Final Project – Creating Chatbot

Nadeem Patel
MSDS 453 - Section 56 (Winter 2023)
Northwestern University, Natural Language Processing
03/11/23

Introduction and problem statement

Chatbots leveraging natural language processing (NLP) can understand users' queries based on context, assess them, and respond to the users in natural language (Chotia 2022). NLP is based on deep learning (DL) that enables computers to acquire meaning from user inputs, and in context of chatbots, NLP analyzes the intent of the input and creates responses based on context similar to a human. Based on benchmarks, the best performing methods for every NLP task relies on deep learning, including the Transformer architecture and BERT (Makadia 2019).

The "traditional" machine learning method would be to build a chatbot using a linear model on the term frequency-inverse document frequency (TF-IDF) approach (Feldges 2022). TF-IDF is a technique used in information retrieval (IR) and can quantify the importance of relevance of string representations, including words, phrases, and lemmas, in a document within a corpus. While TF-IDF provides simplicity and ease of use, it lacks the ability to carry semantic meaning. Such a method considers the importance of words based on how it weights them but cannot necessarily derive the context of the words (Simha 2021). While TF-IDF provides a simple solution, long shot-term memory (LSTM) networks, a type of recurrent neural network (RNN), can learn order dependence in sequence prediction problems. That is necessary in complex problem domains like machine translation, speech recognition, and other complex problems (Brownlee 2017). A transformer model is another type of neural network that learns context and meaning by tracking relationships in sequential data like words in a sentence. Similar to other neural networks, transformer models are large encoder and decoder blocks that process data and tag data elements coming in and out of the network. Attention queries are usually performed in parallel by calculating a matrix of equations in what is considered multi-headed attention (Merritt 2022).

This study uses an ontology graph built on a qualitative approach as a reference and creates a chatbot using three different methods in order to determine which approach is deal. TF-IDF, considered a “traditional” approach, is used as a baseline method. LSTM is used as a deep learning approach and the SentenceTransformer Python framework is also used to build a third model. The outputs of the models are compared to determine how each approach is able to read the text provided as data, understand the user’s question, and provide a sufficient response that is meant to be contextual and accurate. Additionally, the Chat GPT-3 model is used to answer questions in order to understand how an advanced model trained on extensive data compares to the models built in this study. The data used in this study is focused on World War I history that is derived from Wikipedia.

Literature review

The development of chatbots is common in various industries, including the medical field. A study from the *Emerging Trends in ICT for Sustainable Development* built a chatbot using NLP and TF-IDF algorithms for text understanding. According to the study, the progress and advancements of machine learning, deep learning, and artificial intelligence (AI) has given machines the ability to impersonate humans. NLP-based conversational software agents, also known as chatbots, have provided patients with health care consultation more easily with the appropriate doctor while having health problems. The study proposes an AI-based medical chatbot that can detect the ailment and also provide necessary details about any condition. Ultimately, the goal is to minimize healthcare costs and improve the approachability to medical knowledge. In order to develop the solution, the study performs a detailed survey on recent literature and examines publications that are related to chatbots. A hybrid architecture in the study is based on deep learning models like NLP and the TF-IDF algorithm (Soufyane, Abdelhakim and Ahmed 2021). Similar to the study that focuses on medical context, this report focuses on specific context, World War I, in order to answer questions related to the particular topic. However, this study does not focus on using TF-IDF as part of a hybrid model and uses it as a baseline model instead.

According to a study from the *International Journal of Innovations in Engineering and Technology* (IJIET), a basic sequence-to-sequence model consists of two RNNs: an encoder that processes the input and a decoder that generates the output. Sequence-to-sequence is generally used with attention-based that enables the decoder more direct access to the input. Such a model has been used for various NLP tasks like alignment, translation, and summarization. The encoder in used for the model in this study processes an utterance by human and the decoder produces the response to that utterance. To make the bot speak like a specific character, the vector embeddings are trained for different characters in hopes that the embeddings would be able to encode information and style of speech of the characters. A chatbot represents the natural evolution of a Question Answering (QA) system leveraging NLP, according to the study. Developing responses to questions in natural language is a typical example of NLP applied in various enterprises' end-use applications (Dhankhar 2018). In this report, the focus of the chatbot is to provide responses to users who are interested in learning about World War I. Like the study from the IJIET, this report discusses the use of LSTM to create a chatbot. While the IJIET study provides insight into other possible approaches utilized to implement RNN and LSTM, the LSTM model discussed in this report does not necessarily use sequence-to-sequence. Instead, it uses an approach where the input text is converted to a fixed-length sequence of integers using padding and then fed into an LSTM neural network.

