# Da Vinci Code: The Track of Robert Langdon

Kristian van Kuijk, Arthur Vieillevoye

May 2023

## Contents



# 1 Introduction

For this project, we extract information from Dan Brown's book *The Da Vinci Code* [1] with different natural language processing (NLP) and text mining techniques [2]. More specifically, we aim to extract the places visited by the characters throughout the book.

[1] The text file of the entire book was extracted here.
[2] Our Git repository can be found here.

After performing some coreference resolution with Lingmess and SpanBERT, we use BERT and RoBERTa for named entity recognition (NER), normalizing the output with soft TF-IDF and Levenshtein distance. Afterward, we map characters to location and lastly perform some topic modeling comparing LDA and BERTopic. Finally, we end this report by revealing the different sites visited by Robert Langdon throughout his quest.

# 2 Methods & Results

Throughout this section, we introduce the tasks (and put them into the context of our goal), the methods used, and the performance of the different techniques, with an evaluation for each task.

## 2.1 Coreference Resolution

Coreference resolution is a crucial task in NLP that involves identifying and linking expressions in a text that refer to the same entity [4].

Coreference resolution helps in better comprehension of textual data and information extraction tasks. By resolving coreferences, we can determine which pronouns or other expressions refer to the same entity. It allows us to build a more accurate representation of the information and understand the relationships between different parts of the text, such as named entity recognition and relation extraction. Thus, Coreference resolution can enable more precise extraction of relevant information from the text. For instance, we will map Robert Langdon to Paris in the example "*Robert Langdon is in Paris. He is now leaving for London.*". However, we will miss the mapping of Robert Langdon to London due to coreference. That is why coreference resolution must be applied.

### 2.1.1 LingMess

Current coreference systems are based on a single pairwise scoring component, which assigns to each pair (c, q) a score reflecting their tendency to refer to each other, where c is a "candidate span" and q is a "query span" which appears before c in the sentence. LingMess has six types of coreference decisions [3] and learn a dedicated scoring function for each category. According to Otmazgin, Cattan, and Goldberg [3], it improves the accuracy of the pairwise scorer and the overall coreference performance.

### 2.1.2 SpanBERT

SpanBERT extends BERT architecture to perform better at tasks involving span-level predictions, such as question answering and coreference resolution.

[3] PRON-PRON-C: compatible pronouns based on their attributes such as gender, number, and animacy (see Appendix C for more details), PRON-PRON-NC: incompatible pronouns, ENT-PRON: a pronoun and another span, MATCH: non-pronoun spans with the same content words, CONTAINS: one contains the content words of the other, OTHER: all other pairs.

Along with the masked word prediction objective used in BERT, SpanBERT also includes "masked span objective." SpanBERT randomly masks sections of text rather than just individual words. It allows the model to capture context across longer time scales, such as phrases or sentences, in addition to the context at the word level.

### 2.1.3 Evaluation

To evaluate our two techniques, we took a small sample of the book (25 sentences). Then, we applied our two algorithms to the sentences and extracted the results. Afterward, we compared the results produced by the two algorithms with the coreference labels we extracted (ourselves) from the sentences.

| | Positive | Negative | Total |
|---|---|---|---|
| Positive | 19 | 0 | 19 |
| Negative | 16 | 219 | 235 |
| Total | 35 | 219 | 254 |

Table 1: Confusion matrix of LingMess

| | Positive | Negative | Total |
|---|---|---|---|
| Positive | 33 | 0 | 33 |
| Negative | 2 | 219 | 221 |
| Total | 35 | 219 | 254 |

Table 2: Confusion matrix of SpanBERT

With those two tables, it is possible to compute the accuracy, the recall, and the precision of the two algorithms:

- LingMess:
  - $accuracy = \frac{tp+tn}{tp+tn+fp+fn} = \frac{19+219}{19+219+0+16} = \frac{238}{254} = 0.937$
  - $precision = \frac{tp}{tp+fp} = \frac{19}{19+0} = 1$
  - $recall = \frac{tp}{tp+fn} = \frac{19}{19+16} = \frac{19}{35}$
  - $f1 = 2 \times \frac{precision \times recall}{precision+recall} = 2 \times \frac{19}{54} = \frac{19}{27} = 0.704$
- SpanBERT:
  - $accuracy = \frac{tp+tn}{tp+tn+fp+fn} = \frac{33+219}{33+219+0+2} = \frac{252}{254} = 0.992$
  - $precision = \frac{tp}{tp+fp} = \frac{33}{33+0} = 1$
  - $recall = \frac{tp}{tp+fn} = \frac{33}{33+2} = \frac{33}{35}$

– $f1 = 2 \times \frac{precision \times recall}{precision + recall} = 2 \times \frac{33}{68} = \frac{33}{34} = 0.971$

| | Accuracy | Precision | Recall | $f_1$ |
|---|---|---|---|---|
| LingMess | 0.94 | **1** | 0.54 | 0.7 |
| SpanBERT | **0.99** | **1** | **0.94** | **0.97** |

Table 3: LingMess and SpanBERT performance comparison for NER on a test set of 25 sentences.

