighbors concerning a data point is a fundamental problem that any vector database is designed to handle. Vector databases usually employ specialized indexing structures and algorithms to search and retrieve vectors based on their similarity scores efficiently. The term vector store index typically refers to an index which is a data structure that enhances the speed of data retrieval in vector databases. In our solution, we use a vector database service provided by Pinecone². Although the indexing strategy used by Pinecone is their proprietary information, however depending upon the vector database and the application several different types of indexing strategies can be employed such as Ball Tree (Kumar and Kumar (2015)

² <https://www.pinecone.io/>

[15]), KD Tree (Ghojogh et al. (2018) [16]), Local-Sensitive Hashing (Jafari et al. (2021) [17]) and Deep Hashing (Liu et al. (2016) [18]).

Retrieval Augmented Generation (RAG)

Neural language models learn in-depth knowledge from data during pre-training as a parameterized implicit knowledge base (Raffel et al. (2020) [19], Roberts et al. (2020) [20]). This knowledge can be called parametric memory. However, increasing or revising their memory is not straightforward and easy, and predictions based exclusively on this parametric memory can lead to “hallucinations” (Maynez et al. (2020) [21]). Although Guu et al. (2020) [22] and Karpukhin et al. (2020) [23] proposed models that combine parametric memory with retrieval-based memory (or non-parametric memory) that could address the issues mentioned above, they only focussed on Open-Domain Question Answering problems. Lewis et al. (2020) [24] introduced Retrieval Augmented Generation (RAG) models based on a general-purpose fine-tuning approach that could be applied to any downstream NLP task. RAG models consist of two components: Retriever containing knowledge base in the form of vector store index as non-parametric memory along with a pre-trained encoder (a pre-trained embedding model), and Generator (with pre-trained parametric memory), which is a sequence-to-sequence language model. These two components are then fine-tuned end-to-end on a domain-specific NLP task.

In our solution, we use a RAG approach with a small variation in that we do not fine-tune the generator (the language model) as we only need the generator to produce a summarization based on retrieved documents which can be done by any pre-trained language model without fine-tuning. Therefore, Retrieval Augmented Generation refers to a natural language processing approach that combines elements of both retrieval and generation models to

improve the performance of language understanding and generation tasks. This approach is especially useful in situations where a model requires information from a large external knowledge base, and where the knowledge base is required to be updated on a timely basis.

Impact of context length on performance of large language models

Recent developments have enabled the large language models to take longer input context, as high as 128K tokens (GPT-4-turbo by OpenAI). At the same time, it is crucial to know how well these language models use longer context. The approach of using higher context by providing the model with as much input context as possible and letting the model find out the required information may be appealing; however, research done to evaluate this approach has shown that using higher context comes at a cost. Liu et al. (2023) [25] evaluated various state-of-the-art language models' performance by placing relevant information at different locations within context input and repeating the experiments with different sizes of the input context length. They observed that the performance of all models decreased significantly when the relevant information was placed around the middle of the context while the performance was best when the relevant information was at the start of the context. They further observed that, for each position of the relevant information, the performance of all models decreased as the size of the input context was increased. According to Liu et al. (2023) [25], the performance of language models should not be significantly impacted by the position of relevant information if they can handle long input contexts well. However, they noted a significant impact of the position of relevant information in their experiments, indicating that the language models struggled with long input context.

Furthermore, Shi et al. (2023) [26] have shown that when irrelevant information is present in the context, the performance of the language models tends to deteriorate. Therefore, it is

recommended to place the relevant information close to the start of the context given to the model, and further to give fewer documents (or shorter context) as input, as opposed to stuffing the entire available context window of the model. Thus, we have used a retrieval system in our solution that generates context following both above-mentioned recommendations, as comments with higher similarity scores are placed first in the input context and further the system allows to retrieve top 'k' comments out of the complete dataset.

3. Methodology

It is worth re-emphasizing here that this study was a proof-of-concept project executed in accordance with the objectives outlined in Section 1.2. Thus, this study could be done on any topic on which public opinion-related data was readily available from social media. We decided to perform this study on First Home Savings Account³ (FHSA), primarily because this was seen as a favorable topic on which plenty of social media comments with contrasting opinions could be collected. The following sections in this chapter explain the methods that were used to develop a solution to meet the objectives outlined in section 1.2. A process flow concerning data pre-processing, transformation, and storage is illustrated in Figure 1, which is discussed in detail in the subsequent sections.

3.1 Data Pre-Processing and Segregation

The raw dataset, a CSV file named 'fhsa_120days_2023_v3.csv' in the supplementary files, used in this study was web-scraped and contained social media posts concerning the First Home Savings Account (FHSA). As shown in Figure 1, the raw dataset contained commentary on FHSA sourced from Reddit as well as news platforms, with 9456 records in total. The CSV file was imported into Python as a Pandas Data-Frame only using the columns named 'fullText' which contained the comments as text, and 'PageType' which contained the information about the source of text- Reddit or News. This information was used to remove the rows which were not sourced from Reddit. Thus, the processed dataset had social media comments only from Reddit which were 8553 in number as can be seen in Figure 1.

³ <https://www.canada.ca/en/revenue-agency/services/tax/individuals/topics/first-home-savings-account.html>

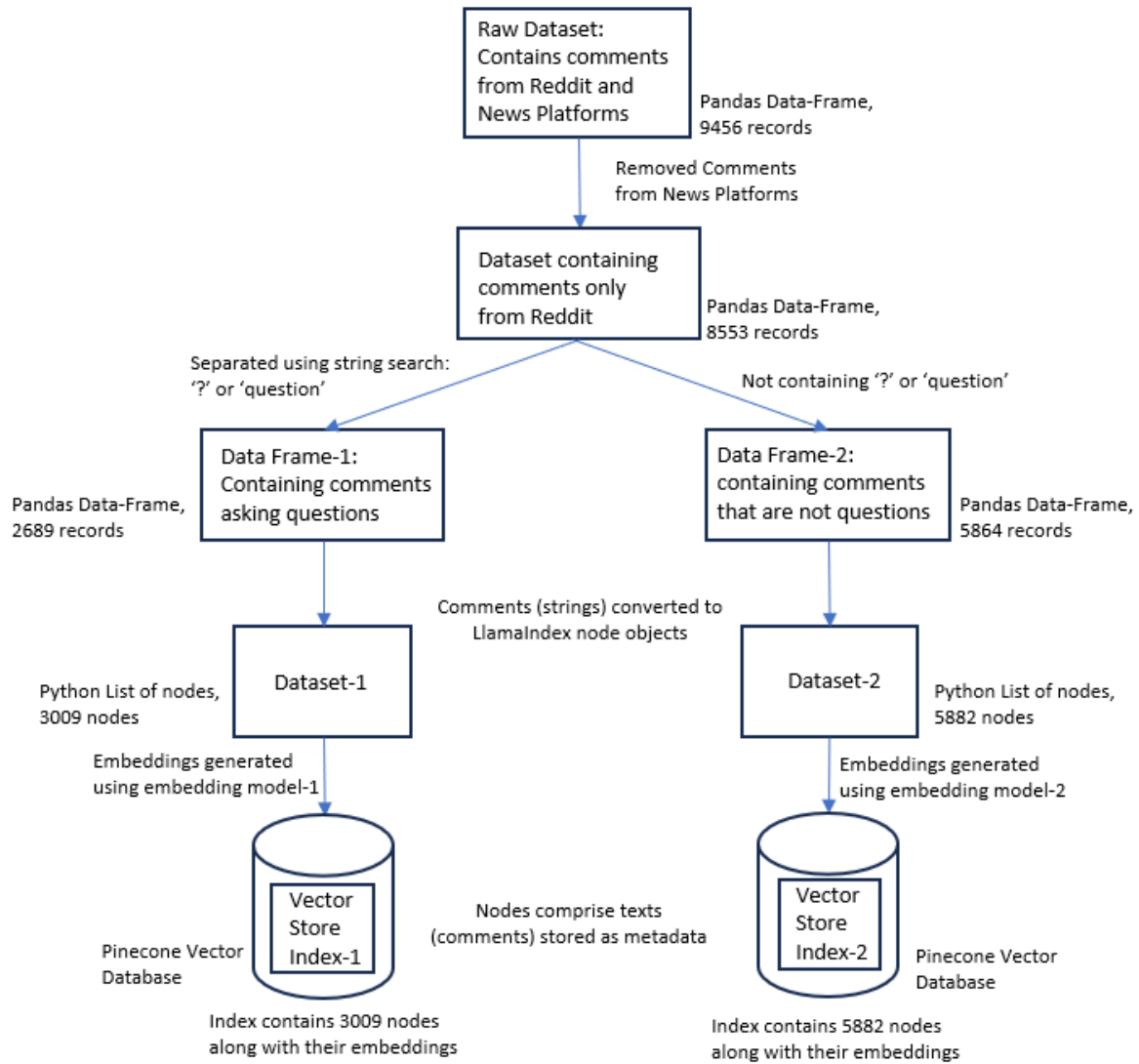


Figure 1: Process flow for data pre-processing, transformation, and storage

Next, a simple string search was employed to segregate the comments that contained the symbol ‘?’ or the word ‘question’ from the rest of the dataset, which resulted in two separate Pandas data frames – Data Frame-1 which contained the comments asking questions, and Data Frame-2 which contained the comments that were not questions as shown in Figure 1. Thereafter, the LlamaIndex Python module was used to convert the comments in each of the

above-discussed data frames into ‘nodes’, which is a term used in the module documentation⁴ to represent chunks of text documents (or comments in this case) that contain the information on the text content as well as how these text documents relate to each other. Typically, the number of nodes would equal the number of text documents unless a text document was large, in which case that document would result in two or more nodes. It should be noted here that the two or more nodes would contain information linking them to the original text document from which they were created. As mentioned in Figure 1, the collection of 3009 nodes which contained the social media comments asking questions would form an ideal dataset to be used for analysis related to uncertainties, hereafter called “dataset-1” in this report. The collection of 5882 nodes which did not contain the comments that were questions would form a dataset on which opinion-related analysis could be done and would hereafter be called “dataset-2”.