Like LSTM, a transformer model provides the ability to transform one sequence into another with the use of an encoder and decoder, but it does not imply recurrent networks (Maxime 2019). A study states that RNN models are often used to determine sequence-related problems like questions and answers, an approach generally considered as sequence-to-sequence (seq2seq) learning. To reinforce the seq2seq model performance, attention mechanism is added to the encoder and decoder, and then the transformer model introduces itself as a high-performance model with multiple attention mechanism for solving sequence-related dilemma. In the study, a transformer model was applied for a Bengali general knowledge chatbot based on the Bengali general knowledge QA dataset. To check for comparison of the transformer model performance, the seq2seq model was trained with attention, which provided accurate results (Masum et al. 2021). Like the study, this report analyzes the use of a transformer to develop a chatbot. More specifically, the SentenceTransformer framework is used for cosine similarity calculations, and since it is based on a transformer architecture, the model inherently includes an attention mechanism.

Data

The data used in this study was scraped from Wikipedia and placed into a .txt file. The file was manually cleaned to remove spaces between paragraphs. Additionally, headings and captions were removed from the corpus. Before preprocessing, exploratory data analysis (EDA) shows that the corpus is made up of seven lines with 24,570 words and 155,245 characters [Figure 1.0]. During the preprocessing stage, the data was tokenized to break down the text into smaller units (tokens), and stopwords and punctuations were removed since they would be irrelevant to the analysis and do not carry much meaning to the text in terms of analysis from an

NLP perspective. Additionally, the tokens were turned into all lowercase words and lemmatized to convert words to their base or dictionary form (lemma).

After performing EDA, the total number of lines increased to 1,050, but the total number of words decreased to 14,973, and the total number of characters decreased to 111,406 [Figure 1.1]. The top 10 most frequent words appeared to be relevant to World War I, including words like "war," "german," "british," "russian," and "battle." However, there were also black terms such as "``" and " '' ," which means the preprocessing may not have been highly successful at removing unnecessary context within the corpus [Figure 1.2]. When looking at the processed sentences, however, it appears that majority of the sentences were successfully processed to provide a corpus that would be sufficient for modeling. Majority of the sentences appeared to go through tokenization, removal of stopwords and punctuations, lowercasing, and lemmatization [Figure 1.3].

Research design and modeling methods

Before building models for chatbots, an ontology was created to determine a structured way of representing the knowledge and relationships between concepts regarding World War I. The ontology was built based on a qualitative approach, which means the entities and relationships were manually selected based on the main themes of the topic. In terms of World War I, the main ideas were regarding the countries involved, deaths and outcomes of the war, causes of the war, and the main dates (beginning and end of war). The ontology did not focus on relationships outside of the clusters that were made in order to prevent complexity that could lead to responses that are not viable. Furthermore, the ontology graph was used to develop the questions that were straightforward, which would allow the models to be analyzed in comparison to the ontology graph [Figure 1.5].

The baseline model used to develop a chatbot was TF-IDF, which is used to vectorize the processed sentences. Specifically, the TfidfVectorizer class from scikit-learn is used to create a matrix of word vectors based on the frequency of each word in every sentence and weighted based on the rarity of the word throughout the corpus. The matrix is a numerical representation of the sentences [Figure 1.4]. A function takes a user input query, preprocesses the query using the same techniques as the sentences in the matrix and generates TF-IDF scores for the sentences in the matrix. The cosine similarity between the query vector and the sentence vectors is calculated, and the sentence with the highest score is selected as the response. This baseline method attempts to put value on words, similar to the ontology, in order to determine the ideal response to the user's question.

