Varin Sikand (VSS180000)
Aditya Guin (ASG180005)
CS 4395.001
11/13/2022

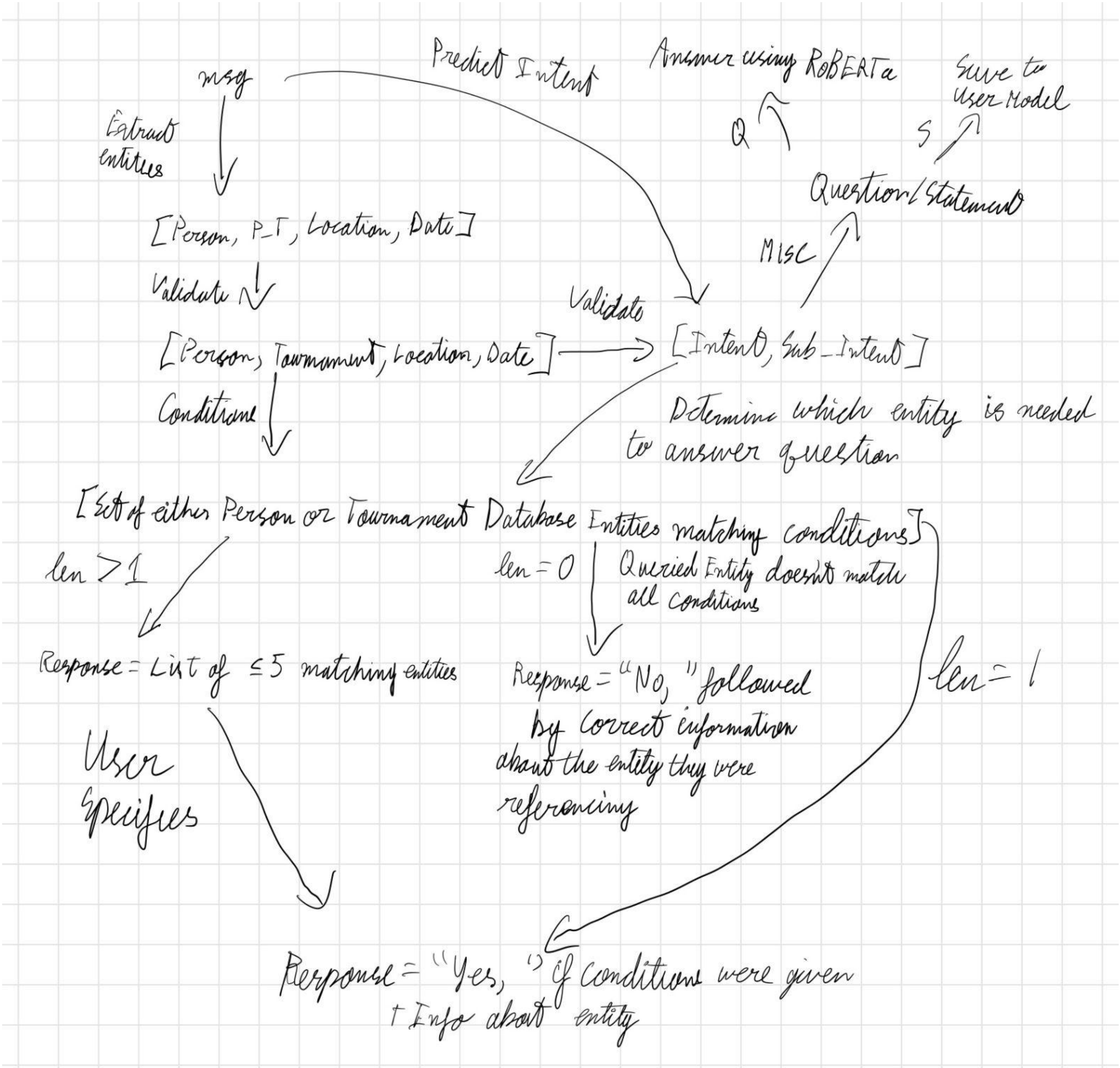
Chester 1.0 Report

System Description

Our chatbot is designed to answer questions about chess grandmasters and major tournaments. When the system starts, a chessbot class is initialized, and the user is prompted to enter a question regarding a well-known chess grandmaster or chess tournament. Then, from the input question, we use the DistilBERT for name entity recognition to extract the entities. Our model has 3 labels; P_Name, P_T_Name, and Location. P_Name represents a person's name, which is a chess grandmaster. P_T_Name represents a player tournament name; the question includes both a player and a tournament. Location indicates a location was given; this is usually a country. From there we make use of DeBERTA, to classify the intents of the question. In our chatbot, we had intents - Person, Tournament, and Misc- for and for Person we had sub-intents -Where_is, When_is, and Who_is- that specified what the answer should look like. Who_is gave general information about whatever player was queried, When_is only gave the location and Where_is only gave the date of birth. This is to enable the chatbot to have specific answers for specific queries. In order for querying to be done, we relied on the entities extracted by the NER which, after undergoing some verification by cross-checking with our knowledge base if the entities were stored within our database, allowed us to specify what conditions the user gave. Additionally, these entities can be used to validate and even overwrite the intent, since whenever a tournament name is found and validated to be within our database, we know that the user could not have been asking about a Person, and any person entity is rather