3.2 Generation of Text-Embeddings

Once the data was pre-processed and segregated into two separate datasets (dataset-1 and dataset-2) as discussed above, the next step was to build a knowledge base in the form of a vector store index. This was to be done for both the datasets: dataset-1 and dataset-2 separately. For the solution developed in this study, we employed vector databases provided by Pinecone as discussed in the Literature Review (Chapter 2). Pinecone uses an indexing technique called vector indexing. In vector databases, indexing is critical for the efficient retrieval of similar vectors (or embeddings in this case). Pinecone employs an index structure that organizes vectors in a way that facilitates fast nearest neighbour searches. We call this structure a vector store index.

⁴ <https://docs.llamaindex.ai/en/stable/>

Before the nodes in the above datasets could be stored in the vector databases and indexed, their corresponding text embeddings had to be calculated using an embedding model. This step is essential as these embeddings or vector representations would then be used by the vector database to create the index and to store the nodes along with their vector embeddings. The embedding model used to generate embeddings for dataset-1 was ‘BAAI/bge-large-en-v1.5’, hereafter called embedding model-1, available as an open-source embedding model on the Hugging Face AI community website⁵. Thereafter, a vector store index was created for dataset-1 using their nodes and corresponding embeddings which hereafter is called vector store index-1, as shown in Figure 1.

On the other hand, dataset-2 consisted of comments that were not questions. Therefore, such comments would either comprise opinions on FHSA or some facts or information about FHSA. Thus, we could say that dataset-2 would contain comments that discussed positives of FHSA, negatives of FHSA, and some comments that would not particularly discuss positives or negatives, or in other words, those comments would be neutral. For this dataset, we needed an embedding model which could generate embeddings such that embedding vectors for positive comments were closer to each other but far from the negative comments, and vice-versa. However, the embedding model-1 could not generate such embeddings which was evaluated using the following steps:

- (i) A random subset of Data Frame-2 was drawn containing approximately a thousand comments, as shown in Figure 2.
- (ii) A labeled dataset from this subset was created using ChatGPT which gave labels as ‘positive’, ‘negative’, or ‘neutral’ corresponding to each of the thousand comments, depending upon if the comments were discussing positives or negatives about

⁵ <https://huggingface.co/>

FHSA, or not particularly discussing positives or negatives. We call this dataset dataset-3, as can be seen in Figure 2.

- (iii) A two-dimensional t-SNE (Maaten and Hinton (2008) [27]) plot, shown in Figure 2, of the embeddings of the entire dataset-3 computed by embedding model-1, was generated since the embedding vectors (with originally 1024 dimensions) had to be visualized. The points in the plot were color-coded by their labels.

From Figure 2, it was clear that there was no clustering of points by their labels. Thus, it was clear that embedding model-1 cannot be used as is, to generate embeddings of dataset-2. There was a need to fine-tune this embedding model so that embeddings generated by the fine-tuned embedding model formed clusters according to the labels of the data points, which meant that embeddings of the comments discussing positives would be closer to each other than those of comments discussing negatives and vice-versa.

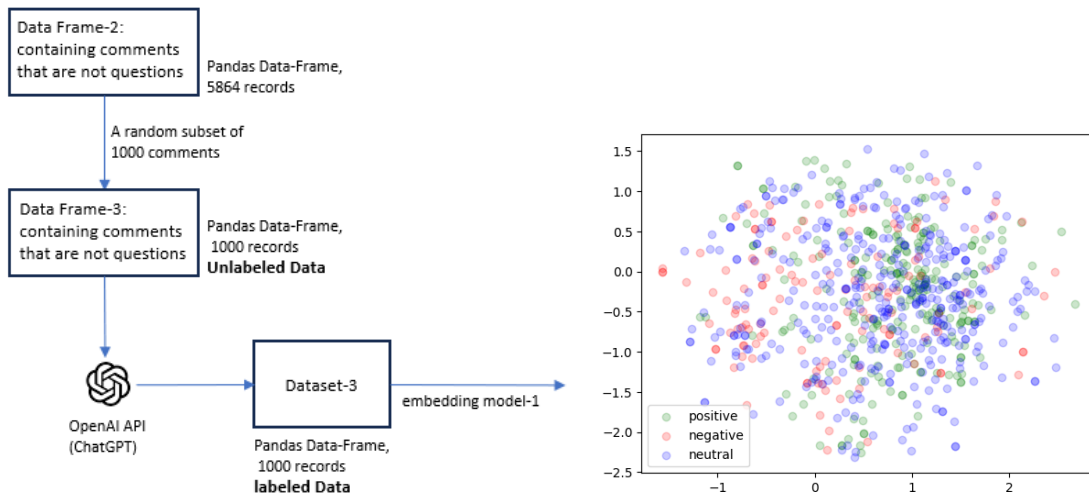


Figure 2: Process flow for preparing dataset-3 and two-dimensional t-SNE plot of embeddings of dataset-3 generated by embedding model-1.

3.3 Preparing Training Examples for Fine-Tuning the Embedding Model

To fine-tune embedding model-1, we used the Python module – Sentence Transformers⁶, which is one of the many transformers (Vaswani et al. (2017) [9]) -based Python modules that can communicate with Hugging Face embedding models such as embedding model-1. Sentence Transformers module was used primarily because of a loss function it provided called ‘Triplet Loss’ which was based on the work done by Hermans et al. (2017) [28], and could be used inside the training loop while fine-tuning the embedding model-1. The way the loss function works is as follows:

- (i) A certain number of training examples in the form of triplets are provided as training data to be used for the embedding model training (or fine-tuning in this case). A typical triplet is of the form – (A, B, C), where ‘A’, ‘B’, and ‘C’ are placeholders for comments, as discussed in subsequent steps.
- (ii) The comments whose embeddings we desire to be closer, take the place of A and B, while comments whose embeddings we desire to be farther from them, take the place of C, as illustrated in Figure 3.

For example, two comments discussing positives about FHSA can be put in place of A and B, while a comment discussing negatives, or another comment that is neutral can be put in place of C (Figure 3(a)).

As another example, two comments discussing negatives about FHSA can be put in place of A and B, while a comment discussing positives, or another comment that is neutral can be put in place of C(Figure 3(b)).

⁶ <https://www.sbert.net/>

- (iii) The loss function for a given triplet is mathematically defined as,

$$\text{Loss} = \max (\|A - B\| - \|A - C\| + \text{margin} , 0) ^7.$$

Here, the expression $\|A - B\|$ represents the distance between the embedding vector of input A and the embedding vector of input B. The calculation of this distance depends on the type of distance metric used while defining the loss function. Furthermore, here ‘margin’ is a hyper-parameter that needs to be tuned as per the specific task. This hyperparameter represents the distance that C needs to be farther away from A than B is from A.

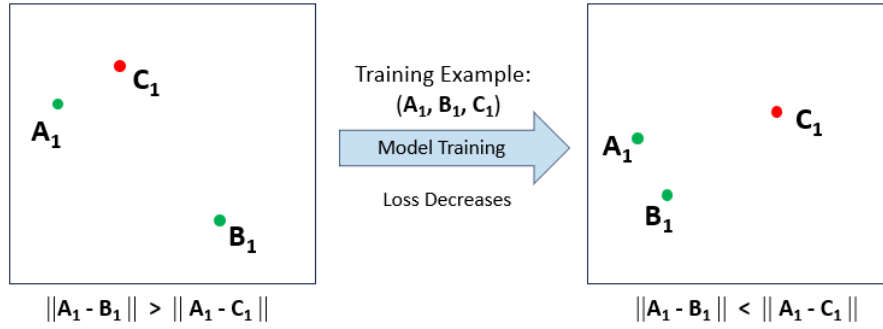


Figure 3(a): When comments discussing positives take the place of A and B in triplet (A,B,C)

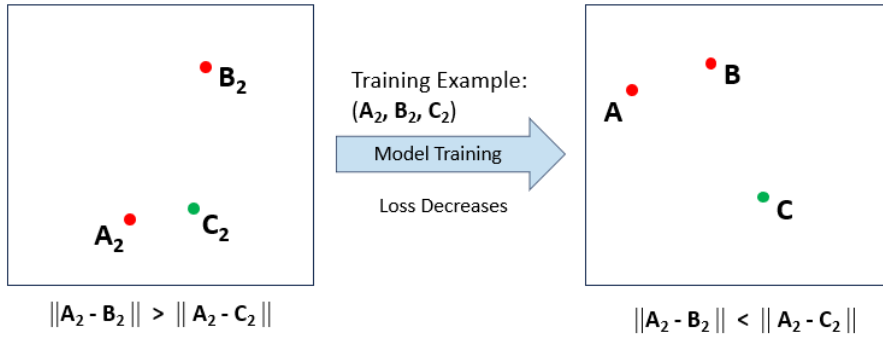


Figure 3(b): When comments discussing negatives take the place of A and B in triplet (A,B,C)

● Comments discussing positives ● Comments discussing negatives

Figure 3: t-SNE plots of embeddings of hypothetical training examples showing the effect of embedding model training on the embeddings of comments in the training examples.