The LSTM model pads the sequences, the list of tokens that occur in a particular order in a sentence, to ensure that the sentences have the same length. This helps to create batches of data to input into the neural network for training. The model is defined using the Sequential() function and includes four layers. An embedding layer is added to help map the input word index to dense vector representations, an LSTM layer is added to understand the sequential patterns in the text data, a dropout layer is added to reduce overfitting, and finally, a dense layer with softmax activation is added to output the probability distribution of the next word in the sequence. The query taken as an input and preprocessed before a predicted sequence is determined, which is then converted back to text and returned as a response. This approach looks to extract key words that may be considered the main focus of the question being asked and attempts to formulate relationships between the words.

The final model built for this study is with the use of SentenceTransformer, which has a pre-trained model meant for sentence cosine similarity calculations. The model generates sentence embedding using the SentenceTransformer model, and then calculates the cosine similarity between the sentence embeddings of the query input and the sentences within the corpus. Finally, the sentence with the highest cosine similarity with the question is returned as the chatbot's response. This approach, while similar to the LSTM model, uses a pre-trained model. More specifically, multi-qa-MiniLM-L6-cos-v1 is used, which is a model that maps sentences and paragraphs to a 384-dimensional dense vector space, is designed for semantic search, and trained on 215 million (question, answer) pairs from diverse sources (Hugging Face n.d.).

After the three models were built, the results were compared to Chat GPT-3, a model developed by OpenAI that allows the chatbot to understand and generate human-like natural language with accuracy and fluency. It is trained on 175 billion parameters and has the ability to process billions of words in a single second (Wampler 2023).

Results

The ontology graph used as a reference in this study was ultimately built on a qualitative approach in order to determine the main entities and relationships relevant to World War I. The clusters were not connected to build complex relationships since the queries used as input are meant to be simple in this study [Figure 1.5].

Results show that the TF-IDF model is able to formulate sentences while LSTM failed to provide coherent sentences. The SentenceTransformer model was the closest to providing relevant information, but none of the models compared to the results of Chat GPT-3, which was able to respond with answers that were contextual and accurate [Figure 1.6].

Analysis and interpretation

Due to the nature of building chatbot models, the analysis is conducted qualitatively. Rather than splitting the data in training and testing datasets, the entire corpus was used. Furthermore, the TF-IDF model and the SentenceTransfomer model use cosine similarity, so logically, performance metrics like accuracy would not be logical. The goal of such algorithms is to set the desired results against the models' performance, so in order to determine the ground truth, the analysis can be done by determining whether the responses are feasible. In this case, TF-IDF was able to provide responses in coherent sentences, but those responses lacked relevant information to the question being asked. Based on the results, it can be assumed that the model does not consider words in the query or sentences and does not understand the meaning of the words [Figure 1.6]. As mentioned, performance metric scores would not make much sense in this situation, but if test questions and answers were run through the model, the accuracy, precision, recall, and F1 scores appeared to be zero [Figure 1.7]. While this may not be a completely accurate representation of the model, those scores do reflect the lack of context and relevance in the responses.

Similar to TF-IDF, which does not consider context of queries or sentences, the LSTM model also does not consider context for the question or response. Additionally, the model appears to have limited vocabulary, which is reflected in the inability to formulate full sentences. This also helps to show the process that the model attempts to perform, which is to predict a sequence rather than using context. This may also be an indication of overfitting [Figure 1.5]. While the results were not close to being correct, results from performing performance metrics analysis show an accuracy rate of 89 percent [Figure 1.8]. When looking at the lemmatized sentences, the data indicates that the responses should be full responses [Figure 1.3]. However, since this model attempts to predict the next word each time, the accuracy may be indicating that the labels set in this model are not accurate to begin with, and therefore, the predicted labels used for the responses are not accurate as well. The results show that the model tries to extract key words from the queries and tries to use those words to develop a response but is unsuccessful. While the approach is logical and similar to the ontology graph, which was built based on a qualitative approach and extracting key words for entities and building relationships, this model fails to utilize context.

