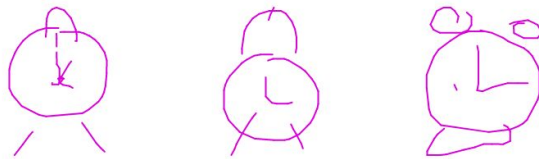
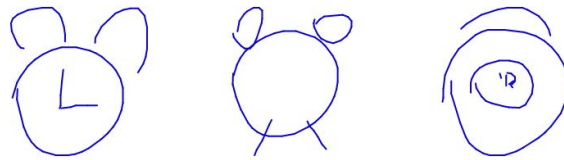
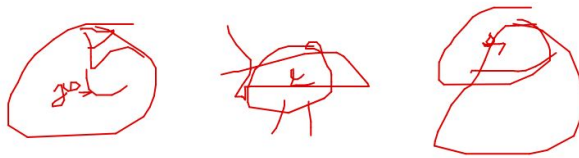


Appendix

ART AND MACHINE LEARNING
CMU 2019 SPRING
Final Project

AI Horoscopes



"Thinking Machine"

Jenn Choi, Julie Kim, Sophia Yoo

Appendix

Concept

In the web age of social media and memes and alternate identities, humans have gained an ability to exist within two worlds at once. The immense amount of faceless interactions in the digital world have led to our living in a state that lacks a quality of depth and introspection. Conventionally, humans have pursued depth primarily by finding meaning in life through religion. Although recent generations have broken out of more traditional constructs and belief systems, we still use heuristics to shape our values and understand how the world works.

A belief system that resonates with a surprising number individuals – especially in the United States, is Horoscopes. Horoscopes are a forecast of one's future based on the relative position of the stars and planets at one's birth. There are 12 horoscopes that align somewhat similarly to the months of a year:

Aries	Leo	Sagittarius
Taurus	Virgo	Capricorn
Gemini	Libra	Aquarius
Cancer	Scorpio	Pisces

Generally, these 12 horoscope signs are described as experiencing different personas and life events when the cosmos align a certain way. The ancient astrological study of personalities with respect to the alignment of stars has turned into a daily fortune teller of sorts, instructing people what events lie in their destiny and how to best handle them. The method of fortune-telling has become very trendy, with daily horoscope features on platforms such as newspapers, Facebook, and Snapchat

While many people read their daily horoscopes, they seldom take them to heart. A primary reason for this rejection is astrology's lack of scientific reasoning. As a group, we have also come to wonder how much truth there is behind horoscopes.

In this project, we investigate the reproducibility of horoscopes. The task that we are faced with is modeling approximately 10 years of daily horoscope data to generate what we hope to be a fun or eye-opening series of fortunes. These oracles will either be universally relatable and allow us to break down false constructs between ourselves, or give us the means to understand ourselves on a deeper level. At the end of it, we believe that anything that spurs self-introspection is incredibly valuable. Our ultimate motivation is to advocate for this form of thought in our audience.

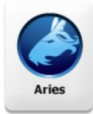
Our final project is an extension of project 3. The report and results of project 3 are included in the appendix.

Appendix

Process + Techniques

Data Collection & Pre-Processing (Sophia)

We used BeautifulSoup to scrape roughly 40,000 lines of data from a daily horoscope archive¹ dating back to August 15, 2009. Each page of the site contains a horoscope for each of the twelve astrological signs. The figure below is an example of the Aries horoscope for March 29, 2019:



• Aries - March 29, 2019

A certain person who does things their way and refuses to be bound by rules or restrictions could inspire you to break free in some way. However, their attitude and approaches might work for them but could bring a world of complications for you if you try to emulate them. Stick with what you know works or is above-board. This person's luck could run out sooner than they think. [Ask 1 free question to a Psychic](#)

Our scraper collected the date, sign, and horoscope of each entry. The data contains a somewhat limited vocabulary set in comparison to the large corpus, which is beneficial for the model to generalize to the dataset. We pre-processed the data to extract the day of week, and properly appended the day of week followed by a dash in the beginning of the horoscope. We wanted to do this because we wanted our horoscopes to be also conditional upon the day of the week, as we had more data (40k divided by 7, or the number of days in a week) when we condition it on day of week compared to day of year (40k divided by 365). Additionally, we removed all quotation marks excluding the apostrophes because we found from Project 3 that our model didn't do very well making sure that the quotation marks were opened/closed.

