Chatbot Development & Implementation:

Personal and Investment Finance Topics

Jay D. Vaidya

School of Professional Studies, Northwestern University

MSDS 453: Natural Language Processing

Dr. Syamala Srinivasan

August 28, 2021

Developing a Question & Answer Chatbot for Financial Literacy

Introduction & Problem Statement

The topic I've chosen to pursue for the final project in this course is that of developing a question-and-answer chatbot centered on the subject(s) that comprise basic financial literacy. The core topics to be included under the heading of "financial literacy" include the following: financial statements – i.e., income statements, balance sheets and statements of cash flows; core balance sheet components, including assets, liabilities, and shareholder's equity; personal finance, including personal income, budgets, interest rates and annual percentage rates; and investment finance, including stocks, bonds, mutual funds and discount rates. The just-articulated list includes 14 topics, each of which will be broken into smaller subsets of topics, based on the process of sentence tokenization, providing a list of documents to be processed in this project.

My desire to pursue this topic stems from my background in corporate and investment finance; I have found through my years in the profession that financial literacy is often a topic poorly, or scantly, taught in basic primary-school and collegiate programs, outside of those courses and programs specifically tailored to finance and economics. According to the National Financial Educators Council, more than 20 percent of Americans felt they had no trusted source to turn to for advice on financial matters (Valladares, 2020). Given the importance of finance to every aspect of managing one's livelihood in the present and for the future, I believe there is a need to provide additional resources to help educate individuals on the basic topics that comprise financial literacy. By producing a chatbot capable of answering basic questions regarding financial topics, I hope to provide the means by which to marginally improve the financial literacy of English-speaking citizens.

Literature Review

The desire to develop and employ an intelligent agent with which one can interact, and engage in discourse, has been a central goal embedded in the foundations of Natural Language Processing (NLP). In his seminal paper, *Computing Machinery and Intelligence*, published in 1950, Alan Turing posited that the mark of intelligence may even be demonstrated through the interaction of a human reviewer and an

'intelligent agent' in which the human reviewer cannot determine, above a margin of random success, whether or not he/she is indeed interacting with the 'intelligent agent' or another human (Turing, 1950). As the field of study has evolved, and the computational power and statistical sophistication underpinning NLP applications have advanced, this goal has been supplanted by ever-more ambitious targets.

The development and deployment of chatbots, once a novelty, is now a commonplace activity, but the level of sophistication still varies widely. In their paper, *How Should My Chatbot Interact*, Chaves and Gerosa sum up the current state of chatbot development, and its' origins, thusly:

> *Chatbots are computer programs that interact with users in natural language (Shawar & Atwell, 2007). The origin of the chatbot concept dates back to 1950 (Turing, 1950). ELIZA (Weizenbaum, 1966) and A.L.I.C.E. (Wallace, 2009) are examples of early chatbot technologies, where the main goal was to mimic human conversations. Over the years, the chatbot concept has evolved. Today, chatbots can have distinct and diverse characteristics, which has resulted in several synonyms, such as multimodal agents, chatterbots, and conversational interfaces. In this survey, we use the term "chatbot" to refer to a disembodied conversational agent that holds a natural language conversation via text-based environment to engage the user in either a general purpose or task-oriented conversation.* (2021)

Of the numerous approaches to NLP, and chatbot development, three categories of approaches, in particular, have been developed and utilized over time with significant success: term frequency analysis, multi-dimensional text embedding vectors, and the latest, Transformers (self-attention) architecture. The first approach is predicated on term frequency occurrences and often takes the form of term frequency-inverse document frequency (TF-IDF) analysis, a technique, which, when viewed from a high level, scores words in a given document by increasing in value for an increase in the frequency of a word within a document and decreasing the value for an increase in the frequency of a word across a corpus of documents; thus, words that are idiosyncratic to a given document can be highlighted (Varun, 2020). By then coupling TF-IDF matrices, for a set of documents, with statistical measures, such as cosine, which

seeks to measure the angle between two vectors, practitioners can 'select' similar documents that best answer a given query or statement when producing a chatbot (Jurafsky & Martin, Ch. 6, 2020).