When looking at the SentenceTransformer results, it is clear that this model provided the most ideal responses in terms of context and coherence. For example, when asked, "When did World War I begin?," the response was "world war i or the first world war 28 july 1914 – 11 november 1918 often abbreviated a wwi wa one of the deadliest global conflict in history." While the sentence may not be structured properly, the response provides a timeframe of the war, as well as additional context about the war. Similarly, it provided a reasonable answer when asked about casualties. However, when asked about the countries that fought in the war and about those who fought for the Allies, the response was the same, which shows it fails to provide responses that are highly specific [Figure 1.6]. However, this also indicates that the corpus may not be large enough with enough details that dive further into more specific topics within World War I.

All three models failed to match Chat GPT-3 in terms accuracy or context. Chat GPT-3 was able to successfully provide shorter or longer answers as necessary. For example, when asked about when the war began, the model provided a straightforward response unlike like the SentenceTransformer model, which added extra context. On the other hand, Chat GPT-3 did provide relevant information that further provided context to a question, such as the number of casualties in the war. TF-IDF simply stated, "Many of the deadliest battles in history occurred during World War I," which somewhat indicates that the model accounts for sentiment and semantic analysis.

Conclusion

When analyzing the results, the SentenceTransfomer model provided the best responses compared to other models built in this study. This is also a model pre-trained on 215 million (question, answer) pairs, so the results can be expected to be more accurate than the TF-IDF and LSTM approaches. While the ontology graph created in this study was used as a reference point, the models did not necessarily reflect the graph. That being said, all the three models did manage to extract key terms to develop responses, which is similar to the approach taken to build the ontology graph. The difference, however, is that the ontology graph was built using context while the models failed to take context into account.

The preprocessing methods used in this study may have also played a role in the results. The use of tokenization, removing stopwords and punctuations, lowercasing all words, and lemmatization may have caused the responses to not be as coherent as they could be. While TF-IDF provided somewhat coherent responses, LSTM completely failed at stringing together words to create a response. That also indicates that the approach of predicting the next word of a sequence may not be the best approach. While both LSTM and SentenceTransformer models used embedding, SentenceTransformer likely excelled because of the pre-trained data. It may be that the corpus was not large enough to train the models appropriately.

Ultimately, all three chatbots attempted to use semantic analysis by trying to extract key words, but the models did not necessarily understand meaning the of the words, especially the

TF-IDF and LSTM models. This may because the models needed further finetuning or the amount of data used in the study was not enough.

Directions for future work

One of the major factors that results in poor responses may be due to the size of the dataset. In order to improve the training process, the corpus needs to be enhanced with additional topics that dive deeper into World War I topics. This also means the ontology graph would need to be reworked in order to adjust for additional clustering and more complex relationships between entities and phrases. In terms of modeling, each model needs various iterations to understand how the models behave when the parameters are finetuned or the layers are adjusted. It is possible that less layers may result in more cohesive responses that make sense. While finetuning may be the first logical step moving forward, the preprocessing may also need to be reduced in order to have responses that are more viable. For example, removing stopwords and punctuations may not be necessary for a chatbot that is trying to respond in natural language. Finally, the use of advanced NLP techniques like part-of-speech tagging and entity recognition can help to develop meaning for the models. TF-IDF and SentenceTransformer models attempt to use semantic analysis through cosine similarity, and LSTM attempts to use semantic analysis through preprocessing the input query. However, all three models can be improved upon in terms of not just semantic analysis but also sentiment analysis, which the models lack.