Conditional RNN-Sequence Modeling (Sophia)

We used code² available for lyric generation conditioned on artist to do a character-level sequence generation using Pytorch. The model uses Python to create a dictionary of all printable characters, with each character represented in a one-hot-encoded tensor. The model then generates sentences by sampling character by character, where we predict the next letter conditional on all of the previous letters, similarly to a Markov chain. The model uses a recurrent neural network (RNN), applicable to our task because it makes use of sequential information and uses output from the previous computation to compute the next character. In a way, RNNs contain contextual information because they store memory of what was computed before.

We tuned the parameters `batch_size` and `num_workers` and found that a batch size of 200 and 4 workers perform best. Due to the lengthy computation time, we tested what worked best for 3 iterations and adjusted accordingly.

The conditional RNN sequence model is conditioned on two parts: the horoscope sign and the input sequence, which we made the day of week. As mentioned in the pre-processing step, we made the day of week the input sequence to condition for which day of week the horoscope was for. We found that horoscopes were more optimistic for the week on Mondays ("Your

¹ <https://sagittarius.com/archive/>

² <http://warmingwinds.github.io/pytorch/rnns/2018/01/27/learning-to-generate-lyrics-and-music-with-recurrent-neural-networks/>

³ https://github.com/keras-team/keras/blob/master/examples/lstm_text_generation.py

⁴ <https://github.com/atpaino/deep-text-corrector>

Appendix

ambitiousness could be heightened at this time “) compared to on a Friday (“It's possible you've arrived where you are in a current situation through a trial-and-error process of some kind”).

We also tuned the temperature parameter, used to control the randomness. When the temperature is higher, we generate a higher volume of random horoscopes. When temperature is lower, we generate sentences that are more similar to sentences that we've seen before in our dataset. We did not want temperature to be too low, as we wanted our generated text to be somewhat unique – however, we also refrained from producing nonsensical gibberish. Given this, we decided that the optimal temperature for text that makes sense was between 0.8 and 1.0.

We generated 20 horoscopes for each horoscope sign for each day of week (20 x 12 x 7), using a random temperature between 0.8 and 1.0, and stored it in a data frame to be accessed when a page was shown.

Grammar and Spelling Checker (Sophia)

First, we split off the final sentence of the horoscope by cutting it off at the last punctuation mark of the horoscope. This was to ensure we had full, complete sentences. We used Python's grammar-check library to quickly modify any simple grammar errors to make sure the text was coherent/

Key Term Extraction (Jenn)

After generating the horoscopes by conditioning on the day of the week, we extracted the important symbolic keyword of the text. We used two methods to extract the keywords: python Textrank³ and NLP frequency extraction.

Python Text Rank, also known as pytextrank, is based on the NLP report written by Rada Mihalcea⁴. Pytextrank is a graph-based ranking algorithm for text analysis. Pytextrank first preprocesses the text by parsing the text to words with their respective part of speech. From the tokenized words, we generated a text relational graph. Words with one lexical units are added as nodes of the graph. If there a node with any possible co-occurrence relation with any other nodes, a edge is formed within them. We determined whether the two nodes are connected if their respective lexical units co-occur within a window of N words, where N words can range from two to ten. By analyzing the co-occurrence relation, we can determine if there are semantic links between two terms. As we iterate this process multiple times until it converges, each node calculates their own score of relational importance. Then by sorting, we were able to extract the most important word. However, for the extracting process, we outputted the words that are nouns.

NLP frequency extraction is simpler than the previous approach. Key term extraction was based on the machine learning article “Automated Keyword Extraction from Articles using NLP”⁵. Like the previous approach, we tokenized the text by lexical units and calculated their frequencies. We, then, extract the top noun that occurred the least amount of time. Through this key term extraction, we want to extract the most unique words. However, due to a simple model, the extracted key terms wasn't that unique or significant. So we also post processed the top 5 - 7

³<https://github.com/DerwenAI/pytextrank>

⁴<http://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf>

⁵https://medium.com/analytics-vidhya/automated-keyword-extraction-from-articles-using-nlp-bfd864f41b34?fbclid=IwAR2mDy9gQuOtrnzjUOg0O8bfStTnBjDblo7r7bXmYBKYktog-8_lieO_jk4

Appendix

noun terms with the lowest frequency by manually filtering terms that are too abstract or aren't objects. Filtered words included "something", "things", "someone", and "feel".

