Smart Tarot Reading: an IR based Game

Rema Salman
School of Computer Science, Physics and
Mathematics, Linnaeus University, Sweden
rs223bc@student.lnu.se

Abstract. Fortune-telling is generally associated with scepticism. Tarot cards come in various designs and have different applications depending on the deck being used. Meanwhile, we were unable to find any games in the context of divination or psychic reading, especially tarot reading, that use IR as a foundation; therefore, we decided to challenge our skills and knowledge to attempt bridging such a gap. In this paper, we describe our process to design and build a Smart Tarot Reading system, using various information retrieval techniques. Often professional services play a key role interpreting the meaning of a certain outcome, thus, our prototype provides the player with the ability to do guided-reading which iteratively retrieves a suitable reading in the form of a subset of the Tarot cards based on his/her input query. The system uses vector space model association and for both the input and wach card, then, calculates the similarity cosine similarity approrach. The results showed that our prototype works as expected, while it renders the top three matching cards in the optimal use case scenario for a successful retrieval.

Keywords: Game, tarot, information retrieval, cosine similarity, text-similarity, machine learning, NLP

INTRODUCTION

Information retrieval (IR) represents the science of finding relevant materials from a large collection that satisfies the user's information need (Galli et at., 2014). According to Pathak and Lal (2017), today and with the massive increase in digital content, IR is being utilized for searching in documents, searching for documents, or searching for meta-data that are found in databases in forms of text, image, audio, HTML, XML, etc.

Fortune telling is usually associated with scepticism and perceived as unreal or not true, instead, it is used for entertainment, self-reflection or reframing personal perspective of the world. According to J. Walter Thompson Intelligence in relation to socially-related stress:

> *"We are increasingly turning to unreality as a form of escape and a way to search for other kinds of freedom, truth and meaning ... What emerges is an appreciation for magic and spirituality, the knowingly unreal, and the intangible aspects of our lives that defy big data" - Solowij (2016).*

Although there are several games that are powered by IR (Von Ahn et al., 2006; Ma et al., 2009; Law et al., 2009; Bennett et al., 2009), we were unable to identify any games in the context of divination or psychic reading, especially tarot reading. Therefore, we decided to complete an experimental study with an attempt to bridge such a gap by prototyping a web application that can intelligently behave for allowing the player to do a self-guided tarot reading. The proposed system was designed as a single-player game and driven by the player's input which forms a query that gets sent to the retrieval agent module in order to retrieve the top matching cards, then displays them to the player. Obviously, the proposed system does not promote any business ideas, rather the thought process behind proposing such a system is a simple game. This decision relates to the fact that fortune-telling itself is used for entertainment and as a sort of escapism from reality. In addition to this purpose, we chose to implement such a system to challenge our technological skills for developing an information retrieval prototype and, as a personal preference, to accumulate domain knowledge about the topics of Tarot, Information retrieval, and basic game design.

The rest of this paper is organized as follows: the related work section contains our search about the domain topics included in this project, being Tarot, game elements, information retrieval models, process and approaches. In the prototype design and development section, we cover our approach to design and build the prototype, covering the tools, libraries and other external resources that were used, after that, we also cover the system architecture and provide details about the prototype implementation. In the result section, we present the results from experimenting and running the system under different cases. Finally, we conclude the report with several highlights and propose some future investigations.

RELATED WORK

Tarot Cards Origins and background

Traditionally a physical card game the digital versions of the Tarot can be found in the Apple Store in the category “Future Insights & Numerology" or on Microsoft under “Entertainment” indicating the wide perception of the importance the cards play in society. As described by Parlett (2009), the Tarot decks were invented in Italy in the 1430s by adding to the existing four-suited pack a fifth suit of 21 specially illustrated cards called tionfi (“triumphs”) and an odd card called il matto (the fool). The symbolism found when reading the cards originates from a blending of Eastern and Western spiritual philosophies, Kabbalah and the Western Hermetic Tradition with Spiritual Alchemy.

“Tree of life” is a concept that plays an essential role, representing an imagery tree that keeps its roots deep in reality and spreads its branches high in the heavens so that it can connect man and God and the world in between. As visible in Fig 1, ten cards are spread between heaven and the earth. Each card represents a power that interacts with the world. The 22 lines interconnecting the powers symbolize the various paths that connect the Will of God and our actions.

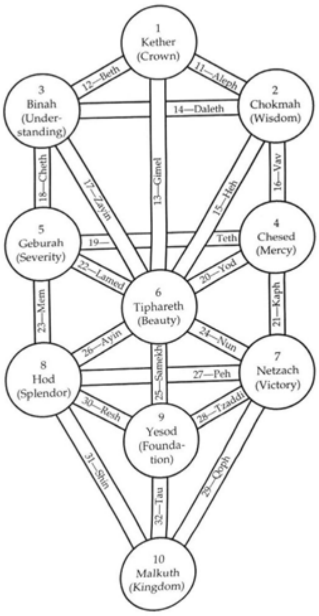


Fig 1. Diagram of the tree of life, sourced by Giles (2021).

The upper three cards represent the Will of Heaven. The cards four to nine represent ourselves in the Real World. The final four cards, cards seven to ten, symbolized the materialistic and animalistic side of ourselves. Furthermore, the leftmost vertical pillar, Boaz, represents a feminine power while the rightmost pillar, Jachin, represents a masculine power. Placed in the middle, the self achieves personal growth and development as it strives to maintain balance and harmony.

The three-card reading

The three-card reading is based on the three figures found on the Lovers card, as shown in Fig 2. The Lovers are ruled by Gemini, indicating that the card is not only a card of love but is a mandala of counties. The figures on this card have a triangle placement, an angel on the top, a woman and a man.