The second approach, that of embedding vectors, was proposed and popularized by Google researchers in a 2013 paper titled *Efficient Estimation of Word Representations in Vector Space*. In short, researchers observed that ratios of word-word co-occurrence probabilities have the potential for encoding a form of meaning on textual data (Pennington, Socher & Manning, 2014). Importantly, the intuition underlying this approach could be expanded, at scale, and trained embedding vectors, developed on corpora containing millions of documents, have been made available to the open-source community. One such open source embedding model widely available is the Global Vectors for Word Representation (GloVe) maintained by researchers at Stanford University. Using trained vector representations, textual data, paired with deep learning algorithms, served as an important avenue of advancement for NLP and chatbot development, in particular.

The third approach, among the latest, and most heralded, developments in NLP, is the Transformer self-attention architecture. The Transformer architecture combines the advances made available through text embedding vectors with a new neural network architecture that can displace previous methods such as recurrent, gated and convolutional networks with a self-attention mechanism (Vaswani et. al, 2017). According to Vaswani et al., "An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key" (2017). In such a manner, the Transformer architecture permits for models built on attention-based mechanisms to better store and retrieve long-range dependencies inherent in a given piece of text, enabling more accurate NLP undertakings such as coreference resolution and handling language ambiguities.

As with embedding vectors, Transformer-based models have been shown to improve and scale with additional training, particularly those engaged in a niche lexical undertakings, such as question and answering. To this end, one particularly important advancement has been the development and publishing

of the Stanford Question and Answering Dataset 2.0 (SQuAD 2.0). Originally developed to promote machine training in Reading Comprehension (RC), the SQuAD dataset "…contains 107,785 question-answer pairs on 536 articles, and is almost two orders of magnitude larger than previous manually labeled RC datasets such as MCTest" (Rajpurkar, Zhang, Lopyrev, & Liang, 2016). By forcing a given model to predict the span of text, from a given document, that best answers a posed query, models trained on the SQuAD dataset, and based on the Transformer architecture, have set a new bar for question-and-answer chatbots, as will be further detailed in the results section of this paper.

Data Analysis – Preparation, Exploration & Visualization

The primary source of text used in generating a corpus for this project is Wikipedia. Wikipedia represents a source of user-generated and submitted content on specific topics, which is further curated by peers and professionals involved in the maintenance of site integrity (Wikipedia, 2021). Based on my review of topics I've chosen to include in my corpus, and which are covered in Wikipedia articles, the quality and veracity of the texts are quite high. This assessment is based on my own personal judgements as a financial practitioner with more than 10 years of practical experience and two master's degrees in relevant fields – one in business administration and another in corporate finance and risk management.

Upon review of the structure of the textual data contained on Wikipedia, prior to any processing, I noted that the data presented on a specific topic is generally presented in sentence and paragraph form, along with sub-topic headings, page breaks and some graphics and tables that needed to be discarded. The average length of documents, for the topics I've reviewed, stretches to several pages on standard 8.5 by 11-inch paper. In order to access the data present on Wikipedia, I made use of the Python 'wikipedia' module, which provides an easy-to-use interface for making calls through the Wikipedia application programming interface (API).

For my work in this project, I determined to make calls on the Wikipedia API to obtain the top two highest ranked Web page matches, as determined by the Wikipedia 'term search' function of the Python Wikipedia package, on each topic I sought to include in my question-and-answer chatbot. In the

end, I queried 25 Web pages for content that constituted the corpus of text used as the primary source material for this project.

The data returned from the Wikipedia API calls required somewhat extensive pre-processing and cleaning. The text preprocessing occurred in two distinct phases. During the first phase, I began by manually reviewing a subset of the collected documents in order to determine the elements that may need to be removed from the corpus prior to engaging in sentence tokenization. I discovered that most documents were presented in proper sentence form, but that line and page-breaks and section and subsection headings were included in the text, and would skew the results of any sentence tokenization efforts. Using word-level tokenization, I separated the words in the respective documents and eliminated these elements from the corpus using Regular Expressions (RegEx).

Having eliminated the most glaring issues within the corpus, I re-joined the tokenized text and then engaged in the second phase of textual cleaning for the corpus. Phase two of the text cleaning began with sentence tokenization of the text, which I needed to perform before further cleaning steps as the punctuation that is often eliminated in text preprocessing is a vital element for sentence tokenization methods. Each sentence was then word-tokenized to permit the application of the following cleaning steps: elimination of punctuation, elimination of 'stopwords', replacement of semantically equivalent words I identified during my initial document review, changing all words to lower casing, and lemmatization. Once cleaned, each word-tokenized document was rejoined, creating a list of 3,041 cleaned, preprocessed sentences. I then eliminated all sentences with five or fewer words, seeking to avoid superficial content being surfaced when producing the chatbot, leaving 2,589 cleaned sentences that constituted my corpus of text. For the purposes of this project, each of the cleaned sentences represented a document containing textual information that could conceivably contain the most appropriate response to a given query on the topics comprising financial literacy.