defining a potential winner of the tournament. For example, differentiating between where a grandmaster is from, and when they won a specific tournament depends on if the user gave a query specifier like “from London” or “born in 1988”. The same applied for Tournaments, except instead of “born in” we denoted Location entities as indicating where the tournament was held. After identifying intent and entities, the response was governed by how many people/tournaments fit the description given by the user. If there were multiple, we had respective ‘Person_mult’, ‘Tourn_mult’, and ‘Tourn_year’ states that we would then enter in order to offer the user a chance to clarify which entity they would like to know about, allowing them to query by name or number after listing some of the options (if there were more than 5 matches, we returned a list of 5 random wons but inform them how many their query returned). Other states include a ‘Person_wrong’ state in case the user queried a person using an incorrectly spelled name. Multiple points of our code use autocorrection using Jaro-Winkler similarity scores in order to determine what entity the user was most likely trying to say, setting a discriminative threshold so that we can identify when the user said something entirely incoherent and needed to enter a new question. When processing entities, we implicitly perform this autocorrection in order to essentially create a mapping between the named entities in the message and the real entities stored within our database. In this way, we store context and allow the chatbot to remember a defined context_sentence in order to answer differently depending on what state it is in. The ‘skip’ state is used as a transition state and allows the user to escape the current state by just asking another question.