Using the two model, we choose key terms that were more suitable for the next process, the Sketch RNN model. Although we used two different methods to extract key terms, since the original texts used very similar and abstract terms, most of the generated key terms were repetitive and hard to input in the sketch rnn model. Since the input of sketch rnn model are usually objective nouns, abstract terms like "issue" and "effect" produced unrelated sketches.

Quick Draw and Sketch RNN (Sophia)

We used the text to art generation code⁶ from the `christmAls` written by Lj Miranda and 4 other contributors. Here, we only took the parts of the code that takes a the keywords generated in the previous step and generates the most similar Quick Draw category. Then, using this, we generated 9 sketches of this category as our image. We originally used style transfer in the `christmAls` code to generate more "artistic" work. However, we found that simple sketches left our images looking cleaner and more easily interpretable by the readers of the horoscope. We pre-generated the images for the horoscopes that we produced.

Image Manipulation (Jenn)

Despite the post processing, sentences in the text still lacked cohesion. Longer the sentence, the harder it was to understand what the sentence meant. When reading the text as a whole, it left us, the audience, with a slight confusion to what the horoscope exactly meant. After talking with the professor, we decided to split the text line by line in our interactive platform to help readers focus on individual sentences rather than the text as a whole. So, we split our key term generated sketch into three sections and alternated the lines from the text and the split images. We wrote a simple image manipulation script to crop each of the sketches into equal dimensions.

RShiny Application (Julie)

In creating a more immersive experience for our audience, we decided to use an interactive platform. By loading our output data into RStudio, we were able to build a RShiny application that takes an individual's birthday and the date, and randomly generates a horoscope-image pairing.

The application contains a `generate()` function that takes the birthday and date inputs to select random horoscopes that are conditional on horoscope sign and day of the week. The horoscopes are then fed into a `recombine()` function, which splits the horoscope paragraph into 4 groups and segments them to be readily available for presentation in line with 3 images. The text groups and image paths are then displayed in the application output, in alternating order. We chose this approach because the model-generated text contains stronger individual sentences than paragraphs. By using our images to split the sentences, the interface makes more sense. The layering of the images also creates more dimensionality on the page.

Audience Response

⁶<https://github.com/thinkingmachines/christmAls>

Appendix

There were a lot responses that our images were pretty interesting. They were very interested in how the images were related to the text. Some of commented that horoscope does sound pretty relevant to their situations. Some seems confused and unconvinced of the text.

“Keys Wow Wait some lines are repeated.”

“This is really cool, Can I take a picture of my horoscope?”

“I can related to some of them.”

“This sentence doesn't really make sense.”

Appendix

Results

My Horoscope

Please select your birthday and click `Reveal Horoscope`

Birthday

Month

January

Which day's horoscope would you like to see?

01-01-2019

Day

1

Reveal Horoscope

Your horoscope sign is: Capricorn

When someone saying, feel were not seeking differences in life.



That s playing the stops occurring in a cost in any way.



You undoubtedly have to seize an issue exists, and you might need to provide some level of passion for a while charge.



Rather than enjoy yourself and you the worst course for too late, then you might feel the norm of your mind is capable of completing trepidation coming your way.

Code: <https://github.com/jupiek/Alhoroscopes/>

Results: <https://github.com/jupiek/Alhoroscopes/tree/master/Alhoroscopes>

Interface: <https://jupiek.shinyapps.io/ArtxML>

Appendix

Project 3 Report:

<https://docs.google.com/document/d/1tqpza4z-2aOg4vvr7yIqCBVrgK1tBxhgfP3mMZwRZtw/edit>

Project 3 Results: https://github.com/seungye1/artxml3/final_results.txt

Project 3 Code: <https://github.com/seungye1/artxml3/>