The final step in my preparation and exploration of the corpus compiled for this project, prior to engaging in the chatbot development, was to explore the semantic relations in the dataset through entity-

relation mapping and knowledge graph production, primarily accomplished using Python's 'spacy' module. The intent of this exploratory analysis was to probe the semantic structure of the data to determine if the structure was appropriate for the development of a chatbot; in other words, was the text presented in such a way as to express important relations between key terms that could then be queried by a chatbot. By visually inspecting the relations of key terms, including 'interest rate' and 'budget', I was able to determine that the corpus of text generated for this project contained a rich set of semantic relations, linking together nuanced financial terms that could, conceivably, be queried by a chatbot (See appendix A).

Research Design & Modeling Methods

For this project I chose to utilize four modeling approaches, differing in their sophistication and difficulty to implement, with which to compare and contrast results in order to determine the best model for the production of a financial literacy chatbot. The four methods are as follows: TF-IDF, Doc2Vec (available through the 'Gensim' Python module), 100-dimensional pre-trained GloVe embedding vectors, and application of a Transformer-based model architecture – specifically, a Bi-directional Encoder Representations from Transformers (BERT) model trained on the SQuAD dataset.

Additionally, for each of the methods employed in this project, I also made use of a combination of defined response functions, such as a 'greeting' function, and chained if-else statements to detect and respond to certain expected contexts and phrases within a given conversation between the chatbot and a user. Expected contexts included typical greetings, chatbot responses when poor results were surfaced in response to a specific query, and concluding remarks that would end a chatbot session (See Appendix B). In applying such an approach, I hope to both enhance to ease of use for potential users of the chatbot, as well as to impart humanoid characteristics that could put users at ease and help facilitate ongoing dialogue.

With respect to the methodologies applied to intake user queries as input and generate responses for the chatbot, the first method I applied was that of TF-IDF, coupled with cosine similarity scoring,

which primarily made use of the Python ‘NLTK’ and ‘sklearn’ packages, to find the closest document matches for a given query. To begin, I fit the ‘sklearn’ TF-IDF Vectorizer to my cleaned corpus of text and produced a TF-IDF matrix for the corpus. Next, I prepared a function such that the input to a given user query would be vectorized using the already-fit TF-IDF Vectorizer; the resulting TF-IDF vector for the user query was then concatenated onto the end of a copy of my corpus TF-IDF matrix. Given that cosine similarity can return a score, representing the angle of difference of two pairs of vectors for multi-dimensional vector pairs, I made us of the ‘sklearn’ function that then computed the cosine similarity between the user query and every other document vector in my corpus. Finally, the document similarity scores were sorted so as to place the most similar documents together; by extracting the index ID of the document deemed to be most similar to the user query, I could then make a call to my original corpus, using the document ID, to return the document with the highest similarity to the user’s query, using that document as the response returned by the chatbot.

The second method I employed in this project was that of generating Doc2Vec 100-dimensional text embedding vectors to represent each document in the corpus as well as user queries, as they were generated. The primary difference in this approach, as compared to TF-IDF vectorization, is that embedding vectors attempt to distill and encode some form of semantic meaning into the vector data by using word co-occurrence to produce the embedding vectors. Once the document vectors have been produced, however, the procedure for calculating document similarities between user queries and the corpus was the same as that articulated in the previous methodology – namely, using cosine similarity scores. In order to assess the likely success of this approach, I produced a plot of silhouette scores for a range of cluster sizes from two to 20, with the optimal score achieved at approximately seven clusters. I then used principal component analysis to generate a two-dimensional representation of the embedding vectors for the documents in my corpus and plotted the results using seven clusters, also noting the centroid of each cluster (See Appendix C). Important to note in this approach, however, is that text embedding vectors are sensitive to the amount of data upon which they are trained, producing

demonstrably improved results on larger versus smaller corpora, which will be illustrated later in the results section of this report.