⁷ https://www.sbert.net/docs/package_reference/losses.html

This loss function was well suited for our task because we could conveniently generate the training examples required by this loss function, from the available data, which was dataset-3.

The following steps were taken to prepare the training examples:

- (i) A random subset, hereafter called Set T, of the labelled dataset - dataset-3, was taken containing 250 comments while maintaining the proportion of comments related to positives, negatives, and neutral comments in both Set T and dataset-3, as shown in Figure 4.

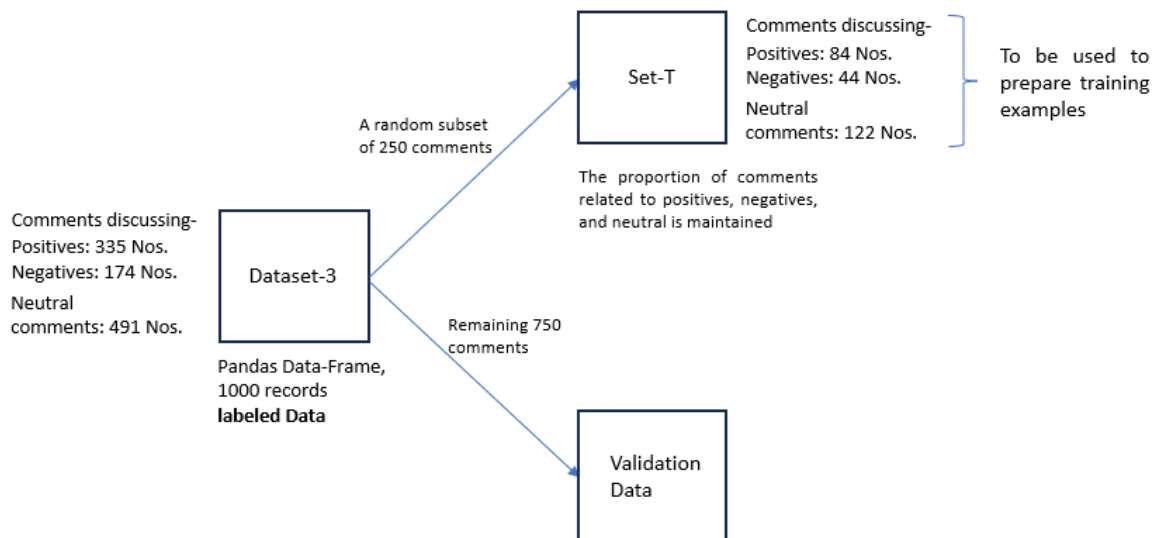


Figure 4: Process flow of splitting labeled dataset-3 into training data and validation data

- (ii) A set of prompts, hereafter called Set P, containing prompts asking about the positives of a product in general, was designed in which all prompts had the same semantic meaning but comprised of different words. Similarly, another set of prompts, hereafter called Set N, containing prompts asking about the negatives of a

product in general, was designed. This was done because the comments discussing positives (or negatives) were required to be closer, in addition the embedding representation of a user prompt asking about positives (or negatives) was required to be closer to the cluster of comments discussing positives (or negatives). This is needed because, during retrieval, the comments in the proximity of the embedding vector of the prompt would be retrieved.

Prompts comprising Set P were:

- a) 'List the positives of the financial product which is being talked about in these comments.'
- b) 'Summarize the advantages of this new product which is being discussed in these social media comments.'
- c) 'Describe the benefits of this new service which people are talking about in these posts.'
- d) 'List the positives or advantages or benefits of the product that is being discussed in these social media posts.'
- e) 'How the financial product discussed here is designed to help the target group of people?'

Prompts comprising Set N were:

- a) 'Summarize the negatives of the financial product which is being talked about in these comments.'
- b) 'List the disadvantages of this new product which is being discussed in these social media comments.'
- c) 'List the limitations of this new service which people are talking about in these posts.'
- d) 'Describe the negatives or disadvantages or limitations of the product that is being discussed in these social media posts.'
- e) 'How does the new financial service being talked about here increase the worries or hardships of the people?'

- (iii) Using Set T, and Sets P and N, 250 triplets of the form (A, B, C) were prepared by drawing out samples from these sets at random by maintaining the following conditions:
- 1) If a prompt from Set P, or a comment from Set T discussing positives, was put in place of A, then place B could only be taken by a comment from Set T discussing positives, or by another prompt from Set P. At the same time, place C could only be taken by a comment from Set T discussing negatives, or by a comment that was neutral.
 - 2) If a prompt from Set N, or a comment from Set T discussing negatives, was put in place of A, then place B could only be taken by a comment from Set T discussing negatives, or by another prompt from Set N. At the same time, place C could only be taken by a comment from Set T discussing positives, or by a comment that was neutral. If a comment from Set T that is neutral is put in place of A, then place B could only be taken by another comment from Set T that is neutral. At the same time, place C could only be taken by a comment from Set T discussing positives, or a comment from Set T discussing negatives.
- (iv) It was made sure that about 100 triplets out of 250 had a prompt from either Set P or Set N since we wanted the embeddings for the comments from Set T discussing positives (or negatives) to be in proximity to the embeddings of the prompts asking about positives (or negatives).

3.4 Fine-Tuning of Embedding Model

Now that we had the training examples (250 in number) in the form of triplets, as discussed in Section 3.3, required by the loss function – Triplet Loss, the next step was to train the embedding model-1 using these examples. For this training step, we used the `fit()` method of the Sentence Transformers module. It is to be noted, however, that given the size of embedding model-1 (340M parameters), it was not feasible to train the model using CPU within a reasonable time duration, thus a Tesla V100 GPU with 16GB memory was employed for this model training which took about 12 minutes for a given hyperparameter.

As mentioned in Section 3.3, the hyperparameter – margin, had to be tuned as per our specific use case. For this purpose, the remaining 750 comments from dataset-3, which were not used in training were used as validation data as shown in Figure 4. Figure 5 shows two extreme cases and an optimized case of the hyperparameter tuning process. All the plots on the left-hand side in Figure 5 show the embeddings of the validation data, color-coded as per the labels from ChatGPT, while all the plots on the right-hand side show the embeddings of the complete dataset-2, color-coded on the condition if the comments contained the word ‘FHSA’/ ‘fhsa’/ ‘Home’/ ‘home’/ ‘Account’/ ‘account’.

The plots in Figure 5(a) represent the embeddings by the embedding model-1, which was not fine-tuned. As can be seen from the left-hand side plot in Figure 5(a), the model had poor clustering performance on the validation data because there is no grouping of the comments with respective labels. However, the model showed a decent ability to filter noise, as demonstrated in the right-hand side plot of Figure 5(a), because all the comments related to the topic (FHSA) are grouped and separated from the rest of the comments.’

The plots in Figure 5(b) represent the embeddings by a fine-tuned embedding model when the margin was 0.5. This embedding model showed impressive clustering results as can be seen in Figure 5(b) in the left-hand side plot because the embeddings are grouped closely as per their respective labels. However, from the right-hand side plot in Figure 5(b), it can be inferred that this model lost its ability to filter noise since the comments related to the topic were no longer separated from the rest of the comments. This loss of ability to filter noise was not desirable as we would always want only the relevant content to be given to the LLM as context while filtering out all the irrelevant information. Thus, there was a need to find an optimized value of the hyperparameter – margin, which would result in a fine-tuned embedding model that displayed good clustering results and at the same time that could filter noise to a reasonable extent.

Figure 5(c) represents the embeddings by the fine-tuned embedding model when the margin was 0.07, hereafter called embedding model-2. As can be seen in Figure 5(c) on the left-hand side plot, this model could cluster the comments with respective labels to a reasonable extent while maintaining its ability to filter noise which can be observed on the right-hand side plot in Figure 5(c). This model with a balanced trade-off between clustering performance and noise segregation performance was selected as the fine-tuned embedding model to be used in the retriever component of our solution.