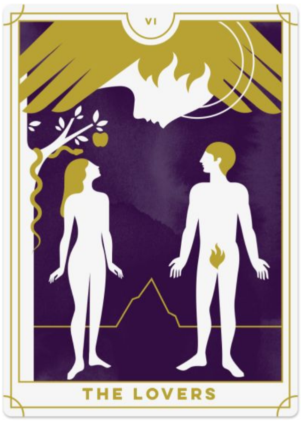


Fig 2. The Lovers card (source[1])

The woman represents the Subconscious mind. This is our feeling or emotional nature. The subconscious mind is everything that we have in our memory, or what we know intuitively. The man represents the Self-conscious mind or the everyday thinking mind. This is the mind that gathers information and interprets our experiences. The angel above these two represents the superconscious mind. This is the highest card in the reading. A placement order is paramount as it follows the position of the pattern of the top three spheres on the Tree of Life; Kether, Chokhmah and Binah. So when a person gets a three-card reading, the card represents a portrait of the individual.

Game Elements

Taking into consideration that researchers have presented their own extended versions of defining games and the components of games, Schell (2015) listed the common components that create games when combined together: *entered willfully, have goals, have conflict, have rules, be won or lost, are interactive, have challenges, create their own internal value, engage players, and are closed (formal) systems.*

However, considering the limitations of this project we focused only on the following components rather than the whole list mentioned by Schell:

1. ***Goals***: Allowing the player to do a self-guided tarot reading, through asking questions.
2. ***Rules:*** The actions which the player can take, thus, limiting the player's input to text rather than other media voice or video.
3. ***Engaging the player through rewards:*** under this point, the player gets rewarded by the feedback displayed.
4. ***Being an interactive game:*** The interactivity of our proposed prototype is embedded in the user-driven input.

Related IR Games

Galli et al. (2014) has extensively reviewed different games that use IR models and mapped them according to their game mechanics. In the paper (Galli et at., 2014), the authors covered several IR matching mechanics such as content (meta-data) indexing and query handling, respectively being, Peekaboom (Von Ahn et al., 2006), PageHunt (Ma et al., 2009), Search War (Law et al., 2009) and Picture This (Bennett et al., 2009) are examples of the games covered in Galli's et at. (2014) review. These games do not only differ in terms of the used IR mechanics but also different elements of their game design such as the rules, player interaction, feedback, and resources. Applying IR in game design has exceeded academic research, where even major search engines are attempting to make their users' search process more engaging by introducing entertainment elements to the information retrieval. Exemplifying this with the puzzle game provided by Google called "*A Google a Day*"[2] where players answer questions through performing intelligent Google searches. Another example is the *"Wikirace"*[3] which represents a competitive race between players where they travel through different Wikipedia pages via links for reaching the destination in the fastest way possible or with the least amount of links.

Information Retrieval (IR)

According to Singhal (2001), searching for information was firstly mentioned by Vannevar Bush in 1945. Years later, operational information retrieval systems were introduced, whereas around 1990 different techniques were built to process less than a thousand of documents. Nowadays, the massive growth of digital content (data) has urged large firms and academics to create scalable retrieval systems to cope with such growth. The general IR

[1] The Lovers Tarot Card Meanings
[2] http://googleaday.com
[3] http://thewikigame.com/

architecture is depicted in Fig 3, where the user input gets executed to the retrieval system. Then, the latter consults a document collection (stored in a database) and returns the matching documents.

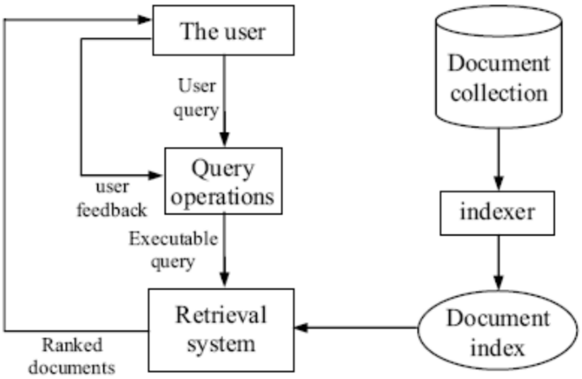


Fig 3. General IR System Architecture, Source by Bassil (2012).

Hence to allow the IR systems to understand the information of the documents and the input query they get transformed into models, which are covered in the following subsections.

A.Vector Space Model (VSM)

The Vector Space Model (VSM) is a data model that represents the documents and queries in an IR system, whilst Features are the dimensions of such vectors and these vectors represent the words accrued within the documents (Salton et al., 1975). As explained by Bassil (2012), each vector set presents an individual document but since IR systems handle a collection of documents, these vectors are compressed together to produce a matrix. Meanwhile, these vectors are normalized for checking the higher value features, meaning all vectors are converted into a standard length.

B. Term Frequency and Weighting (TF-IDF Approach)

Generally speaking, the terms that occur more frequently are assigned with higher scores in comparison with less occurring words in a given document. As mentioned by Luhn (1957), Term Frequency (TF) means the raw frequency of a word in a document. Depending only on the TF when assigning the relevance of documents to the query comes with the disadvantage that all terms have equal importance. Therefore, an Invest Document Frequency (IDF) approach was introduced to lower the weights of the repeatedly occurring words for computing a more relevant determination.

In the introduction to information retrieval book (Schütze et al., 2008), Tokenization is mentioned to be a critical technique for effectively extracting and calculating the TF. Tokenization is the process of splitting a document into smaller chunks called tokens, whereas tokenization can be done on different levels, e.g., sentences or words. Meanwhile, more precision of the IR system can be achieved by applying several preprocessing techniques to the documents. For example, any symbols, punctuations, digits, hyphens are removed. After that, stop words like "and", "I", "am" etc. are also cleaned. Another step is capitalization and case-folding where all uppercase characters are transformed into lowercase instead. Stemming and Lemmatization help to reduce the words into their basic forms by removing the word's morphology, i.e., establish + (ment/ism) or treat + (s/ing). By applying these preprocessing techniques is a form of word normalization since they get reduced to a standard form (Bassil, 2012).