Appendices

Figure 1.0 – EDA before preprocessing

```
Number of lines:  7
Number of words:  24570
Number of characters:  155245
```

Figure 1.1 – EDA after preprocessing

```
Number of lines:  1050
Number of words:  14973
Number of characters:  111406
```

Figure 1.2 – Top 10 most frequent words

```
Top 10 most frequent words:  [('war', 317), ('german', 198), ('germany', 121), ('british', 1
10), ('army', 103), ('``', 100), ("''", 98), ('russian', 78), ('power', 77), ('battle', 75)]
```

Figure 1.3 – Lemmatized sentences (processed)

```
['world war first world war 28 july 1914 – 11 november 1918 often abbreviated wwi one deadli
est global conflict history', 'fought two coalition ally primarily france united kingdom rus
sia italy japan united state central power led germany austria–hungary ottoman empire', 'fig
hting occurred throughout europe middle east africa pacific part asia', 'estimated 9 million
soldier killed combat plus another 23 million wounded 5 million civilian died result militar
y action hunger disease', 'million died result genocide 1918 spanish flu pandemic exacerbate
d movement combatant war', 'first decade 20th century saw increasing diplomatic tension euro
pean great power', 'reached breaking point 28 june 1914 bosnian serb named gavrilo princip a
ssassinated archduke franz ferdinand heir austro–hungarian throne', 'austria–hungary held se
rbia responsible declared war 28 july', 'russia came serbia ’ defence 4 august defensive all
iance drawn germany france britain', 'german strategy 1914 first defeat france attack russi
a', 'however failed end 1914 western front consisted continuous line trench stretching engli
sh channel switzerland', 'eastern front fluid neither side could gain decisive advantage des
pite series costly offensive', 'attempt bypass stalemate caused fighting expand middle east
alp balkan overseas colony bringing bulgaria romania greece others war', 'united state enter
ed war side ally april 1917 bolshevik seized power russian october revolution made peace cen
tral power early 1918', 'freed eastern front germany launched offensive west march 1918 hopi
ng achieve decisive victory american troop arrived significant number', 'failure left german
imperial army exhausted demoralised ally took offensive august 1918 could stop advance', '29
september 3 november 1918 bulgaria ottoman empire austria–hungary agreed armistice ally leav
ing germany isolated', 'facing revolution home army verge mutiny kaiser wilhelm ii abdicated
9 november', 'armistice 11 november 1918 brought fighting close paris peace conference impos
ed various settlement defeated power best–known treaty versailles', 'dissolution russian ger
man austro–hungarian ottoman empire resulted creation new independent state among poland cze
choslovakia yugoslavia', 'failure manage instability resulted upheaval interwar period contr
ibuted outbreak world war ii september 1939', 'term world war first coined september 1914 ge
rman biologist philosopher ernst haeckel', "claimed `` doubt course character feared 'europe
an war ... become first world war full sense word '' indianapolis star 20 september 1914",
```

Figure 1.4 – TF-IDF vectorized matrix

```
  (0, 1874)	0.26721400014803853
  (0, 929)	0.21503559194366964
  (0, 1745)	0.29725397813011023
  (0, 1075)	0.29725397813011023
  (0, 2692)	0.21090498911776434
  (0, 4170)	0.29725397813011023
  (0, 203)	0.29725397813011023
  (0, 2685)	0.2247063048258963
  (0, 65)	0.16199896703634328
  (0, 2643)	0.19078266786134201
  (0, 6)	0.23370508909724108
  (0, 61)	0.16600229068922884
  (0, 2144)	0.20366511111516938
  (0, 115)	0.240944967099836
  (0, 1605)	0.1753249010793713
  (0, 4075)	0.20132546783025387
  (0, 4155)	0.34397904579978583
  (1, 1357)	0.17818911847291544
  (1, 2728)	0.1709955103831362
  (1, 1919)	0.17906564615099513
  (1, 488)	0.1688359015255373
  (1, 1738)	0.14258500262078225
  (1, 2248)	0.20167642355978843
  (1, 2919)	0.16231483802127722
  (1, 769)	0.18372923986356524
  :	:
  (1047, 1738)	0.2118956512504972
  (1048, 3203)	0.2814606973254835
  (1048, 2873)	0.2814606973254835
  (1048, 3187)	0.2814606973254835
  (1048, 2117)	0.2814606973254835
  (1048, 2921)	0.2814606973254835
  (1048, 2127)	0.2814606973254835
  (1048, 4163)	0.2814606973254835
  (1048, 1465)	0.2814606973254835
  (1048, 4146)	0.22318450443318102
  (1048, 1460)	0.26575075919207747
  (1048, 2697)	0.26575075919207747
  (1048, 1671)	0.23292180128406595
  (1048, 1809)	0.2546043806999931
  (1048, 976)	0.21910226122756676
  (1048, 4075)	0.08999441530967796
  (1049, 270)	0.4055435128640696
  (1049, 2434)	0.3829078001054561
  (1049, 3053)	0.3668475063154317
  (1049, 2898)	0.34421179355681825
  (1049, 2912)	0.34421179355681825
  (1049, 1698)	0.3829078001054561
  (1049, 976)	0.31569416809884193
  (1049, 2472)	0.2329813177460779
  (1049, 4075)	0.12966873055327427
```

