Text Mining Twenty Thousand Leagues Under the Seas

louis.gauthy 6188059 & matthew.noack 6334589

May 2023

1 Introduction

Please find our code can be found in the follow repository: Github Link. "FinalTextMiningProject.ipynb" Runs the BERT and LDA programs. Make sure you change file_path variable to the correct file path and remove the mounting to the google drive if not being used. The notebooks rest of the notebooks can be run locally. "text and entity extraction" is the implementation of the named entity extraction using NLTK and Regular expression. Those extracted are visualised in "entity visualisations.ipynb", and evaluated in "entity_evaluation.ipynb". Relations between them are build and visualised in "relation_extraction". Finally, "topic_modeling.ipynb" provides the implementation for the NMF topic modeling and evaluation. All the figures of this report are included in the "plots" folder and all the data used are included in the "data" folder.

1.1 Text Mining Twenty Thousand Leagues Under the Seas

Jules Vernes wrote Twenty Thousand Leagues Under the Seas in March 1869 [Wik23]. This novel is a French classic book. The main characters are Professor Pierre Aronnax, the narrator, Captain Nemo, the captain of the *Nautilus* submarine, Conseil, the P. Aronax servant and Ned Land, a Canadian whale harpooner. The story describes the trip of the main character on board of the *Nautilus* mysterious submarine around the world in different fictitious places.

1.1.1 Problem description

This project had two objectives. First, we aim at extracting the relations between the PERSON and LOCATION entities contained in the book and visualise them in a relation graph. Second, since Twenty Thousand Leagues Under the Seas portrays the journey of a submarine around the world, we aim at visualising and extracting the different topics contained in the book. The topic modeling results may capture the different vocabulary used throughout the trip which may describe the different places.

1.2 Data Used

We use the full text from Guggenheim library [gug]. The full text is freely available and we did not need any preprocessing.

2 Methods

2.1 Entity extraction

2.1.1 NLTK- and REGEX-based NER

The first pipeline we used for extracting the LOCATION and PERSON for text is using NTLK python library [BKL09] and regex expression. We token the text in word tokens using the NLTK word tokeniser and we split the text into sentences using the NLTK Punkt Sentence Tokenizer. We entity extraction we then first assign a tag to each word using NTLK averaged perceptron tagger, then we group the tags into chunks using NLTK maxent named-entity chunker. This last pipeline extracts entities of types 'LOCATION', 'GPE', 'ORGANIZATION', 'PERSON', 'FACILITY' and 'GSP' and we will further only look at locations, persons and GPEs (geopolitical entities).

Then, our book has the specificity of containing several nautical coordinates. Those are given in specific formats, i.e. 42°15' N. lat. 60°35' W. long, we extract them the regular expressions. Unfortunately the reality is that coordinates are different throughout the book and we then design an algorithm for handling different format types, called 'extract_coordinates'. We first assume that every location must contain a degree component. We thus scan through the text to find matches of the degree symbol '°' or its literal version 'deg.'. Then for each match, we scan the neighborhood to extract if they exist the degree value, the minute value, and the carnality direction which could be in different forms: N., north or North, and coordination type: lat., Lat., or latitude. Then, the book contains coordination information in terms of meridians and parallel which are named latitude and longitude. For example, 'The boat passed the fifty-fifth meridian on Sunday'. We also extract those pieces of information and convert them to a numeric, latitude/longitude format. Lastly, we merge the latitude and longitude matches that belong to a unique coordinate based on their distance. For example, if the '°' symbol in 42°15' N. lat. is less than 30 characters away from the '°' symbol in 60°35' W. long, we consider both coordinates to belong to the same entity.

2.1.2 BERT-based NER

The other pipeline we used was BERT, using the pretrained model dbmdz/bert-large-cased-finetuned-conll03-english and the tokenizer model dbmdz/bert-large-cased-finetuned-conll03-english. Using this pretrained models, we were able to find LOCATION and PERSON entities in the text. First, we tokenized the text using the pretrained tokenizer and then find entities using the pretrained model.