C. Cosine-Similarity Measure

According to Pathak and Lal (2017), there are several methods for calculating the similarity between the input query and documents. Meanwhile, the most popular choice is Cosine similarity for measuring text similarity which is used for web and local-directory document similarity. Using the following formula allows calculating the similarity.

$$\text{Cosine} - \text{ similarity } (Q, D) = \frac{Q.D}{||Q||*||D||}$$

Q and D represent the query and document vectors, where ||Q|| and ||D|| are their length respectively. Considering the angle resulting from the previous formula, Fig 4 helps to interpret the cosine similarity in terms of the angle of separation between two vectors. Therefore, any system that uses such a similarity measure needs to execute some predefined function for the scenarios ‘Very similar’, ‘Not related’ and ‘Very Dissimilar’.

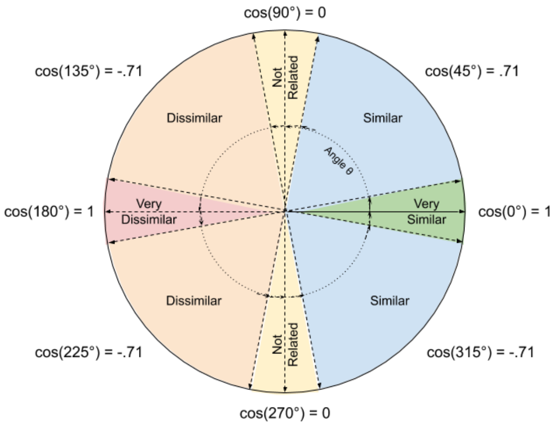
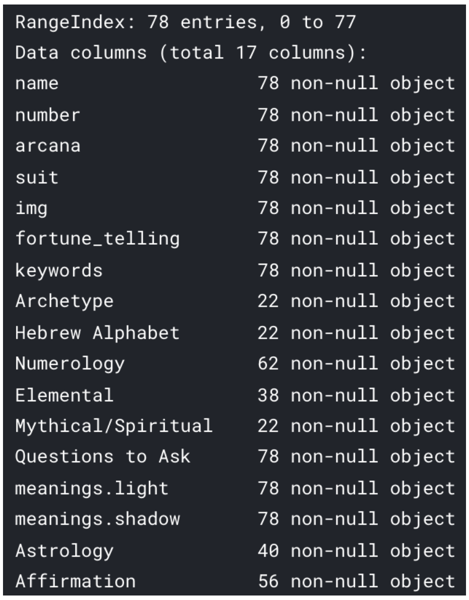


Fig 4. Interpretation of the cosine similarity results between two vectors. Source[4]

PROTOTYPE DESIGN & DEVELOPMENT

System Overview

As explained in the introduction, our proposed system utilizes concepts and technologies for creating a smart system to enable self-guided tarot reading in the form of a single-player game. Under the hood, the game uses an IR model to retrieve the top matching cards to the user’s query input. In order to complete the information retrieval process, we were able to find an open-sourced dataset of Tarot cards on Kaggle[5] that comes in the form of a Json file with a folder containing the cards’ images. The dataset includes 78 entries as each represents a tarot card, where each entry is also described by 17 different objects as listed below (see Appendix A for an example of a card).



```
RangeIndex: 78 entries, 0 to 77
Data columns (total 17 columns):
name                     78 non-null object
number                   78 non-null object
arcana                   78 non-null object
suit                     78 non-null object
img                      78 non-null object
fortune_telling          78 non-null object
keywords                 78 non-null object
Archetype                22 non-null object
Hebrew Alphabet          22 non-null object
Numerology               62 non-null object
Elemental                38 non-null object
Mythical/Spiritual       22 non-null object
Questions to Ask         78 non-null object
meanings.light           78 non-null object
meanings.shadow          78 non-null object
Astrology                40 non-null object
Affirmation              56 non-null object
```

[4] https://www.ml-science.com/cosine-similarity

[5] https://www.kaggle.com/lsind18/overview-of-tarot-deck/data

Designing the prototype while keeping in mind the general IR system architecture, we decided upon keeping the user input as the main functionality of the system. As visualized in Fig 5, we designed the game according to presented scenarios in the flow chart, where the user's input is always available and displayed to the player for entering a query.

The first scenario correlates to the first check of the user input, thus, if the query entered is being empty the player gets directly an error message displayed with the chance of re-submitting a new query.

The second scenario aligns with the second check of the user input, meaning after preprocessing the query, so if it is being empty five random words would be displayed to the player and also the chance of re-submitting a new query.

The optimal use case for a successful retrieval is as follows: A user inputs a text or a question, which is not an empty string nor does not get empty after pre-processing. Then, the system analyzes the input and finds the top matching cards through transforming the card documents and query into vectors and calculating cosine similarity between the query and descriptions. Finally, according to the similarity results, the system will display the top three matching cards to the player. Otherwise, if there are no similarities found the system displays the 5 random words to the player with the chance of re-submitting a new query.