Varin Sikand (VSS180000)
Aditya Guin (ASG180005)
CS 4395.001
11/13/2022

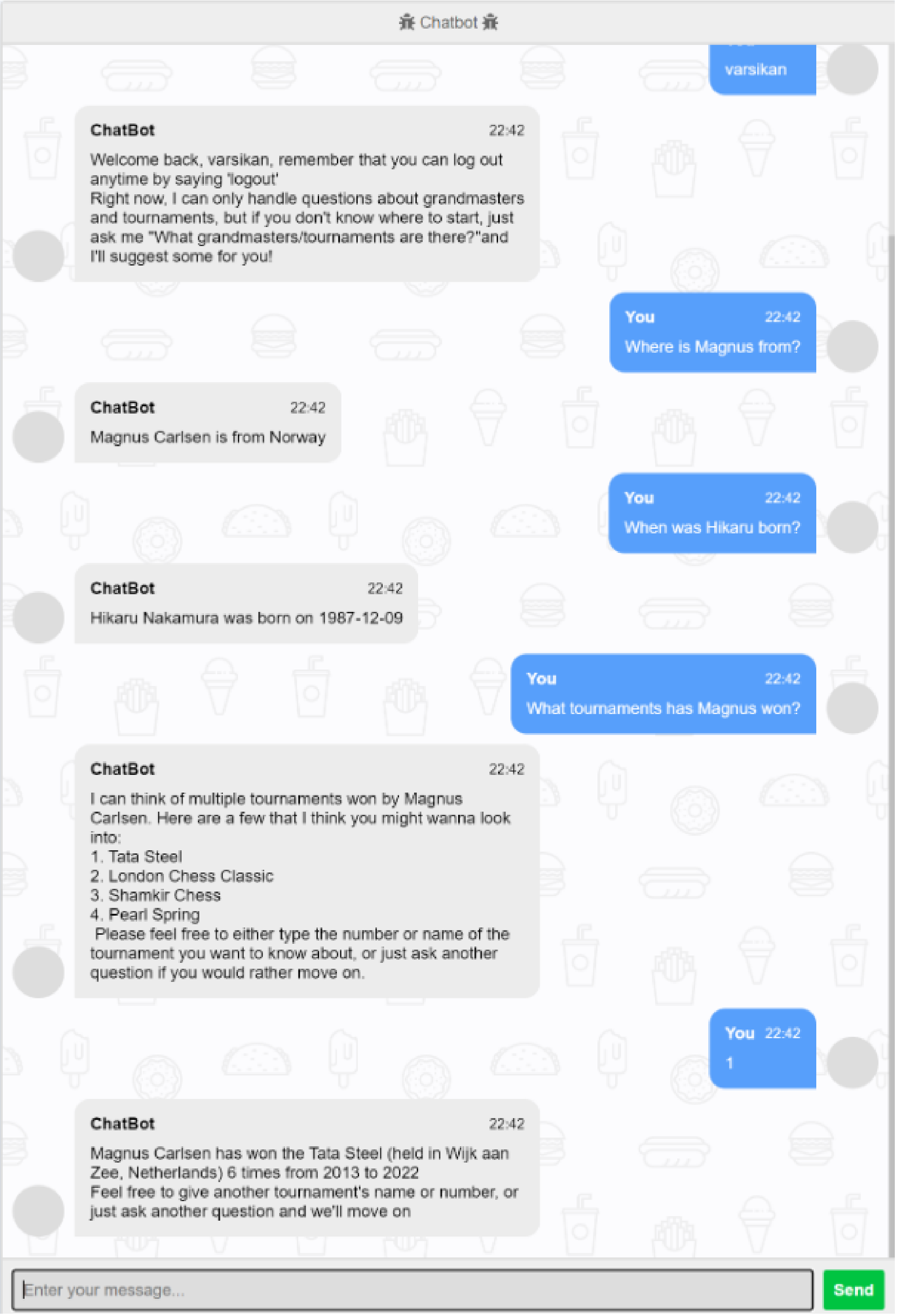
Diagram of Logic


msg
Extract entities
Predict Intent
Answer using RoBERTa
Q
Save to User Model
S
Question/Statement
MISC
[Person, P_T, Location, Date]
Validate
Validate
[Person, Tournament, Location, Date]
[Intent, Sub_Intent]
Conditions
Determine which entity is needed to answer question
[Set of either Person or Tournament Database Entities matching conditions]
len > 1
len = 0
Queried Entity doesn't match all conditions
len = 1
Response = List of ≤ 5 matching entities
Response = "No," followed by correct information about the entity they were referencing
User Specifies
Response = "Yes," if conditions were given + Info about entity

Varin Sikand (VSS180000)
Aditya Guin (ASG180005)
CS 4395.001
11/13/2022

Sample Dialog Interactions

Interaction between a bot and user about chess tournaments or players:



varsikan

ChatBot 22:42
Welcome back, varsikan, remember that you can log out anytime by saying 'logout'
Right now, I can only handle questions about grandmasters and tournaments, but if you don't know where to start, just ask me "What grandmasters/tournaments are there?"and I'll suggest some for you!

You 22:42
Where is Magnus from?

ChatBot 22:42
Magnus Carlsen is from Norway

You 22:42
When was Hikaru born?

ChatBot 22:42
Hikaru Nakamura was born on 1987-12-09

You 22:42
What tournaments has Magnus won?