We then organized the data into chapters, paragraphs, and 200 word segments in order to determine which PERSONS are in which LOCATIONS; this did not produce good results as characters locations are not determined by paragraph or 200 word segments, so we used longitude and latitude coordinates as referred in the above NLTK- and REGEX-based NER section.

2.1.3 Solving Entities Co-references

Throughout the novel, it may happen that composed entities such as 'Captain Nemo' is also referred to as 'Nemo' or 'Captain'. To resolve this issue, we merged composed entities and their components into a unique entity. More specifically, for each composed entity, we search if some of its components are contained in the Named Entity set. We ran into a clash for the 'Captain' component which was part of multiple composed entities such as 'Captain Anderson', and 'Captain Farragut'. We thus decided to assign the 'Captain' entity only the most commonly composed entity it is contained in, which is 'Captain Nemo'.

2.1.4 LLM for Named Entity Recognition

The last tool we considered for Named Entity Recognition is a large language model, more specifically we used the FLAN-T5 base model [CHL$^+$22], containing 250M parameters developed by Google and published in HuggingFace. The model is convenient for our application since it was trained, among others, on Named Entity Recognition tasks, and it achieves performances. We propose to the LLM in the following way. We formulate a task, e.g. 'Find the Named Entities in this sentence:' and we append it to the sentence examined.

2.1.5 NER Evaluation

For evaluating our NER methods, we use three metrics. The Recall is defined by $\frac{TP}{TP+FN}$ and the Precision is defined as $\frac{TP}{TP+FP}$, where TP is the number of True Positive, the entities tagged by both the ground truth and the test algorithm, FN is the number of False Negative which are the entities not discovered by the algorithm, and FP is the number of false positives which are the number wrong entity tags. Finally, we use the F1 score, defined as $2 \cdot \frac{precision \cdot recall}{precision + recall}$, which is a harmonic mean between the Precision and Recall

2.2 Coreference resolution

For completeness, we not only look at extracting the entities but also take into account their coreference, or finding all expressions that refer to the same entity in a text. This process mainly consists of matching pronouns or phrases to their source. For example, 'Captain Nemo', 'he', and 'the old man' would be linked to the same entity. We use the coreference implementation of FastCoref [OCG22] which uses a BERT pretrained model.

2.3 Relation Extraction

Once the Named Entities are extracted, we can compute their relations. The first approach we took to quantity the relation strength is by counting the number of sentence co-occurrence for each pair of PERSON and LOCATION entities.

2.4 Topic Modeling

For topic modeling, we investigated two popular methods, Latent Dirichlet allocation (LDA), and Non-Negative Matrix Factorization (NMF). For evaluating a topic modeling and in order to choose the optimal number k of topics, we use a coherence model measures the relative distance between words within a topic. We considered each paragraph of the text to be a different document.

Using LDA, we were able to find related concepts.

- $(0,' 0.018 * "platform" + 0.016 * "day" + 0.014 * "wa" + 0.014 * "clock" + 0.013 * "'")'$
- $(1,' 0.066 * "captain" + 0.053 * "nemo" + 0.024 * "wa" + 0.013 * "went" + 0.013 * "'")'$
- $(2,' 0.017 * "\_nautilus\_" + 0.015 * "air" + 0.015 * "wa" + 0.014 * "water" + 0.009 * "reservoir"')$
- $(3,' 0.091 * "\" + 0.087 * """ + 0.022 * "captain" + 0.021 * "said" + 0.017 * "conseil"')$
- $(4,' 0.017 * "\_nautilus\_" + 0.015 * "water" + 0.015 * "wa" + 0.014 * "mile" + 0.012 * "surface"')$
- $(5,' 0.017 * "could" + 0.014 * "door" + 0.013 * "opened" + 0.013 * "wa" + 0.012 * "wall"')$
- $(6,' 0.087 * """ + 0.085 * "\" + 0.028 * "ned" + 0.024 * "land" + 0.014 * "said"')$
- $(7,' 0.021 * "wa" + 0.013 * "'" + 0.012 * "sea" + 0.007 * "would" + 0.007 * "already"')$
- $(8,' 0.058 * "wa" + 0.014 * "\_nautilus\_" + 0.011 * "sea" + 0.010 * "long" + 0.007 * "u"')$
- $(9,' 0.019 * "wa" + 0.009 * "shell" + 0.009 * "panel" + 0.008 * "fish" + 0.006 * "thought"')$

Above is the output of using LDA with 10 topics and 5 words per topic. From this output, we can see some patterns. In $(2,' 0.017 * "\_nautilus\_" + 0.015 * "air" + 0.015 * "wa" + 0.014 * "water" + 0.009 * "reservoir"')$, we can see that is nautilus, water, reservoir, and air are all mentioned; this makes sense as the nautilus would be in water and has an air reserve.