The third approach applied to this project also made use of text embedding vectors, but this time using pretrained, open-source vectors trained on a corpus containing millions of documents. Specifically, I employed the GloVe 6 billion tokens, 400,000 vocab, 100-dimension embedding vectors. Unlike the Doc2Vec embedding vectors generated in the previous approach, however, the GloVe vectors represent individual words, not documents; in order to then use these pre-trained embedding vectors for document comparisons, I generated a function that calculated a summation of the individual vector embeddings for each word in a given document, producing one final score for each of the 100 dimensions of each document in the corpus – user queries were also treated in the same manner. In order to do so, the function began by creating a sparse 100-dimension vector, populated with all zeros, for each document in the corpus; then, a loop was used to search the pre-trained embeddings for each word in a given document, if the word was found in the pre-trained embeddings, the values for each dimension were added to the values for the current document-level embedding, and if the word was not found, zero was added to each dimension, leaving the document-level embedding unchanged. Finally, as with the previous two approaches, cosine similarity scores were calculated between a given user query and the documents in the corpus to determine the best chatbot response to a proposed query.

The fourth and final method I employed in this project is that of the deep learning model known as a Bidirectional Encoder Representations from Transformers (BERT). Pretrained models such as BERT can be instantiated through freely available Python packages maintained by organizations such as HuggingFace. The specific BERT model I applied was that of the ‘BERT base-cased squad2’, which is the base BERT model that has had additional training on the SQuAD dataset to best enable question-and-answering tasks. The model was then applied using PyTorch to tokenize the questions posed and determine the model’s response to a given query. Unique to this approach, it should be noted, is that unlike the previous three methodologies I detailed, the BERT model was not trained on a specific corpus I

had generated; importantly, however, the span of text selected as the most appropriate response to a given question was achieved by pointing the model to the document within Wikipedia that most likely contained a suitable answer. Making use of the Python 'wikipedia' module, each question posed to the BERT model was first converted into a search term applied to the 'wikipedia' search function; the document returned by this search, as ranking most representative of the search, was then selected and fed into the BERT model using the 'encode_plus' function from the BERT model tokenizer, which would concatenate the posed question with the selected document, and truncate the document after 512 tokens to match the original training set used to produce the model. Finally, the model would return a response to the query by first calculating the tokens that were most likely to begin and end the response to a given query and then, beginning with the first words of the response, concatenating each subsequent word that was probabilistically most likely to follow the previous word until the ending token was reached. One of the striking features of this approach is that, while more than capable of producing responses to questions regarding financial literacy, it is in no way confined to producing coherent responses solely on that topic. Any topic of discussion with which a Wikipedia article exists can be queried, selected and parsed by the BERT model to produce answers for specific queries. In other words, the BERT model, as opposed to the three prior approaches, is an open-domain, as opposed to closed-domain, question-and-answer chatbot.

Results

The results of my analysis of the methodologies detailed above for the production of a question-and-answer chatbot to promote financial literacy demonstrate a clear hierarchy with respect to ranking of the quality of responses produced by the chatbot. The methodology rankings, in descending order of quality, are as follows: BERT Transformers model, GloVe 100-dimensional pre-trained embedding vectors, TF-IDF, and lastly Doc2Vec (See Appendix D). Methodology rankings are based on my assessment of qualitative considerations including the accuracy of the response, the syntax used as compared to what could be expected from a human respondent, and the breadth versus concision (i.e., was too much or too little information provided) of the response.

To approach the respective methodologies in reverse order of their rankings, I would first point to the deficiencies of the Doc2Vec approach. Of the four methodologies applied to this task, Doc2Vec was the only that produced, on balance, more responses that could be termed non-sensical, or at the very least outside of the domain of the question being posed, as compared to the other methodologies. It is my belief that this occurred due to the limited nature of the training corpus. Though not miniscule, the corpus, comprised of 2,589 cleaned sentences, simply did not provide enough data upon which to fit the embedding vectors. Words used infrequently lacked the appropriate contextual clues, provided by co-occurring words, to embed enough information to be of use, as such, spurious correlations were likely to occur and materially impact results. It is my belief that a corpus of sufficient size, perhaps an order of magnitude larger than that used, would have provided significantly better results.