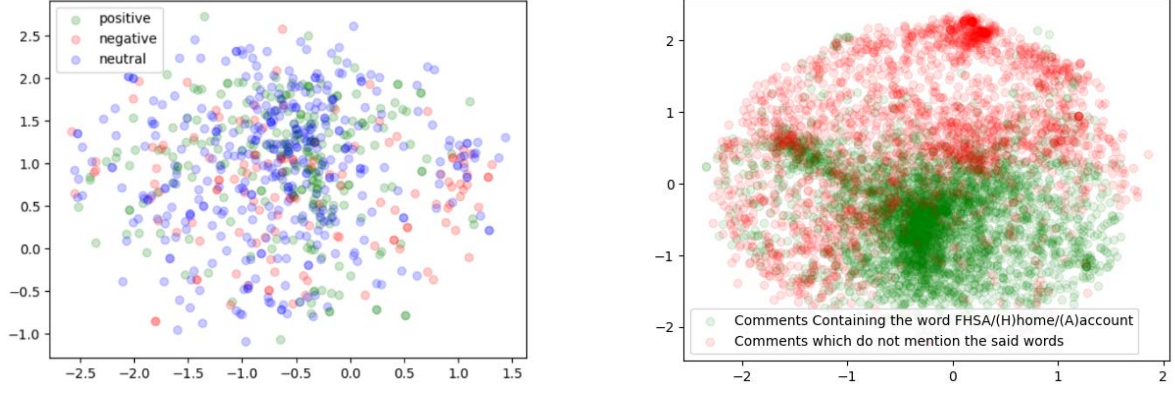


Figure 5(a): Embeddings by embedding model-1(margin=0.0) for validation dataset in the left-hand side plot while for dataset-2 in the right-hand side plot.

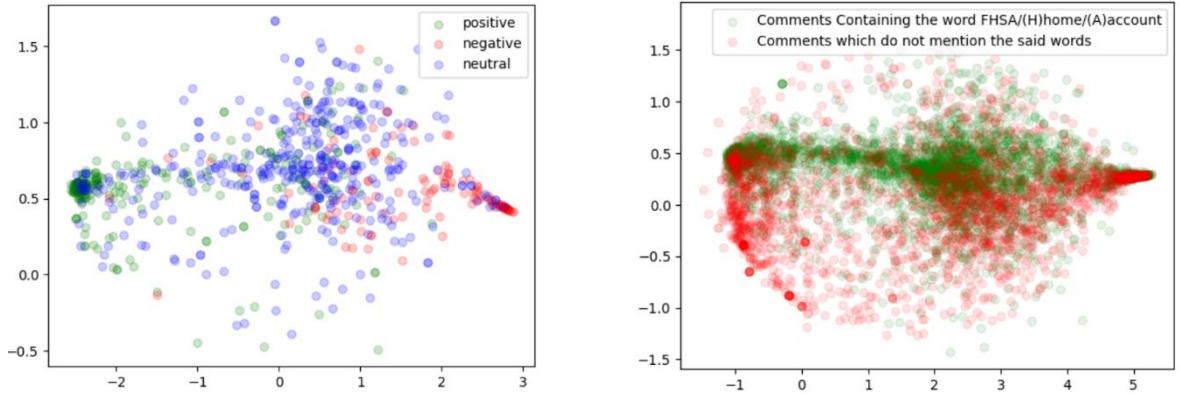


Figure 5(b): Embeddings by the fine-tuned embedding model when margin =0.5, for validation dataset in the left-hand side plot while for dataset-2 in the right-hand side plot.

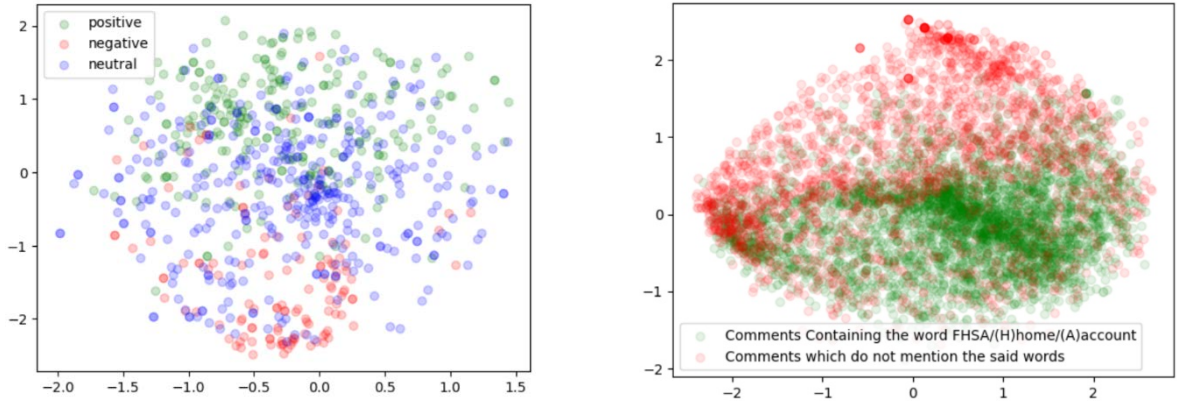


Figure 5(c): Embeddings by the fine-tuned embedding model (selected) when margin =0.07, for validation dataset in the left-hand side plot while for dataset-2 in the right-hand side plot.

Figure 5: Hyperparameter Tuning for the fine-tuned embedding model.

3.5 Validating Selected Embedding Model Using Test Dataset on an Unrelated Topic

As discussed in section 3.4, we used the comments left from dataset-3, after taking out the comments for the training data, as validation data for hyper-parameter tuning of the fine-tuned embedding model. Therefore, we needed another dataset to act as a test dataset which was unseen by the model in training as well as validation. This test dataset was chosen to be on a different topic than FHSA to evaluate the ability of the fine-tuned model to perform well on a dataset with a topic not used during training. If the selected fine-tuned model could show adequate clustering performance on this test dataset, then it would indicate that the model was able to generalize well for different datasets.

For this task of validating the selected embedding model, we used a dataset containing social media comments about the Tax-Free Savings Account (TFSA). From this dataset, which contained about 6500 comments, a random subset of a thousand comments was drawn. A labeled dataset from this subset was created using ChatGPT which gave labels as ‘positive’, ‘negative’, or ‘neutral’ corresponding to each of the thousand comments, depending upon if the comments were discussing positives or negatives about TFSA, or not particularly discussing positives or negatives. We label this dataset as dataset-4, for which text embeddings of the comments in the dataset were generated using the selected fine-tuned embedding model. Figure 6 shows a two-dimensional t-SNE plot of the text embeddings of the comments in dataset-4 generated from the embedding model-2, color-coded as per the labels in dataset-4, as mentioned in the plot legend. It is evident from Figure 6 that the embedding model-2 performed reasonably well on the task of clustering the comments as per their labels, as we can see the comments with ‘positive’ labels are in one cluster on the left in the plot while the comments

with ‘negative’ labels are in one cluster on the right-hand side in the plot. Therefore, the model generalized well on an unseen test dataset concerning an unrelated topic.

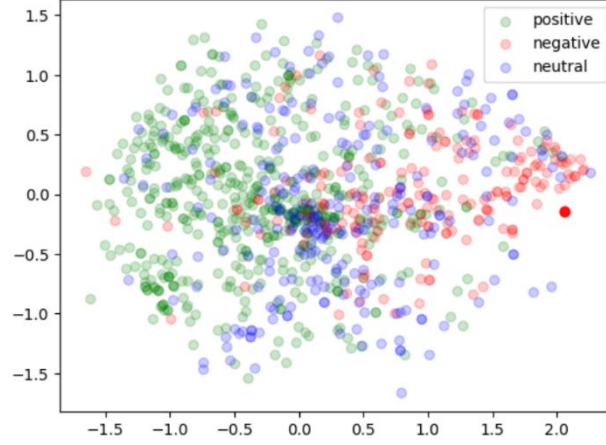


Figure 6: Two-dimensional t-SNE plot of embeddings of dataset-4 generated by the selected embedding model.

3.6 Data Pipeline and Implementation

As mentioned in Section 3.1 and Section 3.2, the set of comments in dataset-1 were related to the questions being asked by people and would form an ideal dataset for the analysis related to uncertainties about FHSA. Recall that these comments were stored along with their embeddings, generated by embedding model-1, in a Pinecone vector database in the form of an index, which we called vector store index-1. Hence, vector store index-1 would act as an external knowledge base for the generator component (LLM) of our RAG system for the prompts concerning the uncertainties about FHSA.

On the other hand, dataset-2 consisted of comments that discussed the positives and negatives of FHSA along with the comments that were neutral (not particularly discussing positives or negatives). However, for the reasons explained in Section 3.2, embedding model-1 was not a recommended embedding model to generate embeddings for dataset-2. Therefore, the

embedding model-1 was fine-tuned as per the specific task of grouping the comments discussing positives in one cluster while the comments discussing negatives in another cluster, as discussed in Section 3.3. The selected fine-tuned embedding model (after hyperparameter tuning), embedding model-2, was then employed to generate embeddings for the comments in dataset-2, which were then stored in another Pinecone vector database along with their embeddings in the form of an index, hereafter called vector store index-2. Hence, vector store index-2 would act as an external knowledge base for the generator component (LLM) of our RAG system for the prompts concerning the positives and negatives of an FHSA.