3 Evaluation & Results

3.1 Named Entity Recognition

Using NLTK, 3747 entities were extracted. Their proportion is represented in 1.

Using the Nominatim API form GeoPy library [Con23], which acts as a gazetteer, providing coordinates of named entities based on the OpenStreetMap data, we can visualise the GPE entities in 6, and the LOCATION entities in 7. If we would try to model the Nautilus journey using the LOCATION entities, we would obtain an inconceivable itinerary, shown in 8. We then try to model it using the extracted coordinates with a regular expression procedure. The

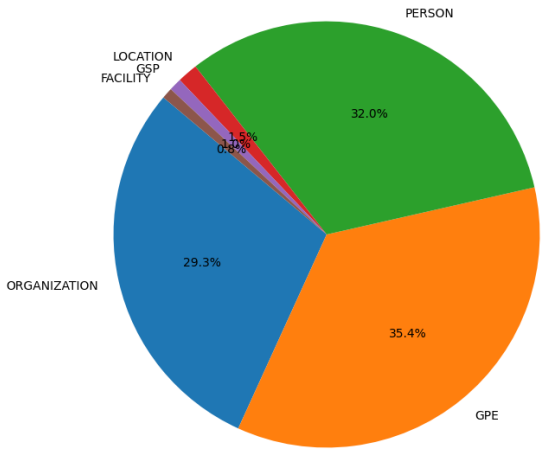

Figure 1: Pie plot of the different entity types extracted using NLTK

extracted coordinates are incomplete, for example the latitude is not always given or the cardinal direction of the coordinate might be missing. Thus we fill in the missing information by propagating the last valid observation forward to incomplete observations. We obtain the coordinates shown in 9. Linking the coordinate based on the chronology of the text, we obtain the itinerary shown in 2. For evaluation, we compare the model of the Nautilus journey with a ground truth extracted from an illustration of the journey contained in the book [Wik23]. We convert this illustration to a set of coordinates manually and we obtain a ground truth shown in 10. It is observed that the start of the modeled journey, in the northern Atlantic does not match with the one of the ground truth in the Japanese sea. It is later found by consulting the text that those first coordinates were not describing the position of the Nautilus but of some monster living in the Atlantic. Further, we observe that the rest of the modeled journey resembles the ground truth: the Nautilus starts by going deep East into the Pacific Ocean, then goes back west, south of Indonesia, reaches India, the red sea, the Suez Canal, the Mediterranean Sea, and finally going straight south to the South Pole and straight north the north pole in the Atlantic Ocean.

To examine the Precision, Recall, and F1 Score of NLTK evaluate those entities by annotating a sample of the text, chapters 3 and 4, and using it as a ground truth. We obtain $Recall = 32.3\%$, $Precision = 71.2\%$, $F1 = 44.4\%$.

Regarding the evaluation of BERT, we created the ground truth for chapter 3 and 4 and compared said truth to the entities that BERT found in chapter 3 and 4; below are the results. This better score of chapter 3 compared to chapter 4 is most likely due to the more complex PERSON and LOCATION entities in chapter 4 than in chapter 3. Chapter 3 - *Recall* : 50.0%, *Precision* : 57.0%, $F1$: 53.0% Chapter 4 - *Recall* : 45.0%, *Precision* : 37.0%, $F1$: 40.0% Chapter 3 and 4 - *Recall* : 47.0%, *Precision* : 48.0%, $F1$: 48.0%

We find that BERT has better recall and F1 score, but worse precision when compared to the NLTK model.