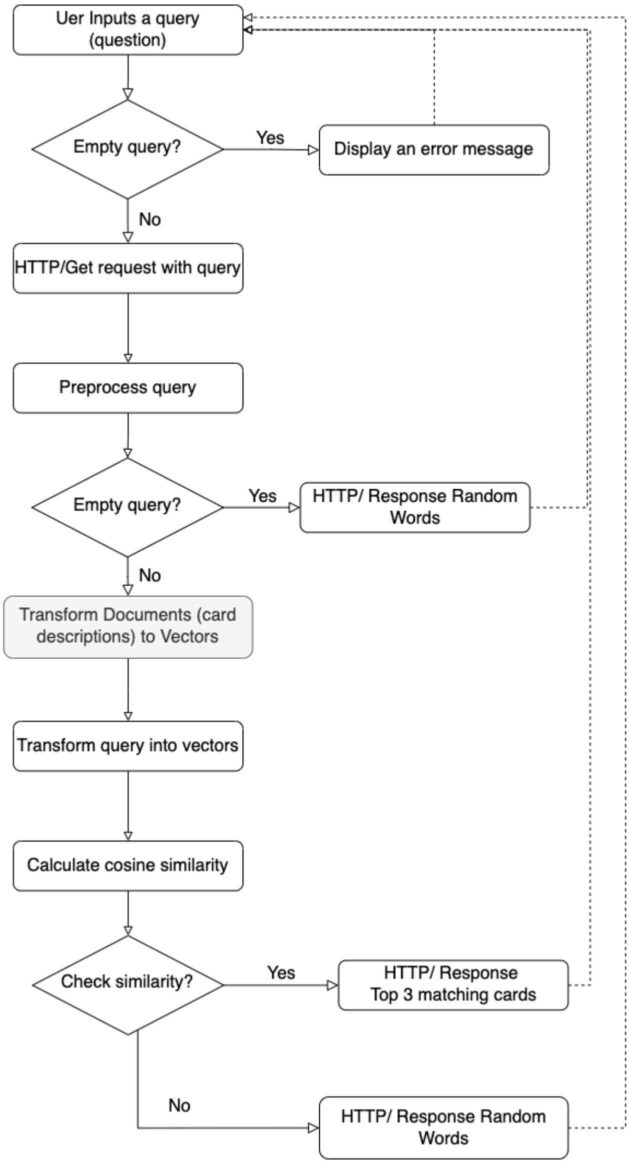

Uer Inputs a query (question)
Empty query?
Yes
Display an error message
No
HTTP/Get request with query
Preprocess query
Empty query?
Yes
HTTP/ Response Random Words
No
Transform Documents (card descriptions) to Vectors
Transform query into vectors
Calculate cosine similarity
Check similarity?
Yes
HTTP/ Response Top 3 matching cards
No
HTTP/ Response Random Words

Fig 5. Flow chart of the different use case scenarios.

Architecture of Proposed prototype

Fig 6 shows the system architecture in a high-level of abstraction, which follows mainly the IR processes, models, and approaches described in the related work.

Document (Card Soup)

A tarot card, as referred to in Appendix A, is described using 14 elements. Thus, we created a collection of unstructured data for each card by combining several of these entities together and refereed to them as a card soup. The entities are: fortune telling, keywords, Questions to Ask, and both light and shadow meanings. The motivation behind doing this comes from the general idea behind IR that works on unstructured data rather than structured, as well as, the convenience of using different entities for implementing the retrieval algorithm rather than depending on one entity such as keywords or meanings.

Query Input

The user interacts with the application through the browser rendered in the monitor. The web-client sends any query entered by the user to the intelligent part of the application to be handled.

Text Pre-processing

Both the input-query and card soup get pre-processing for providing better precision as advised in the book (Schütze et al., 2008). The preprocessing function manipulates the text in several steps: Removal of digits and punctuations. Then, the text was splitted to be lemmatized and stemmed as well as English stop words were removed. After that, the tokens were joined and the spaces were double-checked then the text was returned in form of a joined string, since the open-source library (used for VSM conversion) takes the text as an input and it automatically tokanizes it.

VSM

To retrieve a set of three tarot cards that match the input from the player, we needed a way to convert both the user input and cards' soups into vectors for estimating the similarity between the user input and tarot cards. Applying the VSM was done using an open source TF-IDF vectorization[6] which takes raw texts as input and outputs a matrix of TF-IDF vectors (as explained in the next section). Hence, both user input and the cards were converted into vectors. Noting that the user input was transformed according to the cards which resulted in the same dimensions.

Cosine Similarity

The vectors, resulting from the previous step, are used to calculate the similarity between the user input and each of the tarot cards. An open-source pairwise metrics library[7] was used for commuting the cosine similarity between the vectors. Then, the scores were sorted in a descending order to return the top three matching cards to the web-client to be displayed to the player.

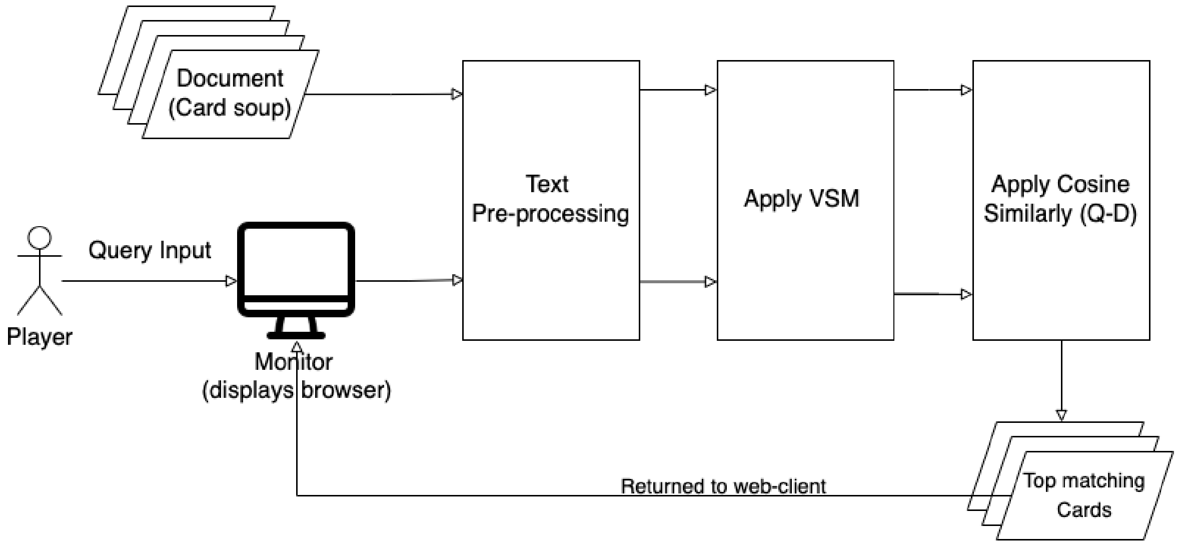


Fig 6. Architecture of the proposed prototype.