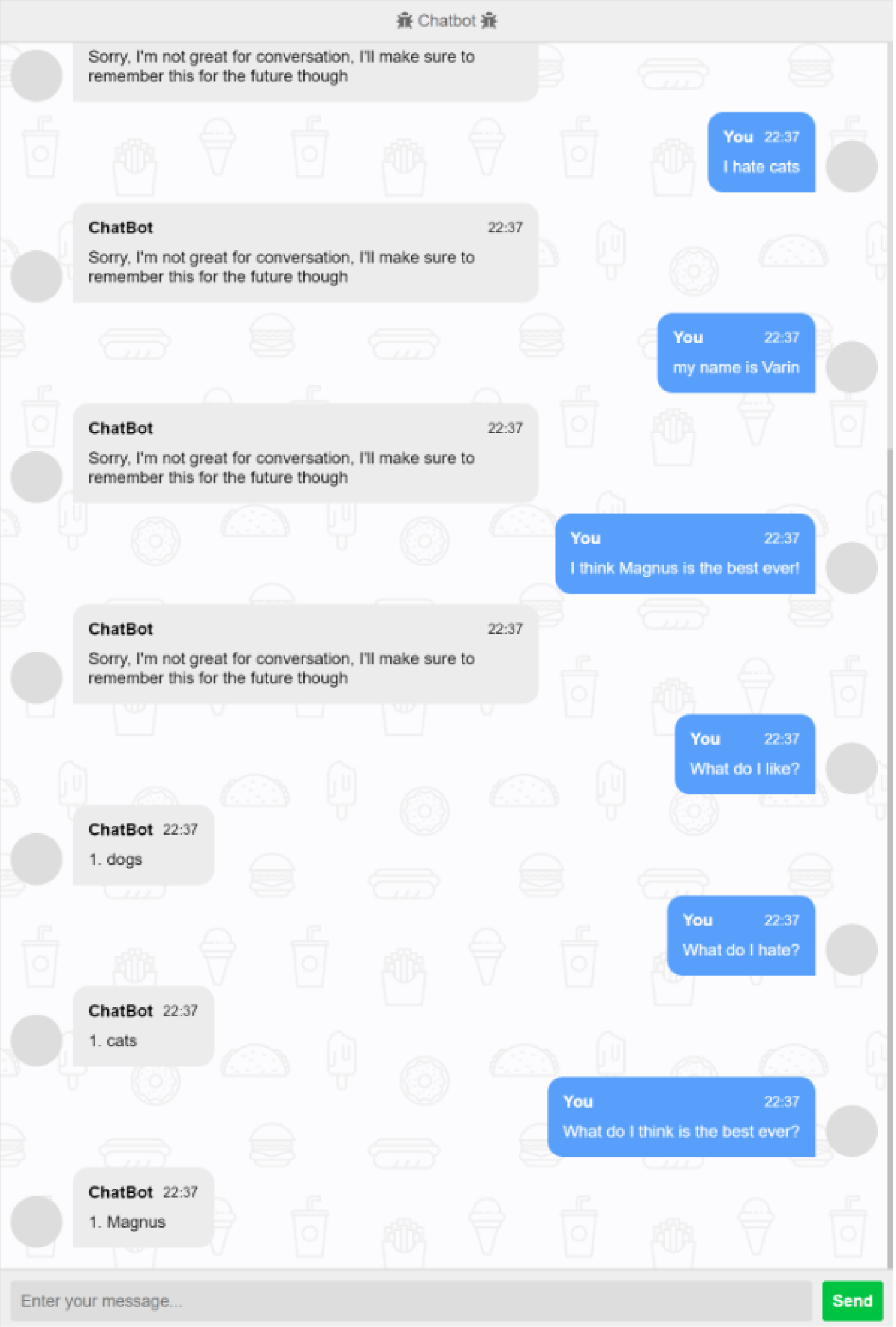
ChatBot 22:42
I can think of multiple tournaments won by Magnus Carlsen. Here are a few that I think you might wanna look into:
1. Tata Steel
2. London Chess Classic
3. Shamkir Chess
4. Pearl Spring
Please feel free to either type the number or name of the tournament you want to know about, or just ask another question if you would rather move on.

You 22:42
1

ChatBot 22:42
Magnus Carlsen has won the Tata Steel (held in Wijk aan Zee, Netherlands) 6 times from 2013 to 2022
Feel free to give another tournament's name or number, or just ask another question and we'll move on

Varin Sikand (VSS180000)
Aditya Guin (ASG180005)
CS 4395.001
11/13/2022

Interaction between the bot and user nonrelated to chess (maintaining the user model):



Sorry, I'm not great for conversation, I'll make sure to remember this for the future though

You 22:37
I hate cats

ChatBot 22:37
Sorry, I'm not great for conversation, I'll make sure to remember this for the future though

You 22:37
my name is Varin

ChatBot 22:37
Sorry, I'm not great for conversation, I'll make sure to remember this for the future though

You 22:37
I think Magnus is the best ever!