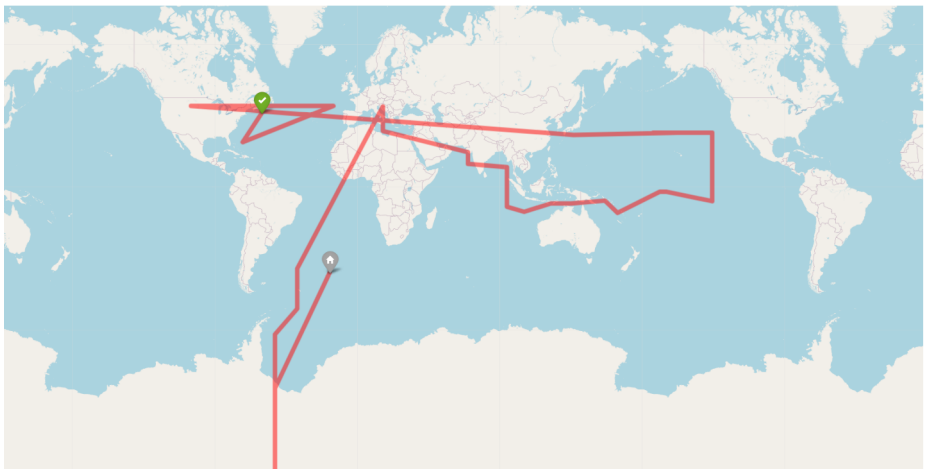

Figure 2: Map of the nautical coordinates extracted using a regular expression, an edge is drawn between each coordinate with its next coordinate in chronological order in the text. This is used for modeling the Nautilus route.

Further, it was observed in the plot of the nautical coordinates 2 some inconceivable routing behavior. Thus, we hypothesize that some of the coordinates do not describe the position of the *Nautilus* submarine. The characters may be talking about the position of something else such as a target destination. We then experimented with the use of LLM for filtering the sentences that describe the actual position of the character and their submarine. With our LLM framework described in 2.1.4, we build the following prompt: 'Are the protagonists talking about where they were? ' + sentence. Out of the 46 coordinate sentences, the model answered 'yes' only for 7 of them:

- The 13th of April, 1867, the sea being beautiful, the breeze favourable, the *Scotia*, of the Cunard Company's line, found herself in 15° 12' long and 45° 37' lat herself was going at the speed of thirteen knots and a half.
- The frigate was then in 31° 15' north latitude and 136° 42' east longitude.
- we were close to Vanikoro, really the one to which Dumont d'Urville gave the name of Isle de la Recherche, and exactly facing the little harbour of Vanou, situated at 16° 4' S lat, and

164° 32' E long.
- There Dillon learned the results of Dillon inquiries, and found that a certain James Hobbs, second lieutenant of the Union of Calcutta, after landing on an island situated 8° 18' S lat, and 156° 30' E long, had seen some iron bars and red stuffs used by the natives of these parts.
- Our course was directed to the west, and on the 11th of January Our doubled Cape Wessel, situation in 135° long and 10° S lat, which forms the east point of the Gulf of Carpentaria.
- On the 13th of January, Captain Nemo arrived in the Sea of Timor, and recognised the island of that name in 122° long.
- The morning of the 24th, in 12° 5' S lat, and 94° 33' long, we observed Keeling Island, a coral formation, planted with magnificent cocos, and which had been visited by Mr. Darwin and Captain Fitzroy.

3.2 Relation Extraction

Using the Networkx Python library [net], we visualise the PERSON, LOCATION relations where the nodes are the entities and the edges represent the sentence co-occurrences. Co-occurrence with values 1 or 0 are not considered in the graph, i.e. the pair of entities appeared only once, or never in the same sentence. Closer nodes have stronger relations. The figures, 11, 12, 3 show the extracted relation graphs from, simple sentence co-occurrence match, sentence co-occurrence match with merged composed entities and their components, sentence co-occurrence match with merged composed entities and their components on the co-reference resolved text, respectively. We first observed that our entity merging pipeline is still on perfect by observing the Ned Land has being decoyed into multiple entities such as 'Master Ned', 'Happy Ned'. Then we observe that the principal persons, in the center of the graph are Captain Nemo, Conseil and the Nautilus. Then, some context-based merging could also be done such as the character 'Ned Land' who is sometimes referred in the book as 'the Canadian'.