In assessing both the GloVe 100-dimensional embedding and TF-IDF methodologies, I note that both made a common mistake in response to question three (i.e., what is a mutual fund?) (See Appendix D). Both approaches returned a response detailing some of the major operators within the mutual fund industry, demonstrating a clear semantic misunderstanding; instead of answering what the entity 'mutual fund" *is*, the responses answered a question related to what *are* some of the mutual funds that can be employed by investors – this was purely due to similarity scores of the respective vector representations with the posed query. Further scrutiny of the two approaches, however, demonstrates that the pre-trained embedding vectors are capable of imparting some additional semantic knowledge into the question-and-answer functions as the responses generated via this method to questions four and five are much more accurate and closer to what I would consider an ideal response.

Finally, with respect to the best performing methodology, that of the BERT model trained on the SQuAD dataset, it can be noted that both the responses, as well as the syntax of the responses, are accurate and appropriate for each of the five questions posed. Notably, the Transformers based model was able to make sense of question three, arguably the most difficult question posed to the chatbot among my selections, returning the definition of a mutual fund as opposed to a list of prominent mutual fund operators. Furthermore, as mentioned previously, unlike the other methods used in this project, the BERT

model is an open-domain model and can be applied to any topic for which Wikipedia provides a reasonable corpus of text from which to draw a response. The Transformers architecture has clearly demonstrated the power of using bi-directional encoding and self-attention to produce greater context from which to generate chatbot responses.

Recommendations & Next Steps

Given the foregoing, it is my recommendation that a financial literacy chatbot should be produced using a model, such as BERT, that is generated using the Transformers architecture and trained on the SQuAD dataset. In order to further enhance the model, however, an uncased version of the model should be instantiated, with the intent of further training the model on a question-and-answer dataset specifically related to the domain of personal and investment finance. Open-domain chatbots, such as that produced and used in this project, can suffer from insufficient context regarding the nuanced terminology and syntax used in a specific field of study. By generating a new question-and-answer dataset upon which to further train a model, one can hope to refine the already-powerful tools at hand to produce a genuinely helpful chatbot to assist in building financial literacy.

Furthermore, I would also recommend that additional work be undertaken, prior to publishing this chatbot, to apply RegEx to the questions entering the chatbot in order to standardize the questions to a format that can easily be digested by the model. Specifically, by developing sets of keywords associated with specific types of queries – such as navigational (those seeking information on the location of some item or fact), informational (those seeking a direct response to open or close-ended query), or transactional (those seeking information on how to execute an intended action) – one could produce consistent, coherent question formats to be fed into the model in order to provide the best probability of returning a suitable answer.

In the end, while I would strongly recommend that any chatbot being produced be generated using Transformers architecture, I do believe that suitable replacements could be achieved with both TF-IDF and pretrained GloVe embedding vectors, both of which would require fewer computational resources and a more modest development budget.

References

Chaves, A. P., Gerosa, M. A. (2021). How should my chatbot interact? A survey on social characteristics in human-chatbot interaction design. *Journal of Human-Computer Interaction, 37(8)*, 729-758. https://doi.org/10.1080/10447318.2020.1841438

Jurafsky, D. & Martin, J. H. (2020). *Speech and language processing: 3rd Edition*. Pearson: Prentice Hall. Upper Saddle River, NJ.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. Google Research.

Pennington, J., Socher, R. & Manning, C. D. (2014). *GloVe: Global vectors for word representation*. https://nlp.stanford.edu/projects/glove/

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ questions for machine comprehension of text*. Computer Science Department, Stanford University.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59(236)*, 433-460. Oxford University Press.

Valladares, M. R. (2020). *Financial literacy is more important than ever*. Forbes. https://www.forbes.com/sites/mayrarodriguezvalladares/2020/12/15/financial-literacy-is-more-essential-than-ever/?sh=40ea5516304f