ChatBot 22:37
Sorry, I'm not great for conversation, I'll make sure to remember this for the future though

You 22:37
What do I like?

ChatBot 22:37
1. dogs

You 22:37
What do I hate?

ChatBot 22:37
1. cats

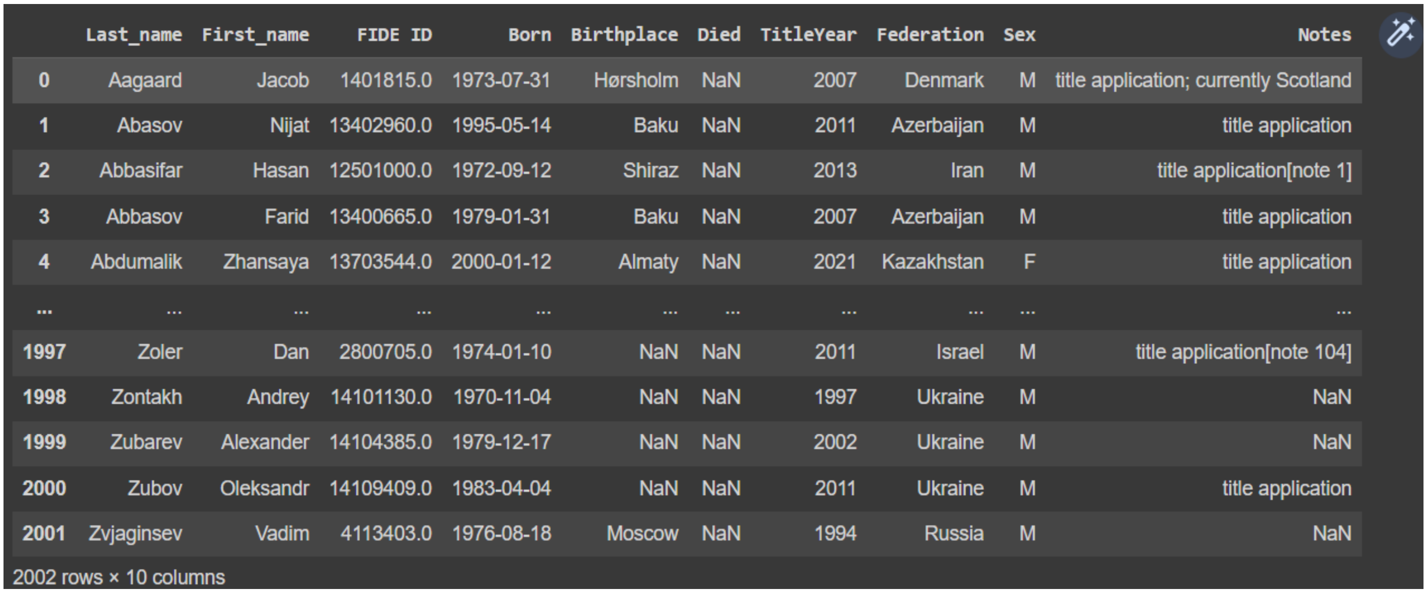
You 22:37
What do I think is the best ever?

ChatBot 22:37
1. Magnus

Varin Sikand (VSS180000)
Aditya Guin (ASG180005)
CS 4395.001
11/13/2022

Knowledge Base

We use SQLite within google collab to create our database. First, we create data frames with pandas for all our data, and then we utilize the to_sql function to create a SQL table for our schema. In our actual program itself, we query the individual tables from the schema. We have three tables; chess_grandmasters, tournament_winners, and tournament_locations. The chess_grandmaster table is read into a pandas data frame from a Wikipedia link that contains a list of every grandmaster in chess. The wikipedia table contains various attributes, including last name, first name, country of origin, and federation they currently represent. A sample of the data frame is shown below.