3.3 Topic Modeling

4 shows the relation between the number of topics and the coherence value. It is observed that $k = 3$ gives the highest coherence. We can visualise those three topics in 5 which shows the ten most relevant words per topic. We can observe that there is not a clear distinction between the topic. Words like 'water' and 'sea' are contained in multiple topics. We do not find meaningful topics. Nevertheless, we can extract some information out of the proposed three topics. Topic 1, containing 'said, sea, captain, nemo' may be related to orders given by or to the captain. Topic 2 may refer to orientating and navigating the submarine, with words like "long (longitude), feet, one, two, sea".

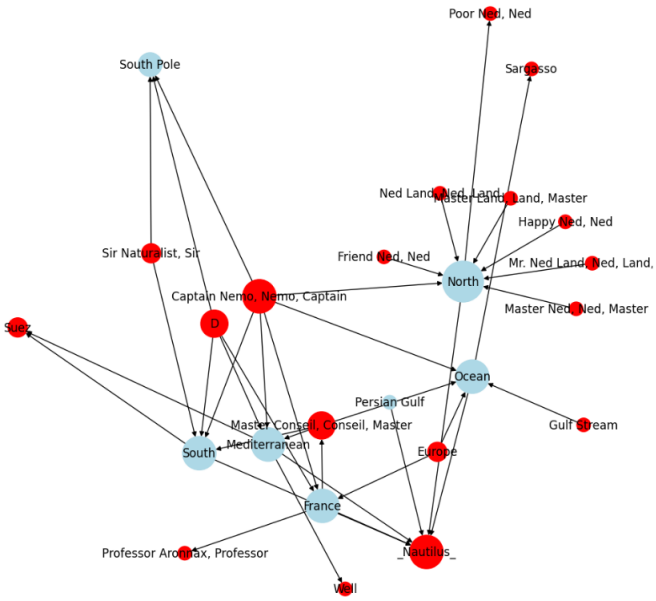

Figure 3: Sentence-Based relationships between PERSON in red and LOCATION in blue, extracted using NLTK on the coreference-resolved text, with FastCoref. The composed entities are merged.

4 Conclusion

In this text mining project *Twenty Thousand Leagues Under the Seas*, we have investigated the extraction of PERSON and LOCATION entities using NLTK and BERT. Then we modeled relation graphs between the PERSON and LOCATION. Finally, we investigated if some meaningful topics could be extracted from the book using LDA and NMF topic modeling methods.

We observed that our named entity extraction algorithm performance is relatively low, 44% F1 with NLTK and 48% with BERT. Hence there is here room for improvement. A legitimate path to explore would be to explore the use of LLM for understanding based on the context where the entities are in the sentence and most importantly if these entities are locations or persons. The named entity extraction could also be improved by resolving manually or automatically entity co-references such as 'Captain Nemo' and 'Captain' or 'Ned Land' who is sometimes referred to as 'the Canadian'. We provided one automatic approach, merging composed entities with their component but we experimented that it was not robust enough, see 3.2.

In BERT, we found a difference in precision, recall, and F1 score from chapters 3 and 4, with chapter 3 producing better results in all three categories. This could be due to the more complex PERSON and LOCATION entities in chapter 4 then in chapter 3, indicating that the underlying training data in the

pretrained dataset that was used in BERT could be changed to produce better results.

For linking PERSON and LOCATION, will need to create a model that can better connect LOCATION to PERSON; an idea would be to use sentiment analysis to find if a PERSON expresses anything about a certain LOCATION (ie PERSON states that they find the island warm).

Regarding topic modeling our experiments revealed some meaningful topics such as "nautilus, water, mile, surface" where it could be hypothesized that the characters are discussing about the submarine, or "long (longitude), feet, one, two, sea" may refer to orientating and navigating the submarine.

5 Overview of who did what

Louis worked on NER using NLTK and regular expression, on the visualisations of the extracted coordinates, on the sentence-based relations extraction and visualisation, and on topic modeling using NMF. Matthew worked on NER using BERT and on topic modeling using LDA.