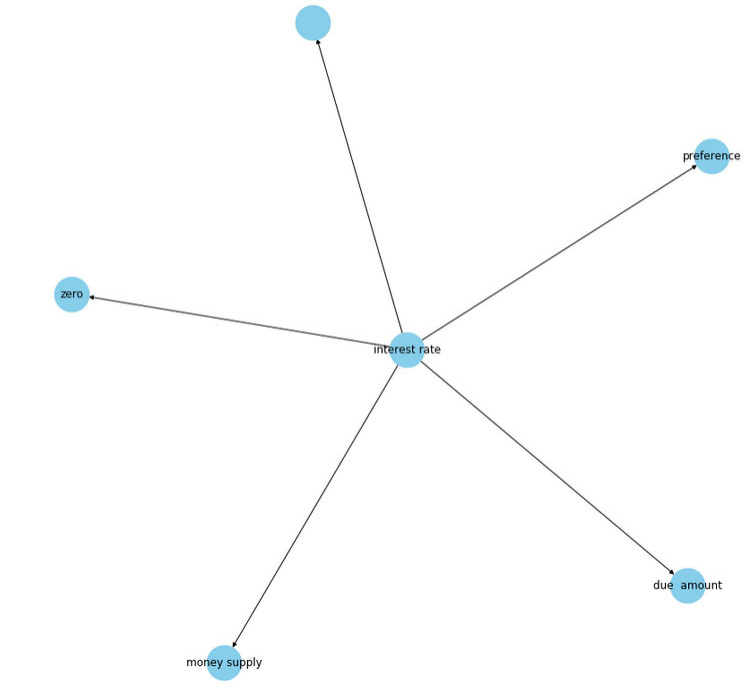
Varun. (2020). *The quantitative value of text, tf-idf and more*. Medium. https://medium.com/analytics-vidhya/the-quantitative-value-of-text-tf-idf-and-more-e3c7883f1df3

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA.

Wikipedia: Database Download. (2021, July 9). Wikipedia. Retrieved July 29, 2021, from https://en.wikipedia.org/wiki/Wikipedia:Database_download

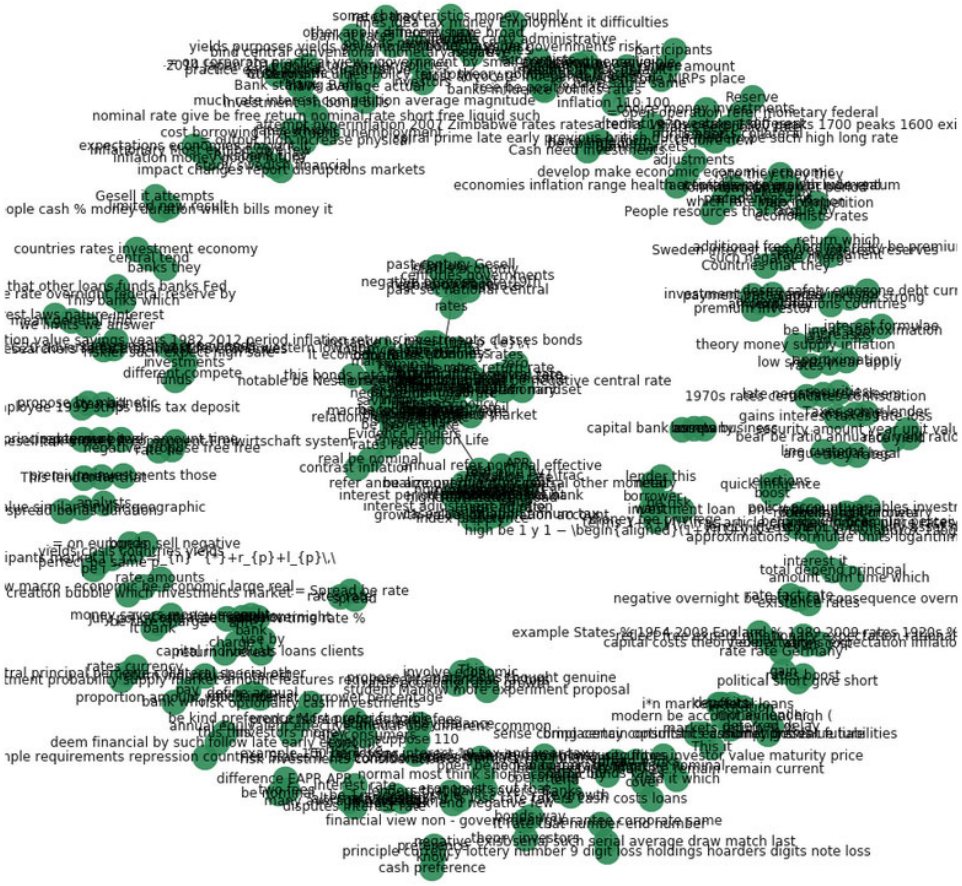
Appendix A-1:

Example Entity-Relation Knowledge Graphs



Appendix A-2:

Example Entity-Relation Knowledge for All Relations in ‘Interest Rate’ Wikipedia Page



Appendix B:

Chatbot Dialogue Design – Greeting, Response & Conclusion

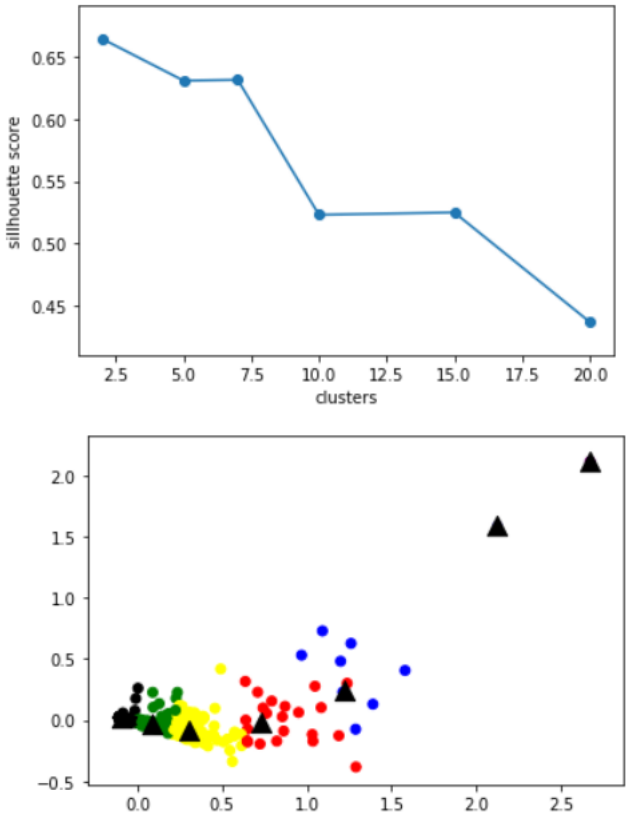
```
# Generate some standard greeting inputs to which the chatbot should respond; also generate typical responses from which the chatbot can randomly select, for a given greeting
greeting_inputs = ("hey", "good morning", "good evening", "morning", "evening", "hi", "whatsup")
greeting_responses = ["hey", "Hey there", "*nods*", "hello, what can I answer for you today?", "hello", "Welcome, what question(s) do you have for me?"]

def generate_greeting_response(greeting):
    for token in greeting.split():
        if token.lower() in greeting_inputs:
            return random.choice(greeting_responses)

# Below code originally sourced from: https://medium.com/analytics-vidhya/building-a-simple-chatbot-in-python-using-nltk-7c8c8215ac6e

def response_tfidf(user_response):
    robo_response=''
    sent_tokens = flat_cleaned_sent_tokens.copy()
    sent_tokens.append(user_response)

    TfidfVec = TfidfVectorizer(stop_words='english')
    tfidf = TfidfVec.fit_transform(sent_tokens)
    vals = cosine_similarity(tfidf[-1], tfidf)
    idx=vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]

    if(req_tfidf==0):
        robo_response=robo_response+"I am sorry! I don't understand your question, can you try to pose it in a different manner?"
        return robo_response
    else:
        robo_response = robo_response+sent_tokens[idx]
        return robo_response

    sent_tokens = flat_cleaned_sent_tokens.copy()
```

```
# Generate functions that handle the core dialogue and response mechanisms for the chatbot

continue_dialogue = True
print("""Hello, I am FinanceBot. You can ask me any question regarding personal finance, I will do my best to answer you. Once you're done, please type 'bye'""", ":")
while(continue_dialogue == True):
    human_text = input()
    # Used the phase two clean doc function, from above, to distill user questions to the essence, as best as possible
    human_text = clean_doc_phase_two(human_text)

    if human_text != 'bye':
        if human_text == 'thanks' or human_text == 'thank you very much' or human_text == 'thank you':
            continue_dialogue = False
            print("FinanceBot: You are welcome")
        else:
            if generate_greeting_response(human_text) != None:
                print("FinanceBot: " + generate_greeting_response(human_text))
            else:
                print("FinanceBot: ", end="")
                print(response_tfidf(human_text))

    else:
        continue_dialogue = False
        print("FinanceBot: Good bye and take care of yourself.")
```

```
Hello, I am FinanceBot. You can ask me any question regarding personal finance, I will do my best to answer you. Once you're done, please type 'bye' :
 Hi there
FinanceBot: hello, what can I answer for you today?
 Can you tell me what is an interest rate?
FinanceBot: The effective interest rate (EIR), effective annual interest rate, annual equivalent rate (AER) or simply effective rate is the interest rate on a loan or financial product resta
ed from the nominal interest rate and expressed as the equivalent interest rate if compound interest was payable annually in arrears.
 How can I make a personal budget?
FinanceBot: A personal budget or home budget is a finance plan that allocates future personal income towards expenses, savings and debt repayment.
 What is a mutual fund?
FinanceBot: Some of the major competitors for SBI Mutual Fund in the mutual fund sector are Birla Sun Life Mutual Fund, Kotak Mutual Fund, HDFC Mutual Fund, ICICI Prudential Mutual Fund, Rel
ance Mutual Fund, UTI Mutual Fund & Axis Mutual Fund.
 What components are included in an income statement?
FinanceBot: A comprehensive income statement involves those other comprehensive income items which are not included while determining net income.
 What is an asset?
FinanceBot: .= Wasting asset .= A wasting asset is an asset that irreversibly declines in value over time.
 Thank you, bye
FinanceBot: I am sorry! I don't understand your question, can you try to pose it in a different manner?
 bye
FinanceBot: Good bye and take care of yourself.
```