Figure 1.5 – Ontology graph

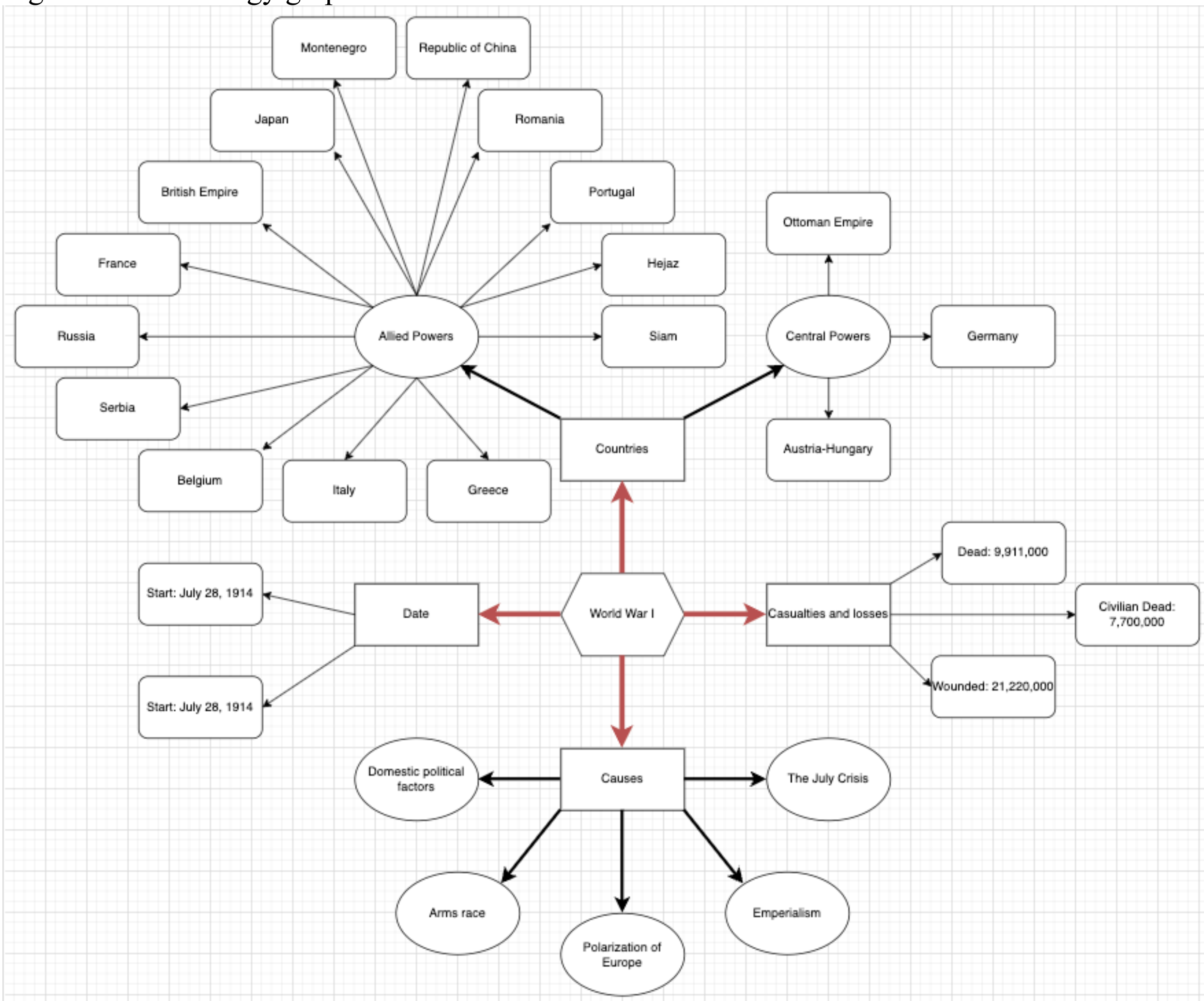

Figure 1.6 – Chatbot results (TF-IDF vs LSTM vs SentenceTransformer vs Chat GPT-3)

Questions	TF-IDF	LSTM	SentenceTransformer	ChatGPT-3
When did World War I begin?	when compared with world war ii the first world war is often thought to be `` a wrong war fought for the wrong reason ''	german force world war i battle	world war i or the first world war 28 july 1914 – 11 november 1918 often abbreviated a wwi wa one of the deadliest global conflict in history	World War I began on July 28, 1914.

What caused World War I?	after world war ii began in 1939 the term became more standard with british empire historian including canadian favouring `` the first world war " and american `` world war i "	for by world war i	world war i had a lasting impact on collective memory	There were several factors that contributed to the outbreak of World War I. Some of the main causes include militarism, alliances, imperialism, and the assassination of Archduke Franz Ferdinand of Austria-Hungary. The complex system of alliances between European countries also played a significant role in the start of the war, as it caused a chain reaction of mobilizations and declarations of war.
How many casualties were in World War I?	many of the deadliest battle in history occurred during world war i	that war casualty were in world war i	of the 60 million european military personnel who were mobilised from 1914 to 1918 8 million were killed 7 million were permanently disabled and 15 million were seriously injured	It is estimated that there were around 8.5 million military deaths and 13 million civilian deaths during World War I. The total number of casualties, including both deaths and injuries, is estimated to be around 37 million.

What countries fought in World War I?	in the aftermath of world war i greece fought against turkish nationalist led by mustafa kemal a war that eventually resulted in a massive population exchange between the two country under the treaty of lausanne	for 000 soldier in world war i	it wa fought between two coalition the ally primarily france the united kingdom russia italy japan and the united state and the central power led by germany austriahungary and the ottoman empire	The main countries that fought in World War I were the Allied Powers, which included France, Great Britain, and Russia, and the Central Powers, which included Germany, Austria-Hungary, and the Ottoman Empire. Many other countries also participated in the war on both sides, including Italy, Japan, and the United States.