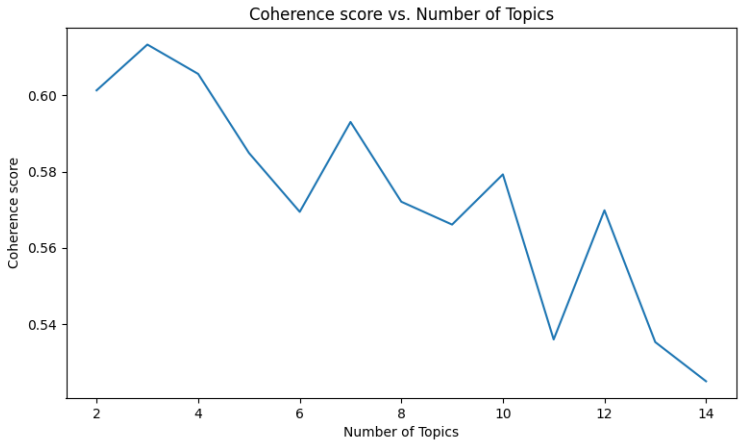

Figure 4: Investigation of number of topic choice influence on the coherence level

Figure 5: The ten most representative words for the different topics

A Appendix

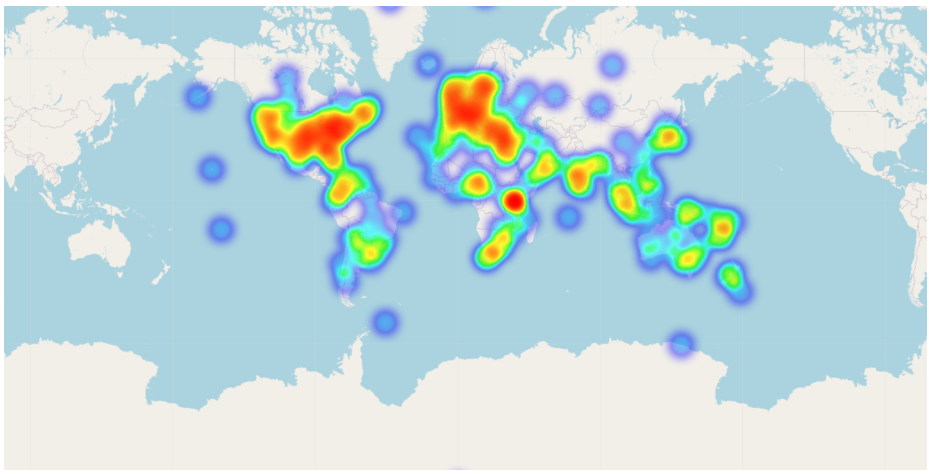

Figure 6: Heat map of the geopolitical entities (GPE) extracted using NLTK

Figure 7: Map of the LOCATION entities extracted using NLTK

References

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. 01 2009.

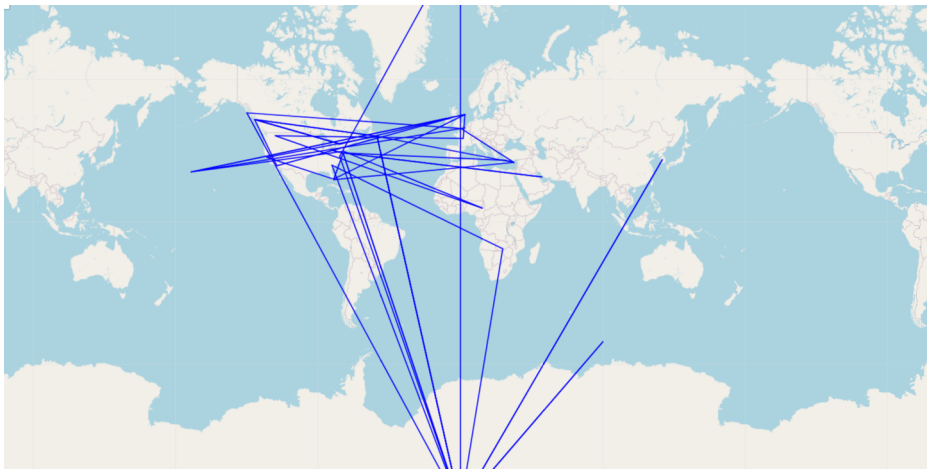

Figure 8: Map of the LOCATION entities extracted using NLTK, an edge is drawn between each of entities with its next entity in chronological order in the text

[CHL+22] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[Con23] GeoPy Contributors. Geopy, 2023.