It should be noted here that we effectively built two RAG systems corresponding to the two vector store indices- vector store index-1 and vector store index-2 - concerning the prompts related to uncertainties and the prompts related to positives or negatives, respectively. Recall that the retriever component in an RAG system consists of a knowledge base, typically a vector store index, and an encoder which is a text embedding model used to create embeddings of the prompts. Having dealt with the knowledge base component, the next logical step was to define encoders for the retrievers of the two RAG systems. For the retriever of the RAG system concerning uncertainties, embedding model-1 was set as the encoder component, while for the retriever of the RAG system concerning positives and negatives, embedding model-2 was set as the encoder. This selection of the encoders was logical because for the system related to uncertainties, embedding model-1 generated the embeddings in the knowledge base, while for the system related to positives and negatives, embedding model-2 generated the embeddings in the knowledge base. Therefore, this selection would result in an adequate retrieval of top-K comments from a given knowledge base. For instance, given a prompt asking about the positives of FHSA, a text embedding of the prompt generated by embedding model-2 would be in proximity to the embeddings of the comments discussing positives because that was how

embedding model-2 was trained (Section 3.3). With this selection, the retriever components of both RAG systems were complete.

Next, we highlight the importance of fine-tuning of the embedding model in the retrieval process by comparing the retrieval performance of embedding model-1 and embedding model-2 on dataset-2, given a pair of prompts asking for positives and negatives of FHSA. Figure 7(a) shows a two-dimensional t-SNE plot of embeddings for comments from dataset-2, where the top 1000 comments were retrieved corresponding to the prompt asking for positives – Prompt_1 and the prompt asking for negatives – Prompt_2, when the embeddings of the comments were generated using embedding model-1. Figure 7(b) shows the same plot but when embedding model-2 was used to generate embeddings of the comments.

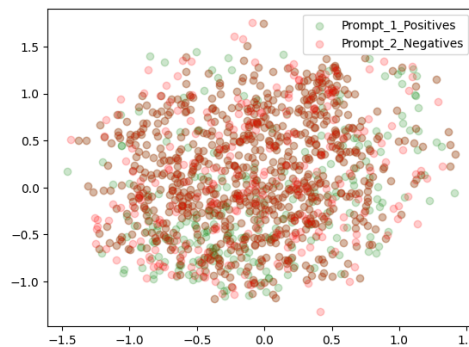


Figure 7(a): Two-dimensional t-SNE plot of embeddings of dataset-2 generated by embedding model-1 for the top 1000 comments each retrieved using Prompt_1 and Prompt_2.

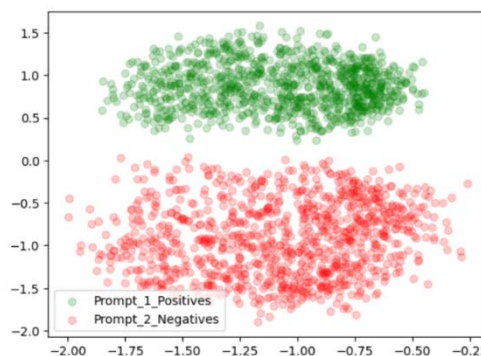


Figure 7(b): Two-dimensional t-SNE plot of embeddings of dataset-2 generated by embedding model-2 for the top 1000 comments each retrieved using Prompt_1 and Prompt_2.

Figure 7: Comparison of retrieval performance on dataset-2 between embedding model-1 and embedding model-2.

The number of common comments retrieved between Prompt_1 and Prompt_2 in the case of embedding model-1 (Figure 7(a)) were 760 while in the case of embedding model-2 (Figure 7(b)), they were 0. Thus, in the case of embedding model-1, in the embedding vector space, there is no separation between the comments retrieved using Prompt_1 and the comments retrieved using Prompt_2 as can be seen in Figure 7(a), which is a cause of concern as this would result in irrelevant information being passed to the context of the LLM. In other words, when the prompt asked for positives, comments discussing negatives would also be passed onto the context inevitably, and vice-versa. On the other hand, there is a clear separation between the comments corresponding to Prompt_1 and the comments corresponding to Prompt_2 which is ideal because this separation would avoid irrelevant information being passed onto the LLM context.

Having validated the retrieval performance with the fine-tuned embedding model, the next step was to put together all the pieces in a data pipeline to build the required RAG system. Figure 8 shows how data was transferred between different components of our solution. We start with LlamaIndex which converted texts to nodes as discussed in Section 3.1. These nodes were sent to the respective embedding models- embedding model-1 for uncertainties and embedding model-2 for dataset-2. It is worth mentioning that embedding model-1 was downloaded locally using the Hugging Face Python module, and we used the Langchain Python module to establish a connection between the embedding model and LlamaIndex. When embedding model-1 was fine-tuned, the fine-tuned version was saved locally, as well, which we called embedding model-2. The embedding models returned the embeddings of the text data (social media comments) which were then uploaded along with the text data itself to respective vector databases by creating indices – vector store index-1 for uncertainties, and vector store index-2 for positives and negatives. These vector databases were from a vector database service provider – Pinecone. Also, the Pinecone Python module was used to connect LlamaIndex with

Pinecone vector databases to perform the upload process of embeddings along with their text data. For a given set of external data (social media comments in this case), the data flow discussed until now was a one-time activity. However, this activity had to be done each time new data was to be added to the respective vector databases.

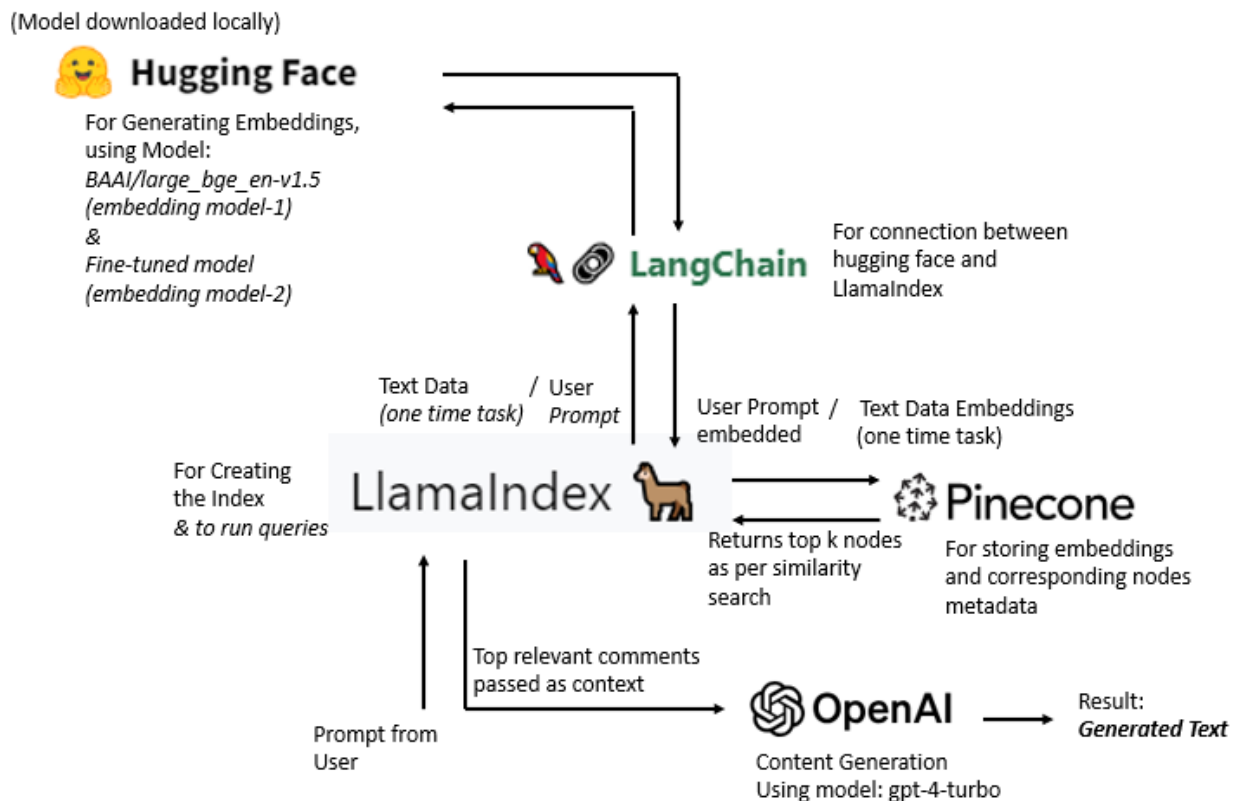


Figure 8: Data Flow between different components of the solution.