[6] sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 1.0.2 documentation

[7] sklearn.metrics.pairwise.cosine_similarity

Implementation details

The proposed system is a web application which consists of two parts: 1) a web client which gets rendered on the browser for allowing the user to interact with the application and displays the results and 2) an information retrieval module that represents the web server application and the intelligent part of the system, as displayed in Fig 7. The client was implemented using React (a JavaScript framework) and the server was implemented using Python. The communication between these two parts is through HTTP protocol, especifically, one route was created for requesting the matching cards to the query (via an HTTP/Get request), where the query is parsed and sent as the request body from the client using Axios[8]. The web-client uses the images folder that was downloaded with the data for displaying the cards' images into the browser, while the styling framework Bootstrap[9] was used to provide an acceptable interface design.

In the case of the Server, it used mainly Flask to be initialized as a restful API. Mainly, the program uses the Json file with the dataset of the Tarot-cards which was saved in the same root directory. In order to achieve the workflow of the IR system as described in the related work by Schütze et al. (2008), the Natural Language Toolkit (NLTK[10]) was utilized for the preprocessing and the library, Scikit-learn[11], was used for completing the intelligent part of the application, being the VSM and cosine similarity. As presented in Fig 7, the exact external resources or functions from the libraries used in the implementation, where the source code of the project is open-sourced and available on GitHub via this link.

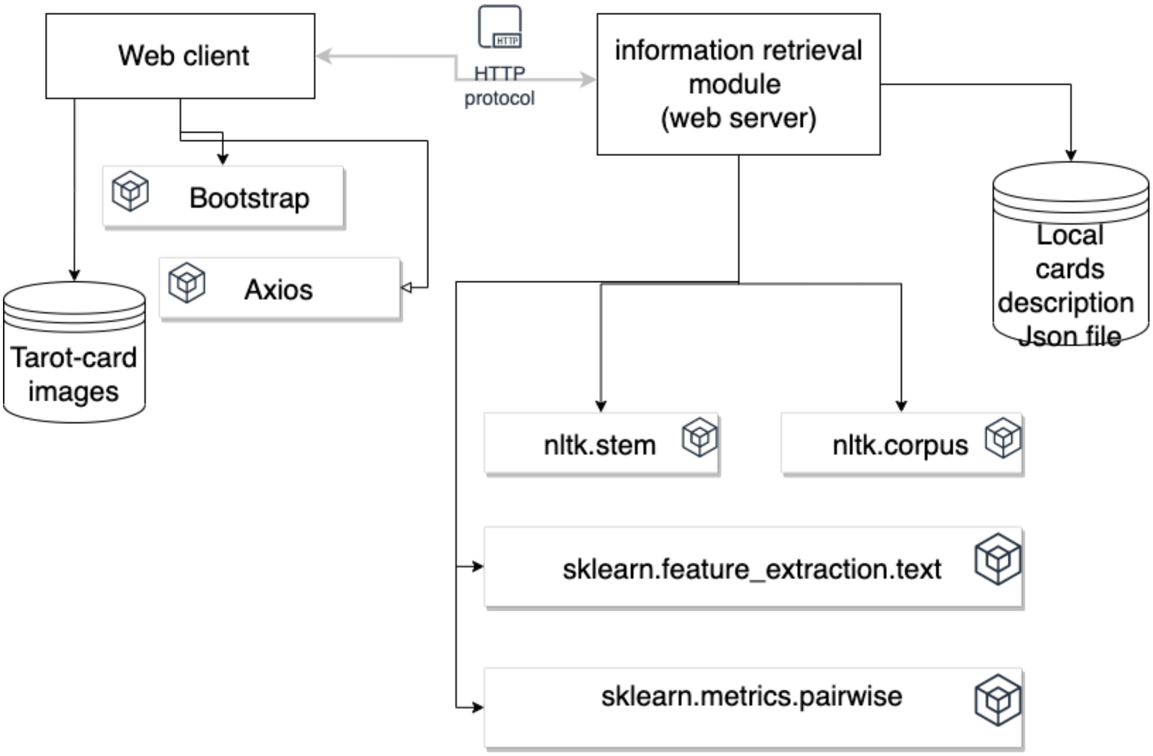
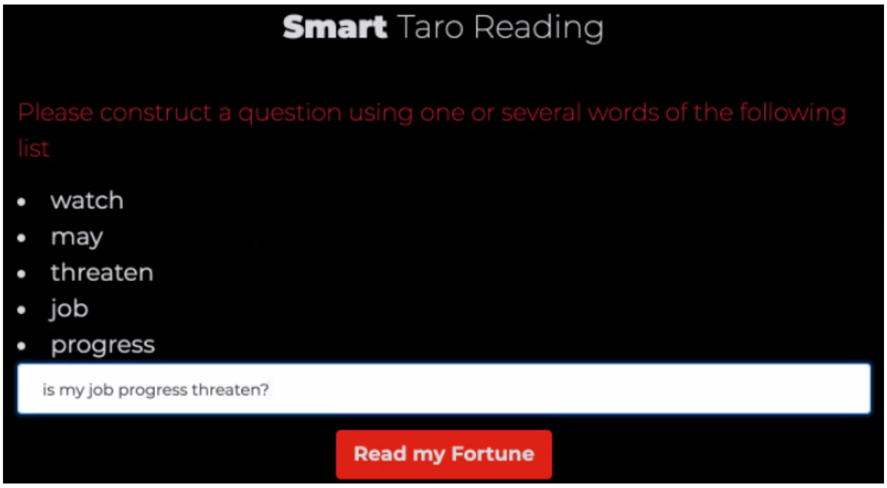

Web client
HTTP
HTTP protocol
information retrieval module (web server)
Bootstrap
Axios
Tarot-card images
Local cards description Json file
nltk.stem
nltk.corpus
sklearn.feature_extraction.text
sklearn.metrics.pairwise

Fig 7. Architecture of the proposed prototype, covering the implementation details.