[gug] Guggenheim library.

[net] Exploring network structure, dynamics, and function using networkx.

[OCG22] Shon Otmazgin, Arie Cattan, and Yoav Goldberg. F-coref: Fast, accurate and easy to use coreference resolution, 2022.

[Wik23] Wikipedia contributors. Twenty thousand leagues under the seas — Wikipedia, the free encyclopedia, 2023. [Online; accessed 27-May-2023].

Figure 9: Map of the nautical coordinates extracted using regular expression

Figure 10: Hand-mane ground truth of the Nautilus journey, based on 19^{th} illustrations contained in the book [Wik23]

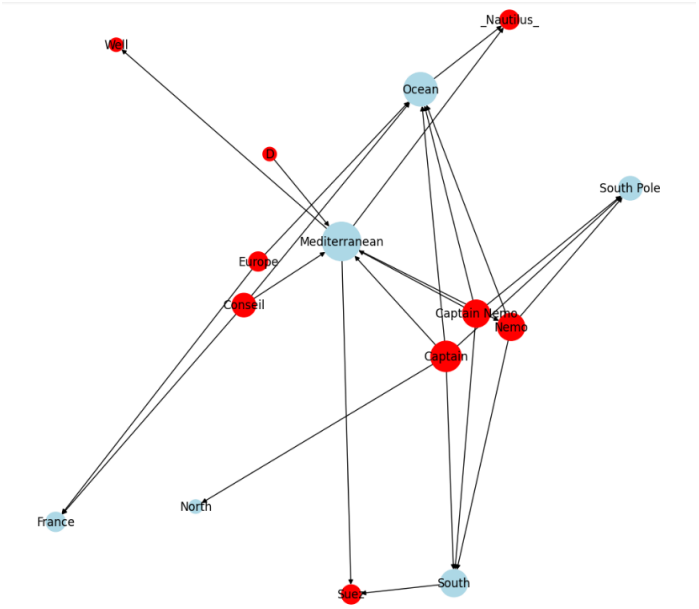

Figure 11: Sentence-Based relationships between PERSON in red and LOCATION in blue, extracted using NLTK.

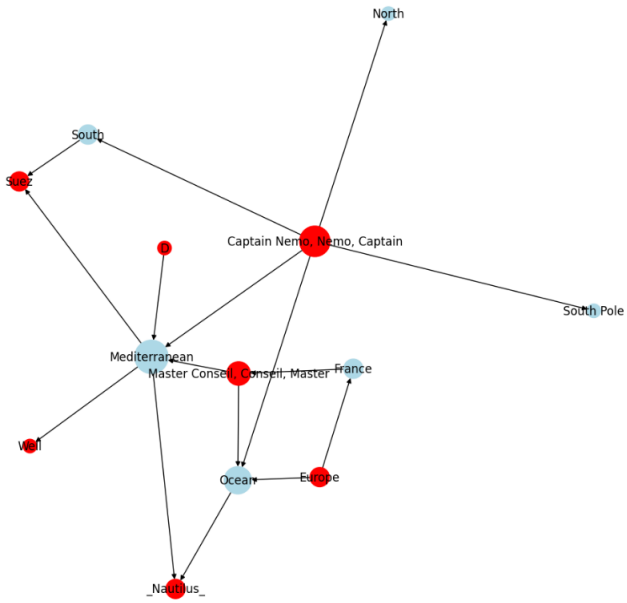

Figure 12: Sentence-Based relationships between PERSON in red and LOCATION in blue, extracted using NLTK. Here the composed entities such as 'Captain Nemo' and their components 'Nemo' are merged.