Having discussed the knowledge base construction and dataflow in detail, we can shift our attention to the retrieval process and generation process of the RAG system, and how information flowed when the user gave a prompt to our solution. A prompt from the user was fed to a LlamaIndex object which sent it to the required embedding model – embedding model-1 for uncertainties and embedding model-2 for prompts asking for positives or negatives. The embedding model returned the embedding of the prompt using which top ‘K’ (a parameter)

number of comments based on their similarity scores were retrieved from the respective vector database. The parameter 'K' could be 100, 500, 1000, etc. The similarity score was based on the similarity metric which was used to construct the vector store index. For both vector store index-1 and vector store index-2, the similarity metric used was cosine similarity. This process based on similarity scores essentially implemented a semantic search and retrieval process because the embeddings of two comments with high cosine similarity score would simply mean that the two comments carried very similar semantic meanings even if they were composed of different words. Now the prompt was given to the generator component, which was the set large language model (GPT-4-Turbo in our case), along with the top retrieved comments as context. The large language model utilized the input context and gave the required output as requested in the prompt.

4. Results and Discussion

Although discussed in Section 1.2, we reiterate that the scope of this study was limited to preparing a summarization of the positives of FHSA, limitations of FHSA, and any questions or uncertainties that were being discussed about FHSA based on the social media comments in the given initial dataset (before dividing into dataset-1 and dataset-2). Therefore, in this chapter, we look at the summaries generated by our solution corresponding to prompts about positives, negatives, and uncertainties. However, in addition, it was realized that it was important to gauge how the LLM used responded to different sizes of input context, and what the impact was on the output generated if we had not fine-tuned the embedding model used to generate the embeddings of dataset-2. In the following sections, we discuss the results related to all the aspects outlined here.

4.1 Impact of Input Context Size on the Summarization Generated by LLM

We used GPT-4-Turbo as the large language model in our solution which was an LLM by OpenAI. This model had a context window of 128K tokens which meant that it could accept an input context that consisted of 128000 tokens or roughly 96000 words. However, it would not be recommended to utilize 100% of this context window just because it was available, as discussed in Chapter 2 in the section on the impact of context length. Therefore, it was essential to find out whether there was any significant impact of the context size. We studied this impact by generating a summarization of the positives and negatives of FHSA by providing input context of different sizes. It should be noted that all the results presented in this section were

generated with fine-tuned embedding model. To analyze the impact of fine-tuning, we performed a comparison which is outlined in Section 4.2.

A. Summarization Results on Positives with Different Context Sizes

Prompt: *'Highlight the positives or advantages of an FHSA in 5 unique points using the information from these social media comments. Arrange the points in decreasing order of the number of times they are discussed.'*

The summarization generated by the solution is listed as follows:

- I. With a Context Size of about 95000 words (~ 100% use of context window):
 1. **Homeownership:** Many individuals discuss the FHSA as a valuable tool for saving for a down payment on a home. It allows individuals to accumulate funds specifically designated for purchasing a home, making homeownership more attainable.
 2. **Tax Deduction:** The FHSA's tax deduction feature is frequently mentioned as a major advantage. By contributing to an FHSA, individuals can reduce their taxable income and potentially save on taxes.
 3. **Tax-Free Growth:** The opportunity for tax-free growth within the FHSA is often highlighted. Any interest or investment gains earned within the account are not subject to taxation, allowing for potential long-term growth of savings.
 4. ****Accessibility:** The FHSA is frequently mentioned as an accessible savings option that can be utilized by individuals with higher incomes. It provides a tax-advantaged way to save for various financial goals.**
 5. ****Contribution Limits:** The FHSA's specific contribution limits are often discussed. With a maximum annual contribution of \$8,000 and a total lifetime contribution limit of \$40,000, these limits ensure that individuals can save and benefit from the tax advantages of FHSA within defined boundaries.**

In the above output, the points enclosed in ‘**...**’ were the points that were not desirable in the output. This was because Point 4 provided no further information beyond Point 2 and Point 3, whereas Point 5 only discussed some facts and did not provide any further information related to the benefits of an FHSA. Though, points 1 to 3 were acceptable as they mentioned three different advantages of FHSA, the complete output was still not satisfactory. We could infer from this result that when a larger context was provided, some relevant information was missed by the LLM and instead the LLM got distracted by irrelevant information. This claim was further supported by the following result when a concise context was provided.

II. With a Context Size of about 20,000 words (~20% use of context window):

1. **Tax Deduction:** Many users discuss the tax deduction as a major advantage of FHSA. By contributing to an FHSA, individuals can lower their taxable income and potentially save on taxes.
2. **Homeownership:** Another frequently mentioned benefit of FHSA is its ability to help individuals save for a down payment on a house. By contributing to an FHSA, individuals can specifically allocate funds for their first home purchase.
3. **Tax-Free Growth:** Users also highlight the advantage of tax-free growth within an FHSA. Any interest or investment gains earned within the account are not subject to taxation, allowing for potential long-term growth of the funds.
4. **Access to Government Programs:** FHSA is often discussed in relation to other government programs, such as the Home Buyers Plan (HBP) and the Registered Retirement Savings Plan (RRSP). These programs can work together with an FHSA to provide additional benefits and options for saving and investing.
5. **Contribution Room:** The contribution room of FHSA is mentioned as an advantage. With a maximum annual contribution limit of \$8,000, individuals have the opportunity to accumulate contribution room over time, allowing for larger contributions in future years if desired.

All points in the above output were unique and provided valuable information about the benefits of FHSA. Details on the points as to why they were beneficial were well explained.

Besides the difference in content, the time taken to get the response from OpenAI API was about 16 minutes when the context size was about 95,000 words while it took only 2 minutes when the context size was about 20,000 words. Thus, we could get better results with much less computation time (as well as computation cost). This comparison of the generated output with different context sizes clearly suggested that fully utilizing the available context window might not be the best way forward. Though 20% use of context window showed decent performance, we could not say that it was the optimum context size as judging that would require further experimentation that would be dependent on the size of the available dataset.

B. Summarization Results on Negatives with Different Context Sizes

Prompt: *'Highlight the negatives or limitations of an FHSA in 5 unique points using the information from these social media comments. Arrange the points in decreasing order of the number of times they are discussed.'*

The summarization generated by the solution is listed as follows:

- I. With a Context Size of about 95000 words (~ 100% use of context window):
 1. Limited availability of financial institutions offering FHSA makes it challenging to access this savings vehicle.
 2. The strict 15-year time limit of the FHSA means that funds must be used for a home purchase within that timeframe or be transferred to an RRSP or taxed as income.

3. ****Any interest earned in the FHSA is subject to taxation, potentially reducing the overall returns on the investment.****
4. ****The FHSA may disqualify individuals from other first-time homebuyer incentives or programs, limiting their options for financial assistance.****
5. Limited investment options within the FHSA may not meet the needs of individuals seeking more diverse or higher-yielding investments.

In the above output, the points enclosed in ‘***...***’ were the points that were not desirable in the output. This was because both Point 3 and Point 4 were factually incorrect. Since there was a possibility that these results were because of misinformation in the comments themselves, further analysis was done using the retrieved comments, and were used to generate the above output. It was found that such information was not mentioned in any of the retrieved comments. This generation of unexpected output could be attributed to “hallucinations” (Maynez et al. (2020) [21]. Further, to identify, for instance, what comment(s) could have led to Point 4, comments were identified using a secondary prompt – ‘Give a list of comments from the social media comments below which support this statement’. It was found that Point 4 could have been generated because of the comments below:

- (i) “You may not qualify for first time home buyers’ incentives or the FHSA nor RRSP withdrawal under the first-time buyer plan. Bad idea for all these other reasons too.”
- (ii) “My mom put me on her deed for similar reasons. I become ineligible for things like HBP and FHSA.”

As was evident from the two comments above, the claim made in Point 4 is not supported by any of these comments. These comments were just comprised of words that might have led to Point 4 but carried a different meaning. This exercise on the

negatives of an FHSA with almost 100% use of context further supported our claim (as the similar exercise on the positives) that it might not be in the best interest to make the input context as large as possible.

II. With a Context Size of about 20,000 words (~ 20% use of context window):

1. Many individuals express frustration with the limited investment options offered by big banks for an FHSA, such as the lack of self-directed options and high fees associated with available mutual funds.
2. Commenters are disappointed with the 15-year time limit on FHSA, as it can be seen as a drawback. If the funds are not used for a home purchase within that time, they are either withdrawn as taxable income or transferred to an RRSP.
3. Some commenters find the eligibility criteria for an FHSA to be restrictive, particularly for common-law couples who may be ineligible to open an FHSA. Additionally, living in a home owned by a spouse disqualifies one from the program.
4. Several individuals express concerns that an FHSA primarily benefits homebuyers rather than renters, leading to increased house prices and widening the wealth gap.
5. The FHSA program is criticized for its perceived failure to effectively address wealth disparity and housing affordability. Commenters argue that demand-side policies like FHSA only worsen the issues without addressing the underlying problems.

All points in the above output were unique and provided valuable information about the negatives or limitations of an FHSA. Therefore, results with a lower context size were significantly better than the results with a much larger context size.

Besides the difference in content, the computation time or the time taken to get the response from OpenAI API was about 18 minutes when the context size was about 95,000 words while it took only 3 minutes when the context size was about 20,000 words. Thus, we could get better results with much less computation time (as well as computation cost).