[8] GitHub - axios/axios: Promise based HTTP client for the browser and node.js
[9] Introduction · Bootstrap v5.0
[10] NLTK
[11] Scikit-learn

RESULTS

To complete our experiment we run our prototype using different scenarios as described earlier in the system overview. In the case of user input: “1534 ME”, since this query input contains digits and a stop word, it gets empty after preprocessing (text cleaning). Therefore, the feedback is as expected a message with random words “*please construct a question using one or several words of the following list*”, as displayed in Fig 8.



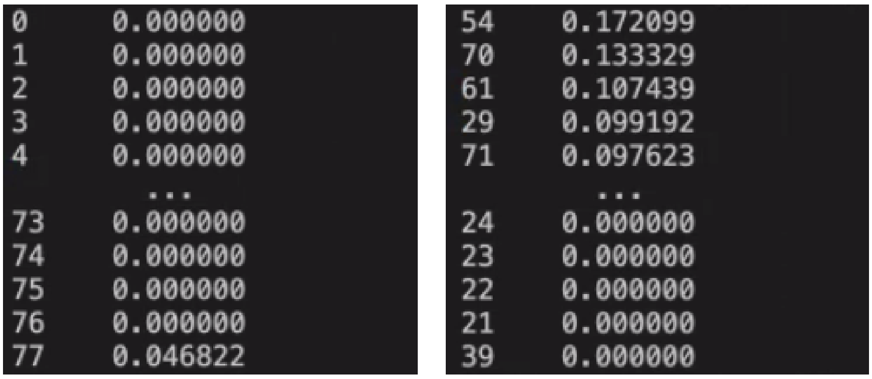
Please construct a question using one or several words of the following list

- watch
- may
- threaten
- job
- progress

is my job progress threaten?

Fig 8. Feedback in case of an empty string after preprocessing the player’s query.

As visualized in Fig 8, with a new attempt where the player constructs a question based on suggested words, being “Is my job progress threaten?”. The retrieval agent module (server) performs the indexing of the cards using the weighted indexing technique (TF-IDF), transforming the vectors into a multi-dimensional space to calculate the similarity using the cosine similarity measure. Fig 9 (A) displays the similarity score between the query and each document (card description), while Fig 9 (B) depicts the same information but sorted in descending order, from the highest to the lowest scores.



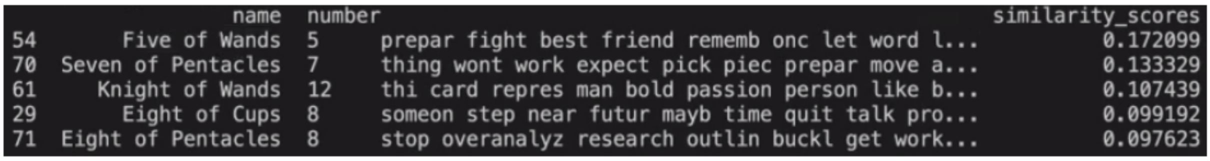
```
0      0.000000
1      0.000000
2      0.000000
3      0.000000
4      0.000000
         ...
73     0.000000
74     0.000000
75     0.000000
76     0.000000
77     0.046822
```

```
54     0.172099
70     0.133329
61     0.107439
29     0.099192
71     0.097623
         ...
24     0.000000
23     0.000000
22     0.000000
21     0.000000
39     0.000000
```

(A) Without Sorting *(B) After Sorting.*

Fig 9 . Cosine Similarity Scores between the user’s query and the tarot cards.