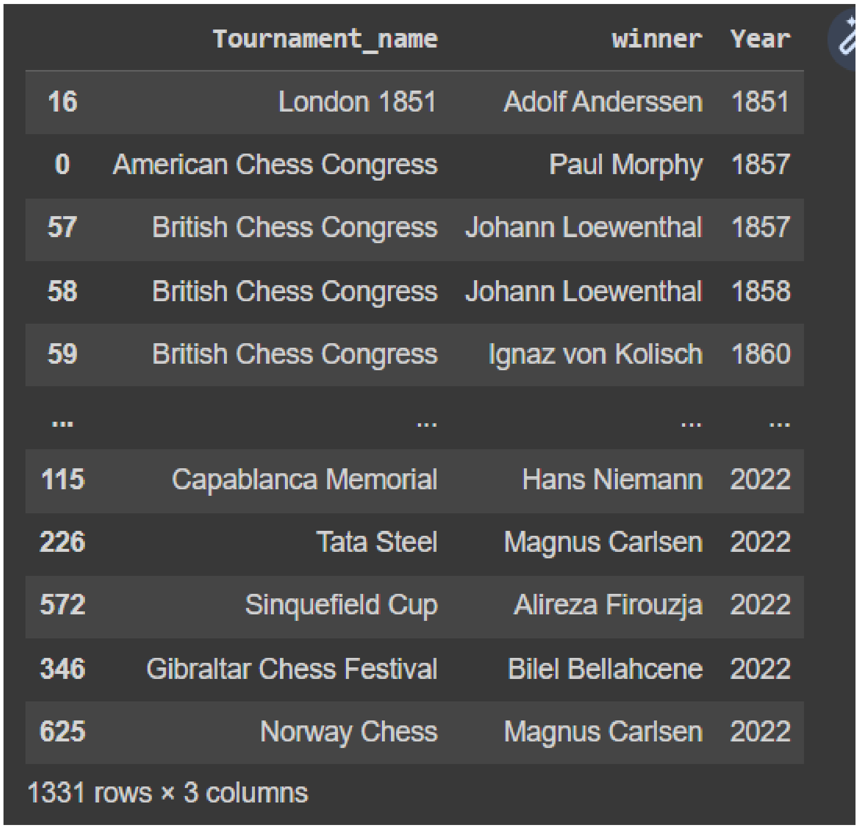


	Last_name	First_name	FIDE ID	Born	Birthplace	Died	TitleYear	Federation	Sex	Notes
0	Aagaard	Jacob	1401815.0	1973-07-31	Hørsholm	NaN	2007	Denmark	M	title application; currently Scotland
1	Abasov	Nijat	13402960.0	1995-05-14	Baku	NaN	2011	Azerbaijan	M	title application
2	Abbasifar	Hasan	12501000.0	1972-09-12	Shiraz	NaN	2013	Iran	M	title application[note 1]
3	Abbasov	Farid	13400665.0	1979-01-31	Baku	NaN	2007	Azerbaijan	M	title application
4	Abdumalik	Zhansaya	13703544.0	2000-01-12	Almaty	NaN	2021	Kazakhstan	F	title application
...	...	...	...	...	...	...	...	...	...	...
1997	Zoler	Dan	2800705.0	1974-01-10	NaN	NaN	2011	Israel	M	title application[note 104]
1998	Zontakh	Andrey	14101130.0	1970-11-04	NaN	NaN	1997	Ukraine	M	NaN
1999	Zubarev	Alexander	14104385.0	1979-12-17	NaN	NaN	2002	Ukraine	M	NaN
2000	Zubov	Oleksandr	14109409.0	1983-04-04	NaN	NaN	2011	Ukraine	M	title application
2001	Zvjaginsev	Vadim	4113403.0	1976-08-18	Moscow	NaN	1994	Russia	M	NaN

2002 rows × 10 columns

Chess Grandmaster table

Tournament winners contain the winners of well-known tournaments. Some tournaments have only occurred once; for example, the AVRO 1938 tournament was only held in 1938. Other tournaments, such as the London Chess Classic are recurring tournaments that happen annually. To account for recurring tournaments, we have three attributes in our table; Tournament Name, Winner, and Year. The year attribute in each row uniquely identifies a tournament; for example, London Chess Classic 2021 vs London Chess Classic 2016. A majority of the tournaments in the winners table are recurring, and a sample of the data frame can be seen below.