4.2 Analyzing Summarizations Generated without Fine-Tuning of Embedding Model

In the previous chapter, we saw the impact of fine-tuning the embedding model only in the retrieval process (Section 3.6). In this section, we discuss how the fine-tuning impacts the end product- the summarization generated by our solution. We studied this impact for both cases of summarization – positives of an FHSA as well as negatives of an FHSA. However, to keep a variable (context size) fixed for an apt comparison, we examined the results where the context size used was about 20,000 words. The following results only show the output for the case when the embedding model was not fine-tuned. For the results with the fine-tuned embedding model, please refer to Section 4.1.

I. Summarization Results on Positives without Fine-Tuning of Embedding Model

Prompt: *‘Highlight the positives or advantages of an FHSA in 5 unique points using the information from these social media comments. Arrange the points in decreasing order of the number of times they are discussed.’*

1. **Homeownership support:** The FHSA is specifically designed to help individuals save for their first home, making homeownership more accessible and achievable.
2. **Tax benefits:** The FHSA offers tax advantages, such as tax deductions on contributions and tax-free growth on investments.
3. ****Flexibility:** The FHSA allows individuals to contribute as much as they want and have multiple FHSA accounts. This flexibility is mentioned in the comments, indicating that it is seen as a positive feature of the FHSA.**

4. ****Investment opportunities:** The FHSA allows individuals to invest their contributions, potentially earning higher returns than traditional savings accounts.**
5. ****Long-term Savings:** The FHSA encourages individuals to save for the long term, promoting disciplined saving habits and helping individuals build wealth over time. This is seen as a valuable feature of the FHSA.**

In the above output, points 3, 4, and 5 were not desirable, because Point 3 and Point 4 were factually incorrect. FHSA did not have unlimited contribution room and did not offer particularly higher returns on investment than traditional accounts. Point 5 only discussed a general statement, nothing particular about the benefits of an FHSA. Although the first two points were accurate, the output as a whole was not satisfactory.

II. Summarization Results on Negatives without Fine-Tuning of Embedding Model

Prompt: *'Highlight the negatives or limitations of an FHSA in 5 unique points using the information from these social media comments. Arrange the points in decreasing order of the number of times they are discussed.'*

1. Many individuals express frustration with the limited contribution limit of the FHSA, stating that \$8,000 per year is insufficient to combat the high cost of housing. They propose increasing the limit to \$10,000 per year or higher.
2. ****Individuals express concerns about the administrative burden and complexity of the FHSA. They mention difficulties in accessing information about the program and frustration with the lack of clarity and ease of use.****
3. ****Some people are disappointed with the overall effectiveness of the FHSA, as they have reached the maximum contribution limit but receive only a small amount back**

in return. They feel that the program does not provide significant benefits considering the amount they have invested.**

4. Critics argue that the FHSA primarily benefits homeowners rather than renters and does not effectively address the underlying issues of housing affordability and wealth disparity. They believe the program fails to achieve its intended goals and exacerbates the housing market for individuals struggling to afford a home.
5. **Critics argue that the FHSA primarily benefits those who are already financially stable, widening the wealth gap. They believe the program exacerbates the housing market for individuals struggling to afford a home and fails to effectively address housing affordability.**

Points 2, 3, and 5 in the above result were not desired. Point 2 did not mention any limitation specifically, rather discussed difficulties in general which was not valuable. Similarly Point 3 lacked clarity. Point 5 provided no information beyond Point 4. Thus, the quality of content in this output was below expectations.

When compared to the results with the fine-tuned embedding model for the above cases- positives and negatives- it was clear that the results with the fine-tuned embedding model were better where all points were acceptable, accurate, and unique, thus highlighting the importance of fine-tuning the embedding model.

4.3 Summarization on Positives of FHSA

Although in Section 4.1 we had discussed the outputs of our solution on positives of FHSA with input context size of 95,000 words (~100% of context window) and 20,000 words (~20% of context window), it was still a point of interest to look at the results with a different context size such as 38,000 words (~40% of context window), approximately equivalent to 800 comments, and check if this context size would still result in an acceptable output.

Prompt: *‘Highlight the positives or advantages of an FHSA in 5 unique points using the information from these social media comments. Arrange the points in decreasing order of the number of times they are discussed.’*

Following was the result generated by our solution with a context size of 38,000 words,

1. **First-Time Home Buyer Advantage:** The FHSA's focus on first-time home buyers is frequently mentioned as a major advantage. It provides a dedicated savings vehicle for individuals looking to save for a down payment on their first home, while also enjoying the tax advantages.
2. **Tax Deduction:** The FHSA's tax deduction feature is often highlighted as a significant benefit. By deducting contributions from taxable income, individuals can lower their tax burden and potentially save more towards their home purchase.
3. **Tax-Free Growth:** Another commonly discussed benefit is the tax-free growth of savings within the FHSA. This means that any interest or investment gains earned within the account are not subject to taxes, allowing for potential significant growth over time.
4. **Flexibility:** The FHSA's flexibility in terms of investment options is frequently mentioned. Some comments suggest investing in high-interest savings accounts or GICs, while others mention the possibility of investing in ETFs or other products. This flexibility allows individuals to tailor their investment strategy to their risk tolerance and financial goals.
5. ****Max Contribution Room:** Many comments emphasize the importance of maximizing contributions to the FHSA. By contributing the maximum allowed amount each year, individuals can fully leverage the tax benefits and potentially accumulate a substantial amount for their home purchase.**

In the result presented above, only Point 5 could be challenged. Although, the point was factually correct, it failed to present any further information about the benefits of an FHSA. Therefore, we could say that the output with 38,000 words context was still acceptable though one of the points in the summary was not very specific. Thus, one would judge a

recommended context window for the concerned dataset – dataset-2, to be limited below 40%.

4.4 Summarization on Negatives of FHSA

Along the same lines as in Section 4.3, we looked at the results for negatives on FHSA, with a different context size which was 38,000 words (~40% of context window), and studied if the context size would still result in an acceptable output.

Prompt: *‘Highlight the negatives or limitations of an FHSA in 5 unique points using the information from these social media comments. Arrange the points in decreasing order of the number of times they are discussed.’*

Following was the result generated by our solution with a context size of 38,000 words,

1. The FHSA has a 15-year time limit, meaning that if the funds are not used for a home purchase within that time, they will be transferred to an RRSP or taxed as income.
2. Some financial institutions may not offer the option to open an FHSA, limiting accessibility for potential investors.
3. ****Opening an FHSA may make individuals ineligible for other first-time homebuyer incentives or programs.****
4. The limited selection of investments in the FHSA may not be appealing to potential investors.
5. The FHSA is seen as a bad policy by many, with criticisms including its regressive nature, inflationary effects, and lack of effectiveness in addressing housing affordability issues.

The output listed above contained points that were acceptable apart from Point 3, enclosed inside ****...****. Point 3 made a claim that was inaccurate and hence indicated that the use of 38,000-word context, 40% of the context window, was questionable. Thus, one would judge a recommended context window for the concerned dataset – dataset-2, to be limited below 40%.

Therefore, from the above observations of Section 4.3 and 4.4, we could infer that for a dataset of the size similar to dataset-2 (containing about 6000 comments), more than 40% use of context window (which was equivalent to 800 comments) might not generate acceptable results for the problem at hand. Furthermore, it must be noted that this inference was made only on the basis of dataset-2, and if one intended to estimate a general rule of the relationship between optimum context size and the size of the dataset, then one would have to perform multiple case studies with different datasets and with different context sizes.

4.5 Summarization on Uncertainties about FHSA

As mentioned earlier in this report, the uncertainties or questions about FHSA were summarized using the dataset-1. To generate this summary, we used the prompt provided below.

Prompt: *‘Summarize the uncertainties or doubts about an FHSA that are being discussed in these social media comments in 5 points. Arrange the points in decreasing order of the number of times they are discussed.’*

I. With a Context Size of about 20,000 words (~ 20% use of context window):

1. **Uncertainty about the future of the FHSA:** Many individuals are unsure about what will happen to their FHSA if they are not ready to purchase a home within 15 years. They are also unsure about what happens to the money in the FHSA if it is not used for a home purchase.
2. **Confusion about the tax deduction:** People are unclear about how the \$8k tax deduction works for FHSA contributions. They are unsure if it is similar to the tax credit for tuition and if they can save the FHSA tax credit until they have a higher income.
3. **Limited information and resources:** Many individuals express frustration about the limited information available online about an FHSA. They mention that online resources have been limited and that they have had difficulty finding answers to their questions.
4. **Concerns about eligibility and contribution limits:** Some individuals are unsure about the eligibility criteria for FHSA and whether they can contribute if they already own a home. There are also questions about the maximum lifetime limit of contributions and how it is calculated.
5. **Lack of investment options:** People are interested in investing their FHSA funds but are uncertain about the investment options available. They are specifically looking for information about institutions that offer stocks within their FHSA.