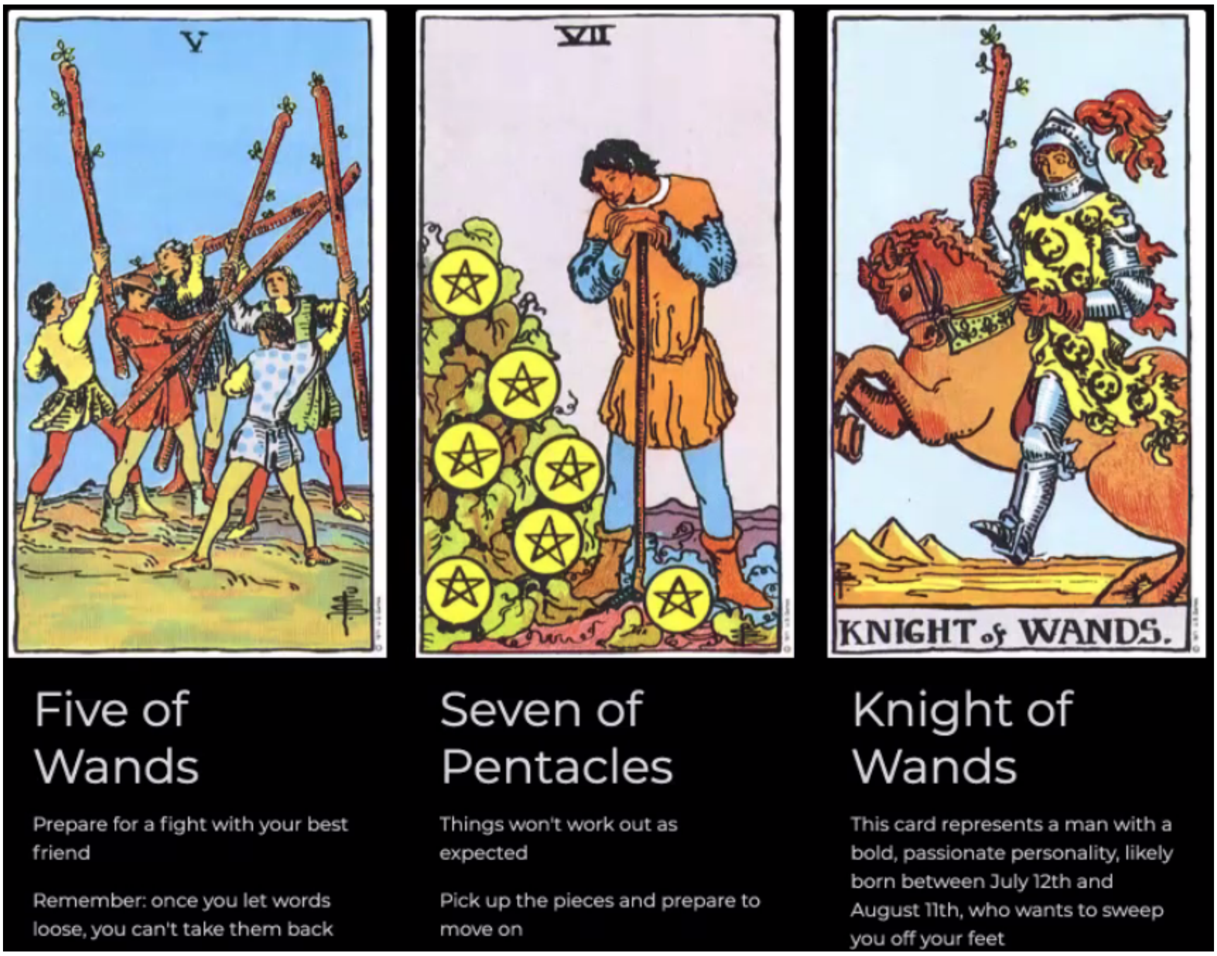
To verify the result, Fig 10 was screenshotted from the debugger. The cards with the highest similarity score for our example query is the ‘Five of Wand’, followed by the ‘Seven of Pentacles’ and ‘Knight of Wands’.



```
                 name  number                                                 similarity_scores
54      Five of Wands  5      prepar fight best friend rememb onc let word l...          0.172099
70  Seven of Pentacles  7      thing wont work expect pick piec prepar move a...          0.133329
61     Knight of Wands  12     thi card repres man bold passion person like b...          0.107439
29       Eight of Cups  8      someon step near futur mayb time quit talk pro...          0.099192
71  Eight of Pentacles  8      stop overanalyz research outlin buckl get work...          0.097623
```

Fig 10. Screenshot of the data frame after sorting the cards according to the similarity scores.

Fig 11 shows the same results in Fig 10 but they are displayed to the player in a more visual manner to enable them to do the readings of the three top matching cards.



Five of Wands

Prepare for a fight with your best friend

Remember: once you let words loose, you can't take them back

Seven of Pentacles

Things won't work out as expected

Pick up the pieces and prepare to move on

Knight of Wands

This card represents a man with a bold, passionate personality, likely born between July 12th and August 11th, who wants to sweep you off your feet

Fig 11. Reading displayed for the player on the browser.

CONCLUSION

By completing this project, we can say that we attempted to bridge the gap that we mentioned about not finding any games in the context of divination or psychic reading, especially tarot reading. Our experimentation to prototype a web application that can intelligently behave for allowing the player to do a self-guided tarot reading. Although the project had its limited time, which constrained us to focus only on the parts of the game components rather than everything listed by Schell (2015). Meanwhile, classifying our project as a game, we felt it was necessary to use the concept of the Three card spread in our game design, which even though the end result might appear similar, the methods are quite distinct. In real-life, the tarot game is based on more complex rules with an element of randomness, while in the case of this project, we presented the game that is based on algorithmic predictions using text similarity models to retrieve a matching set of cards to the user input query.

Leaving the assignment in an open-topic manner allowed us to think creatively how we can utilize the theoretical knowledge from IR into a practical web-application that can be used for entertainment purposes. However, we were able to accumulate the technical skills required to successfully finish the prototype of this project while gaining domain knowledge about the topics of Tarot, Information retrieval, and basic game design. Furthermore, our proposed system was designed as a single-player game and driven by the player's input which forms a query that gets sent to the retrieval agent module in order to retrieve the top matching cards, then displays them to the player.

Future Work

Taking into account that the proposed system in this project does not handle the semantics of the text during the information retrieval operations, we propose a further investigation that consists of designing and developing an ontology for the Tarot cards and implementing the retrieval process accordingly. In such cases, the ontology can also be expanded to cover different languages besides English to enable a larger target audience to interact with the web application.

Having in mind the limited time of this project, the game elements were not extensively designed; rather we focused on the design and implementation of the intelligent behaviour of the system. Thus, another point can be raised as future work is adding other game elements, such as multiple players, forming some rules for swapping cards to introduce a competitive behaviour between the players and, accordingly, different winning conditions can be constructed.

REFERENCES

Bassil, Y. (2012). A survey on information retrieval, text categorization, and web crawling. arXiv preprint arXiv:1212.2065.

Bennett, P. N., Chickering, D. M., & Mityagin, A. (2009). Learning consensus opinion: mining data from a labeling game. *In Proceedings of the 18th international conference on world wide web* (pp. 121-130).

Galli, L., Fraternali, P., & Bozzon, A. (2014). On the application of game mechanics in information retrieval. In *Proceedings of the First International Workshop on Gamification for Information Retrieval* (pp. 7-11). DOI: 10.1145/2594776.2594778.