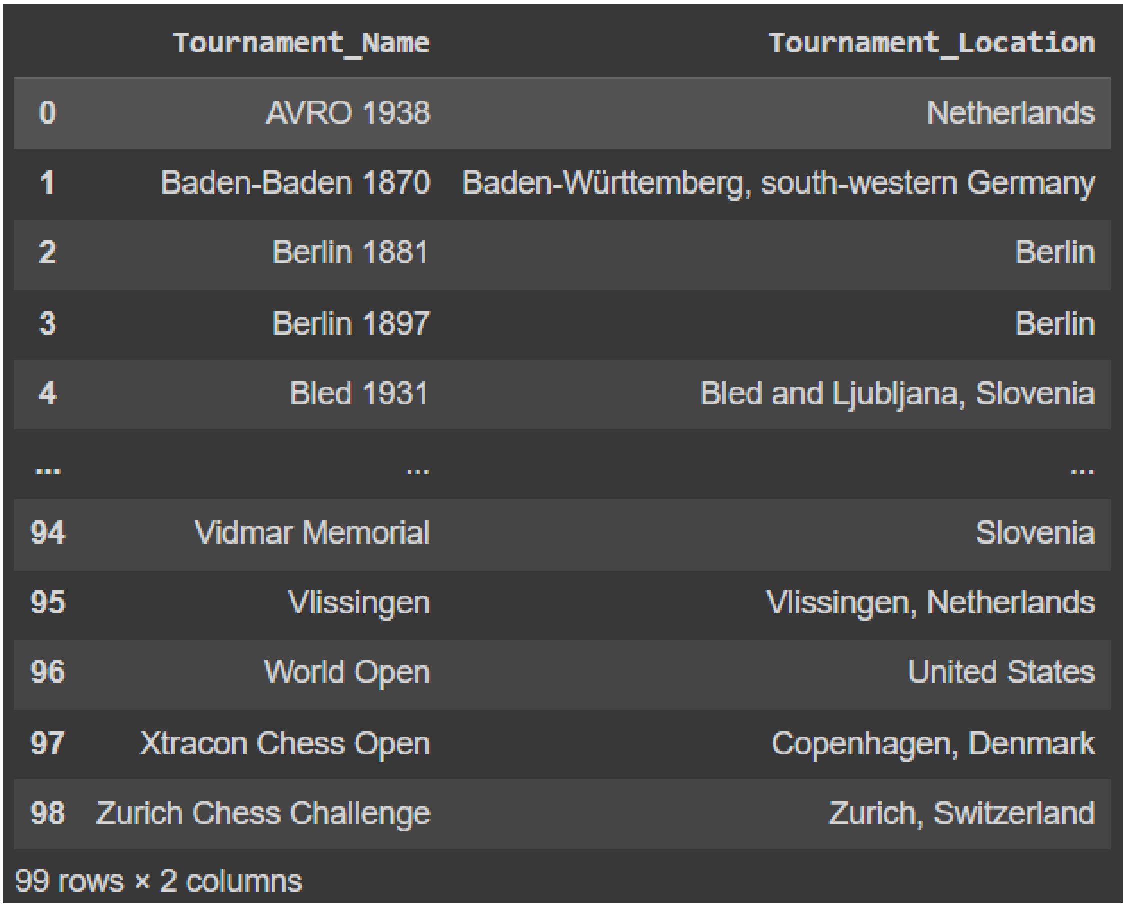
	Tournament_name	winner	Year
16	London 1851	Adolf Anderssen	1851
0	American Chess Congress	Paul Morphy	1857
57	British Chess Congress	Johann Loewenthal	1857
58	British Chess Congress	Johann Loewenthal	1858
59	British Chess Congress	Ignaz von Kolisch	1860
...	...	...	...
115	Capablanca Memorial	Hans Niemann	2022
226	Tata Steel	Magnus Carlsen	2022
572	Sinquefield Cup	Alireza Firouzja	2022
346	Gibraltar Chess Festival	Bilel Bellahcene	2022
625	Norway Chess	Magnus Carlsen	2022

1331 rows × 3 columns

Tournament Winner Table

Varin Sikand (VSS180000)
Aditya Guin (ASG180005)
CS 4395.001
11/13/2022

The tournament location table contains all the tournament names, along with the location where they are played. This table is used to query questions inquiring about where a tournament was held. It has the tournament name and the corresponding location.