Who fought for the Allies?	the optimism of la belle époque wa destroyed and those who had fought in the war were referred to a the lost generation	's soldier for the ally	it wa fought between two coalition the ally primarily france the united kingdom russia italy japan and the united state and the central power led by germany austriahungary and the ottoman empire	The Allies in World War I consisted of France, Russia, and Britain, and later included the United States and several other countries.

Figure 1.7 – TF-IDF model performance metrics

```
# define function to evaluate chatbot performance
def evaluate_chatbot(test_questions, test_answers):
    predicted_answers = []
    for question in test_questions:
        predicted_answer = generate_response(question)
        predicted_answers.append(predicted_answer)

    accuracy = accuracy_score(test_answers, predicted_answers)
    precision = precision_score(test_answers, predicted_answers, average='weighted')
    recall = recall_score(test_answers, predicted_answers, average='weighted')
    f1 = f1_score(test_answers, predicted_answers, average='weighted')

    print("Accuracy: {:.3f}".format(accuracy))
    print("Precision: {:.3f}".format(precision))
    print("Recall: {:.3f}".format(recall))
    print("F1 Score: {:.3f}".format(f1))

# evaluate chatbot performance on test data
test_questions = ["What was the main cause of World War 1?",
                  "Who were the Central Powers?",
                  "What was the Treaty of Versailles?"]
test_answers = ["The main cause of World War 1 was the assassination of Archduke Franz Ferdinand.",
                "The Central Powers were Germany, Austria-Hungary, and the Ottoman Empire.",
                "The Treaty of Versailles was the peace treaty that ended World War 1."]

evaluate_chatbot(test_questions, test_answers)
```

```
Accuracy: 0.000
Precision: 0.000
Recall: 0.000
F1 Score: 0.000
```