Giles, C. (2021). Kabbalah and Tarot: The Tree of Life. Medium. https://medium.com/tarot-a-textual-project/kabbalah-and-tarot-the-tree-of-life-ef0c170390c9. Accessed 4 Jan 2022.

Law, E., von Ahn, L., & Mitchell, T. (2009). Search war: a game for improving web search. In Proceedings of *the acm sigkdd workshop on human computation* (pp. 31-31).

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, *1*(4), 309-317.

Ma, H., Chandrasekar, R., Quirk, C., & Gupta, A. (2009). Page hunt: improving search engines using human computation games. *In Proceedings of the 32nd international acm sigir conference on research and development in information retrieva*l (pp. 746-747).

Pathak, B., & Lal, N. (2017). Information retrieval from heterogeneous data sets using moderated IDF-cosine similarity in vector space model. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)* (pp. 3793-3799). IEEE.

Parlett, D. (2012). tarot game. *Encyclopedia Britannica*. https://www.britannica.com/topic/tarot-game. Accessed 26 December 2021.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613-620.

Schell, J.(2015). *The art of game design : A book of lenses (Second ed.)*. Boca Raton: CRC Press.

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval*. Cambridge University Press.

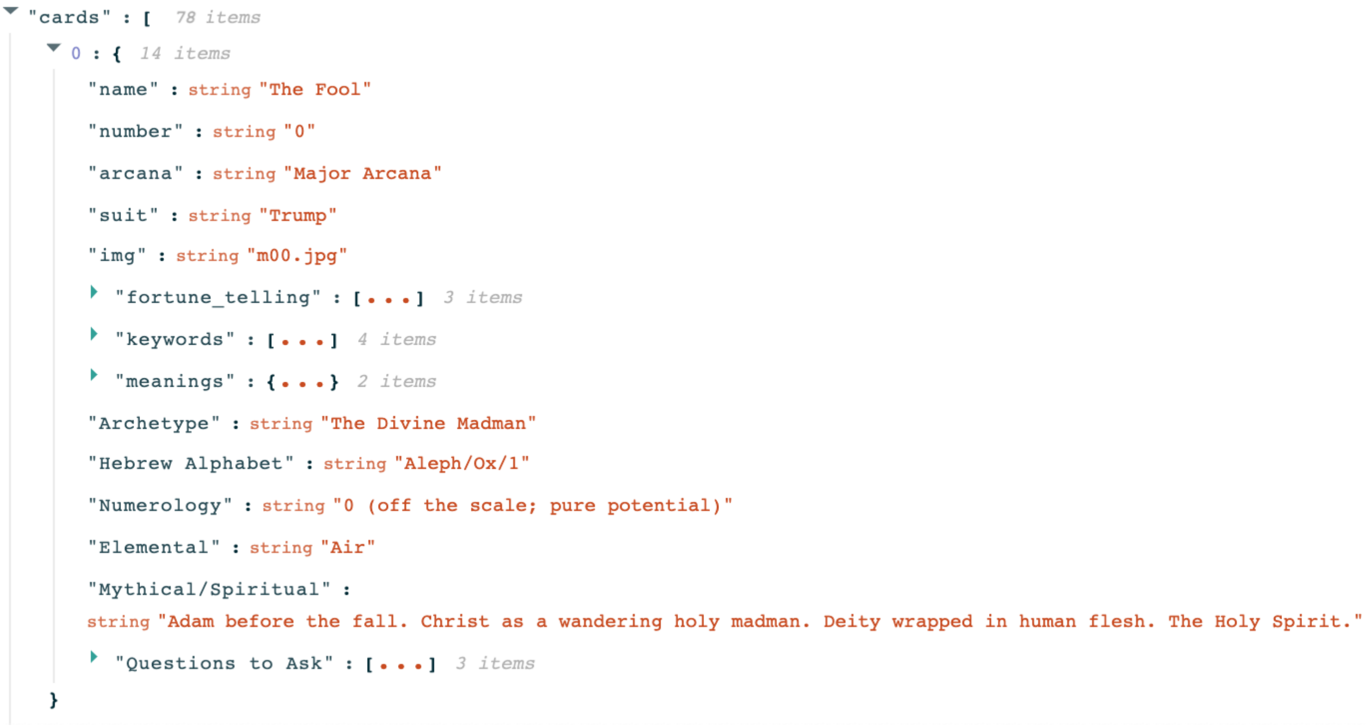
Singhal, A. (2001). Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, *24*(4), 35-43.

Solowij, A-M. (2016). Unreality: Next-Gen Fantasy, Utopias, and Unquantified Landscapes in a World of Hyper-Digital Realism. *J. Walter Thompson Intelligence*. Retrieved from https://www.jwtintelligence.com

Von Ahn, L., Liu, R., & Blum, M. (2006). Peekaboom: a game for locating objects in images. In Proceedings of the SIGCHI conference on Human Factors in computing systems (pp. 55-64).

APPENDICES

Appendix A: An example of a card entity in the Json file.



```
▼ "cards" : [  78 items
    ▼ 0 : {  14 items
        "name" : string "The Fool"
        "number" : string "0"
        "arcana" : string "Major Arcana"
        "suit" : string "Trump"
        "img" : string "m00.jpg"
        ▸ "fortune_telling" : [•••]  3 items
        ▸ "keywords" : [•••]  4 items
        ▸ "meanings" : {•••}  2 items
        "Archetype" : string "The Divine Madman"
        "Hebrew Alphabet" : string "Aleph/Ox/1"
        "Numerology" : string "0 (off the scale; pure potential)"
        "Elemental" : string "Air"
        "Mythical/Spiritual" :
        string "Adam before the fall. Christ as a wandering holy madman. Deity wrapped in human flesh. The Holy Spirit."
        ▸ "Questions to Ask" : [•••]  3 items
    }
```