Appendix C:

Doc2Vec 100-dimensional Embedding Vectors – Clustering Analysis



Note: 1) N-Clusters = 7 ; 2) Black triangles represent cluster centroids

Appendix D:

Ranked Results by Methodology – Questions & Responses

Ranked Results By Methodology

- Five Questions Posed to Each Chatbot
  - **1)** Can you tell me what is an interest rate?
  - **2)** How Can I make a personal Budget?
  - **3)** What is a mutual fund?
  - **4)** Which components are included in an income statement?
  - **5)** What is an asset?

- BERT & Transformers: Responses – Rank # 1
  - **1)** The effective interest rate (EIR), effective annual interest rate, annual equivalent rate (AER) or simply effective rate is the interest rate on a loan or financial product
  - **2)** A personal budget or home budget is a finance plan that allocates future personal income towards expenses, savings and debt repayment. Past spending and personal debt
  - **3)** an open - end professionally managed investment fund
  - **4)** to show managers and investors whether the company made money (profit) or lost money (loss) during the period being reported
  - **5)** any resource owned or controlled by a business or an economic entity. It is anything (tangible or intangible) that can be used to produce positive economic value

- GLoVE Embeddings: Responses – Rank # 2
  - **1)** The nominal interest rate is the rate of interest with no adjustment for inflation.
  - **2)** Past spending and personal debt are considered when creating a personal budget.
  - **3)** Some of the major competitors for SBI Mutual Fund in the mutual fund sector are Birla Sun Life Mutual Fund, Kotak Mutual Fund, HDFC Mutual Fund, ICICI Prudential Mutual Fund, Reliance Mutual Fund... (cont.)
  - **4)** A business entity adopting IFRS must include: a statement of comprehensive income or two separate statements comprising: an income statement displaying components of profit or loss...(cont.)
  - **5)** Assets can be divided into e.g., current assets ,and fixed assets, often with further subdivisions such as cash, receivables and inventory.

- TF-IDF: Responses – Rank # 3
  - **1)** The effective interest rate (EIR), effective annual interest rate, annual equivalent rate (AER) or simply effective rate is the interest rate on a loan or financial product restated from the nominal interest rate and expressed as the equivalent interest rate if compound interest was payable annually in arrears.
  - **2)** A personal budget or home budget is a finance plan that allocates future personal income towards expenses, savings and debt repayment.
  - **3)** Some of the major competitors for SBI Mutual Fund in the mutual fund sector are Birla Sun Life Mutual Fund, Kotak Mutual Fund, HDFC Mutual Fund... (cont.)
  - **4)** A comprehensive income statement involves those other comprehensive income items which are not included while determining net income.
  - **5)** .= Wasting asset .= A wasting asset is an asset that irreversibly declines..(cont.)

- Doc2Vec Embeddings: Responses – Rank # 4
  - **1)** The amount going toward the principal in each payment varies throughout the term of the mortgage.
  - **2)** Rather than work with r and r' as separate variables, we can assume that they are equal and let the single variable r denote their common value.
  - **3)** It is expressed as a percentage of the average market value of the portfolio's long-term securities.
  - **4)** The meeting was the first of four seminars in which France would outline his vision of an organized group of race car drivers.
  - **5)** Early in NASCAR's history, foreign manufacturers had shown interest in entering the series; the British car manufacturer, MG, found a few of its vehicles entered, with some placing.