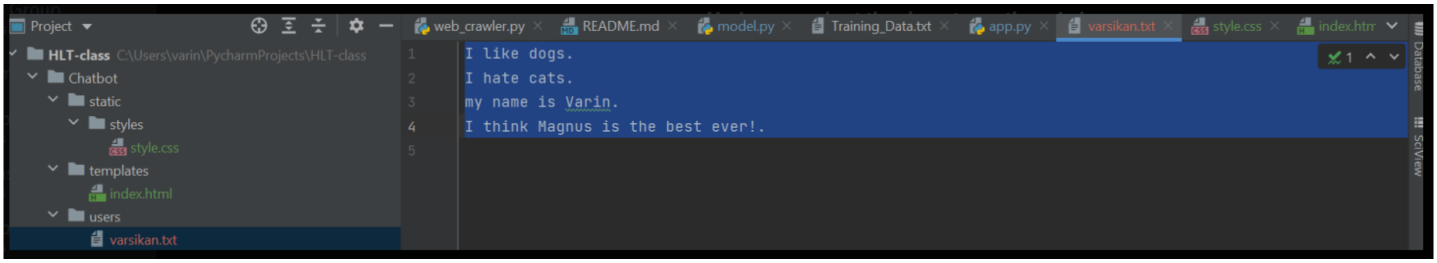
	Tournament_Name	Tournament_Location
0	AVRO 1938	Netherlands
1	Baden-Baden 1870	Baden-Württemberg, south-western Germany
2	Berlin 1881	Berlin
3	Berlin 1897	Berlin
4	Bled 1931	Bled and Ljubljana, Slovenia
...	...	...
94	Vidmar Memorial	Slovenia
95	Vlissingen	Vlissingen, Netherlands
96	World Open	United States
97	Xtracon Chess Open	Copenhagen, Denmark
98	Zurich Chess Challenge	Zurich, Switzerland

99 rows × 2 columns

The data was primarily accumulated from Wikipedia. If we couldn't find specific details for one tournament or a player, we also used other chess database websites such as lichess or chess.com.

User Model

When the program starts, the user is prompted to enter a username. Then a folder named "users" is created in which a text file for that user exists. From there, all statements and questions not directly related to chess grandmasters or tournaments are stored in order to remember information about the user. We use the RoBERTa-Large-Squad2 question-answering model to retrieve information about a user. For example, if the user named Bob enters "I like bread.", and later asks "What do I like?", we first store the statement into Bob.txt and use RoBERTa to retrieve the answer. The same idea applies to dislikes as well. A sample of the user models is shown below (an extension of the previous sample dialog interaction shown).



```
I like dogs.
I hate cats.
my name is Varin.
I think Magnus is the best ever!.
```

Varin Sikand (VSS180000)
Aditya Guin (ASG180005)
CS 4395.001
11/13/2022

Strengths and Weaknesses

Using the DeBERTa model to train for intents enabled the chatbot to properly identify what type of question the user was asking. From that, the bot is able to retrieve the correct answers to the question using the Jaro-Winkler similarity metric. For example, if the question asks about where a grandmaster is from, the sub-intent is properly classified, and the answer is returned. In addition, our chatbot maintains a user model and remembers likes and dislikes that the user mentions during the conversation. In addition, our model accounts for any potential typos a user could make while asking a question and returns information about the intended tournament or player. Our bot is also able to maintain the conversation, even if the question posed isn't known to it.

Some weaknesses of our Chester is its limited knowledge; it can only answer questions about chess grandmasters and tournaments. Moreover, there might exist some chess tournaments or players that aren't in our database. While the intents are accurately recognized most of the time, there are some instances where the classification of the statement is incorrect, which could lead to a response with incorrect information. Something we wished we would have explored more was fine tuning our question answering model in order to better answer questions regarding the user model rather than the basic implementation we have currently.

Varin Sikand (VSS180000)
Aditya Guin (ASG180005)
CS 4395.001
11/13/2022

Conclusion

In conclusion, our chatbot answers questions about chess grandmasters and tournaments. By using NLP techniques such as NER and QA, the bot is able to identify question intents and maintain user models in order to carry out a logical conversation with the user. We enjoyed creating this bot, and look forward to improving upon it in the future.