II. With a Context Size of about 38,000 words (~40% use of context window):

1. **The duration and fate of FHSA accounts:** Many individuals are discussing their concerns about the 15-year lifespan of FHSA accounts and what happens to the money in the account if they are not ready to purchase a home within that timeframe. They are also unsure about the implications of withdrawing funds from the FHSA before the 15-year period ends.
2. **Maximum annual contribution limit:** There is confusion regarding the annual maximum contribution limit for an FHSA and whether it changes each year. Some individuals are also curious about the possibility of carrying forward unused contribution room from previous years.
3. **The tax deduction for FHSA contributions:** Some individuals are unsure about how the tax deduction works for FHSA contributions. They are uncertain if they

have a specific tax credit space like tuition credits and if they can choose to save the FHSA tax credit until they have a higher income.

4. **Investment options and returns for FHSA:** There are questions about the investment options available for FHSA accounts and the potential returns one can expect. Some individuals are specifically looking for safe and consistent investment options for their FHSA funds.
5. **Eligibility and implications of opening FHSA accounts:** There are uncertainties regarding the eligibility criteria for opening FHSA accounts, such as age, residency, and first-time homebuyer status. Some individuals are also unsure about the implications of opening FHSA accounts under their spouse's name if they are currently working or earning income.

For both the above-presented cases, Case-I and Case-II, all five points summarized by our solution were reasonable, unique, and acceptable. It must be noted here that the dataset used for generating the above two outputs on uncertainties, dataset-1, was separated from the original dataset based on whether the comments contained questions. Therefore, it was safe to assume that the dataset mostly comprised of relevant information, which might be the reason why the output with a larger input context (~40% use of context window) was as good as the output with a smaller input context (~20% use of context window).

5. Recommendations and Future Work

Based on the results and findings discussed in Chapter 4, we have identified some key recommendations:

- (i) Retrieval Augmented Generation (RAG) is a recommended method when the application involves connecting an LLM to an external data source. This method is useful especially when the entire text data from the source exceeds the context window of the LLM, which is usually the case with any real-world dataset. This method is effective because it helps in retrieving only the relevant information from the dataset and provides it as context to the LLM. RAG also helps in keeping the size of input context low which we now know to be highly recommended.
- (ii) The embedding model being used to generate the text embeddings for each document or social media post in the dataset, must be fine-tuned as per the specific use case, to obtain better results. This is because the retrieval process in RAG is effectively a nearest neighbour search and fine-tuning improves retrieval performance by bringing embeddings of ‘similar’ documents together. Here, the meaning of similarity between documents will depend on the specific application. Thus, the process of fine-tuning will largely avoid the irrelevant documents being given as context to the LLM, along with relevant documents. As an instance, for our application and FHSA dataset, we would not have intended to give comments discussing negatives to the LLM, when we only wanted to summarize the positives about the FHSA.
- (iii) A lower usage of the context window is recommended as opposed to using the context window completely as it results in generated content that is highly aligned with the data given as context and content that has minimal redundancy. This is

based on the results discussed in Section 4.1, where the results produced by a smaller input context generated unique summarization points that all contained useful information.

Our solution in its current stage has areas where there is scope for future work. Amongst them, is the ability to find the optimal size of the input context, as a function of dataset size, given to the LLM. For instance, in Chapter 4 we found that, given dataset-2, the recommended size of input context was limited to nearly 38,000 words. The task to determine a general rule to approximate the context size as a function of dataset size, is challenging, because firstly there is no direct metric to measure the quality of the content generated, and secondly multiple results are required to be generated with a variety of datasets of different sizes. Additionally, for each dataset, results must be generated with different percentages of context window, and finally judging them all subjectively. At the same time, this task is important because it helps to find the right balance on how much context, given a dataset, is to be provided to the LLM. To elaborate, we do not want to provide too little context to the LLM as it carries a risk of losing out on useful information in the dataset, and at the same time, we do not want to utilize the full available context window, as this poses a risk of providing irrelevant information to the LLM along with required information and thus resulting in an output that misses out on useful information.

Another area that requires further work is the algorithm for fine-tuning of the embedding model. The fine-tuning methodology adopted in this work was based on subjective opinion in judging if the clustering of social media posts with similar opinions was reasonable enough while maintaining the capability of the embedding model to filter noise. There is a possibility of developing metrics that can quantify the extent of clustering and the extent to

which the embedding model can filter noise. Such metrics can subsequently be used to develop a fine-tuning and hyper-parameter tuning algorithm that essentially automates the entire process of fine-tuning the embedding model.

References

- [1] Q. Hou, M. Han and Z. Cai, "Survey on Data Analysis in Social Media: A Practical Application Aspect," *BIG DATA MINING AND ANALYTICS*, pp. 259-279, 2020.
- [2] P. Ducange, R. Pecori and P. Mezzina, "A glimpse on big data analytics in the framework of marketing strategies," *Soft Computing*, vol. 22, pp. 325-342, 2018.
- [3] Z. Wang, C. Chong, L. Lan, Y. Yang, S. Ho and J. Tong, "Fine-grained sentiment analysis of social media with emotion sensing," in *IEEE*, San Francisco, CA, USA, 2016.
- [4] T. Young, D. Hazarika, S. Poria and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55-75, 2018.
- [5] M. N. M. I. M. G. C. C. K. L. L. Z. Matthew E. Peters, "Deep contextualized word representations," *arXiv:1802.05365*, 2018.
- [6] M. C. Mlađan Jovanović, "Generative Artificial Intelligence: Trends and and Prospects," *COMPUTER*, no. 10, pp. 107-112, 2022.
- [7] Y. Goldberg, "A Primer on Neural Network Modelsfor Natural Language Processing," *Journal of Artificial Intelligence Research* , vol. 57, pp. 345-420, 2016.
- [8] Y. Kim, Y. Jernite, D. Sontag and A. Rush, "Character-Aware Neural Language Models," in *AAAI Press, Palo Alto*, Phoenix, Arizona, USA, 2016.
- [9] N. S. N. P. J. U. L. J. A. N. G. L. K. I. P. Ashish Vaswani, "Attention Is All You Need," in *The MIT Press*, Long Beach, CA, USA, 2017.
- [10] M.-W. C. K. L. K. T. Jacob Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2019.
- [11] K. N. T. S. I. S. Alec Radford, "Improving Language Understanding by Generative Pre-Training," 2018.
- [12] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," Beijing, China, 2014.

- [13] Y. Han, C. Liu and P. Wang, "A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge," *arXiv preprint arXiv:2310.11703*, 2023.
- [14] A. P. C. S. N. Sridhar Dilip Sondur, "Similarity Measures for Recommender Systems: A Comparative Study," *Journal for Research*, vol. 2, no. 3, pp. 76-80, 2016.
- [15] S. Kumar and S. Kumar, "Ball-tree: efficient spatial indexing for constrained nearest-neighbor search in metric spaces," *arXiv preprint arXiv:1511.00628*, 2015.
- [16] B. Ghogh, S. Sharifian and H. Mohammadzade, "Tree-based optimization: A meta-algorithm for metaheuristic optimization," *arXiv:1809.09284*, 2018.
- [17] O. Jafari, P. Maurya, P. Nagarkar, K. M. Islam and C. Crushev, "A survey on locality sensitive hashing algorithms and their applications," *arXiv:2102.08942*, 2021.
- [18] H. Liu, R. Wang, S. Shan and X. Chen, "Deep Supervised Hashing for Fast Image Retrieval," in *Proceedings of the IEEE conference on computer*, Beijing, 2016.
- [19] N. S. A. R. K. L. S. N. M. M. Colin Raffel, "Exploring the Limits of Transfer Learning with a Unified," *Journal of Machine Learning Research*, no. 21, 2020.
- [20] C. R. N. S. Adam Roberts, "How Much Knowledge Can You Pack into the parameters of a language model," *arXiv e-prints*, 2020.
- [21] S. N. B. B. R. M. Joshua Maynez, "On faithfulness and factuality," *arXiv preprint arXiv:2005.00661*, 2020.
- [22] K. L. Z. T. P. P. M.-W. C. Kelvin Guu, "REALM: Retrieval-augmented language model pre-training," *ArXiv, abs/2002.08909*, 2020.
- [23] B. O. S. M. L. W. S. E. D. C.-t. Y. Vladimir Karpukhin, "Dense passage retrieval for open-domain question answering," . *arXiv preprint arXiv:2004.04906*, 2020.
- [24] E. P. A. P. F. P. V. K. N. G. H. K. M. L. W.-t. Y. T. R. S. R. D. K. Patrick Lewis, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv:2005.11401*, 2020.
- [25] K. L. J. H. A. P. M. B. F. P. P. L. Nelson F. Liu, "Lost in the Middle: How Language Models Use Long Contexts," *Transactions of the Association for Computational Linguistics (TACL)*, 2023.
- [26] X. C. K. M. N. S. D. D. E. C. N. S. D. Z. Freda Shi, "Large Language Models Can Be Easily Distracted by Irrelevant Context," in *International Conference on Machine Learning*, Honolulu, Hawaii, USA, 2023.
- [27] L. v. d. Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, no. 9, pp. 2579-2605, 2008.
- [28] L. B. B. L. Alexander Hermans, "In Defense of the Triplet Loss for Person Re-Identification," *arXiv preprint arXiv:1703.07737*, 2017.