Figure 1.8 – LSTM model performance metrics

```
import numpy as np
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report

def evaluate_model(model, data, labels):
    # make predictions
    predicted_labels = np.argmax(model.predict(data), axis=-1).ravel()

    # convert labels to 1D array
    labels = labels.ravel()

    #report = classification_report(labels, predicted_labels)
    #print(report)

    # calculate metrics
    accuracy = np.mean(predicted_labels == labels)
    tp = np.sum((predicted_labels == 1) & (labels == 1))
    fp = np.sum((predicted_labels == 1) & (labels == 0))
    fn = np.sum((predicted_labels == 0) & (labels == 1))
    precision = tp / (tp + fp)
    recall = tp / (tp + fn)
    f1 = 2 * precision * recall / (precision + recall)

    return accuracy, precision, recall, f1

# evaluate model on training data
accuracy, precision, recall, f1 = evaluate_model(model, data, labels)
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
33/33 [==============================] - 2s 47ms/step
Accuracy: 0.8930659983291562
Precision: 1.0
Recall: 1.0
F1 score: 1.0
```

References

Soufyane, Ayanouz, Boudhir Anouar Abdelhakim, and Mohamed Ben Ahmed. 2021. "An Intelligent Chatbot Using NLP and TF-IDF Algorithm for Text Understanding Applied to the Medical Field." *Emerging Trends in ICT for Sustainable Development* (January): 3-10. https://10.1007/978-3-030-53440-0_1.

Brownlee, Jason. 2017. "A Gentle Introduction to Long Short-Term Memory Networks by Experts." Machine Learning Mastery, Last modified July 7, 2021. https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/.

Chotia, Rachana. 2022 "NLP Chatbots: Why Your Business Needs Them Today." *Verloop.io*, October 25, 2022. https://verloop.io/blog/nlp-chatbots/#:~:text=NLP%20chatbots%20give%20users%20more,their%20assessment%20in%20the%20future.

Dhankhar, Poonam. 2018. "RNN and LSTM based Chatbot using NLP." *International Journal of Innovations in Engineering and Technology* 10, no. 2 (May): 214-217. http://dx.doi.org/10.21172/ijiet.102.32.

Feldges, Claude. 2022. "Text Classification with TF-IDF, LSTM, BERT: a comparison of performance." *Medium*, April 2, 2022. https://medium.com/@claude.feldges/text-classification-with-tf-idf-lstm-bert-a-quantitative-comparison-b8409b556cb3.

Hugging Face. n.d. "multi-qa-MiniLM-L6-cos-v1." Accessed February 20, 2023. https://huggingface.co/sentence-transformers/multi-qa-MiniLM-L6-cos-v1.

Makadia, Mitul. 2019. "5 Reasons Why Your Chatbot Needs Natural Language Processing." *Towards Data Science*, May 1, 2019. https://towardsdatascience.com/5-reasons-why-your-chatbot-needs-natural-language-processing-ed20fb0a3655.

Masum, Abu Kaisar Mohammad, Sheikh Abujar, Sharmin Akter, Nushrat Jahan Ria, and Syed Akhter Hossain. 2021. "Transformer Based Bengali Chatbot Using General Knowledge Dataset." *IEEE International Conference on Machine Learning and Applications*, (2021). https://doi.org/10.48550/arXiv.2111.03937.

Maxime. 2019. "What is a Transformer?" *Inside Machine Learning*, January 4, 2019. https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04.

Merritt, Rick. 2022. "What is a Transformer Model?" March 25, 2022. Nvidia. March 25, 2022. https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/.

Simha, Anirudha. 2021. "Understanding TF-IDF for Machine Learning." Capital One. October 6, 2022. https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/.

Wampler, Matt. 2023. "The Technology Behind Chat GPT-3." *ClearCogs*, January 5, 2023. https://www.clearcogs.com/post/the-technology-behind-chat-gpt-3.