As we can see, SpanBERT performs better than LingMess. All of the quality measurements, except the precision, show the SpanBert winner. Even if SpanBERT and LingMess have the same precision, SpanBERT makes the difference with its higher recall. An additional point of the SpanBert that can be noted is that it also does the coreference learning for the objects. In contrast, LingMess does not (at least on our test set). Another element that made the difference is that SpanBert performed better on the dialog. It successfully recognized the characters while LingMess struggled.

## 2.2 Named Entity Recognition (NER)

We perform NER to obtain all characters and locations across the text. In this project, we use pre-trained and pre-fine-tuned models available on Hugging Face. More specifically, we compare the performance of BERT and Roberta for NER. Both models are fine-tuned with the following labels: O - Outside of a named entity; MISC - Miscellaneous entity; PER - Person's name; ORG - Organization; LOC - Location.

### 2.2.1 Evaluation

To evaluate the two NER algorithms, since we do not have any label data for *The Da Vinci Code*, we manually evaluated a small set of data (25 sentences, a total of 214 words)—those sentences where then fed to the model to extract the location and the characters. Then we compared their results with the expected output.

| | Positive | Negative | Total |
|---|---|---|---|
| Positive | 7 | 3 | 10 |
| Negative | 2 | 202 | 204 |
| Total | 9 | 205 | 214 |

Table 4: Confusion matrix of Bert

| | Positive | Negative | Total |
|---|---|---|---|
| Positive | 9 | 1 | 10 |
| Negative | 1 | 203 | 204 |
| Total | 10 | 204 | 214 |

Table 5: Confusion matrix of Roberta

With those two tables, it is possible to compute the accuracy, the recall, and the precision of the two algorithms:

- Bert:
  - $accuracy = \frac{tp+tn}{tp+tn+fp+fn} = \frac{7+202}{7+202+3+2} = \frac{209}{214} = 0.977$
  - $precision = \frac{tp}{tp+fp} = \frac{7}{7+3} = \frac{7}{10}$
  - $recall = \frac{tp}{tp+fn} = \frac{7}{7+2} = \frac{7}{9}$
  - $f1 = 2 \times \frac{precision \times recall}{precision+recall} = 2 \times \frac{7}{19} = \frac{14}{19} = 0.737$
- Roberta:
  - $accuracy = \frac{tp+tn}{tp+tn+fp+fn} = \frac{9+203}{9+203+1+1} = \frac{212}{214} = 0.991$
  - $precision = \frac{tp}{tp+fp} = \frac{9}{9+1} = \frac{9}{10}$
  - $recall = \frac{tp}{tp+fn} = \frac{9}{9+1} = \frac{9}{10}$
  - $f1 = 2 \times \frac{precision \times recall}{precision+recall} = 2 \times \frac{9}{20} = \frac{9}{10} = 0.9$

| | Accuracy | Precision | Recall | $f_1$ |
|---|---|---|---|---|
| BERT | 0.98 | 0.70 | 0.78 | 0.74 |
| RoBERTa | **0.99** | **0.90** | **0.90** | **0.90** |

Table 6: BERT and RoBERTa performance comparison for NER on a test set of 25 sentences.

Results are summarized in Table 6. As expected, Roberta has a higher $f_1$ value than Bert. The difference and the better $f_1$ score of Roberta can be explained by the fact that Roberta is an optimized version of Bert. First, RoBERTa has been trained on a more extensive and diverse dataset. Secondly, while Bert has been trained using masked language modeling (MLM) and next sentence prediction (NSP), RoBERTa is trained with dynamic masking, full-sentences without NSP loss, large mini-batches, and a larger byte-level Byte-Pair Encoding [2].

## 2.3 Normalization

The output of our NER methods needs to be normalized. We want to map all instances of Robert Langdon to a unique name, regardless of whether he is referred to in the sentences as Robert, Mr. Langdon, or any other name. To do so, we started by scrapping all the book's characters. We then compared the list of names we scrapped with all *PER* instances outputted by our NER. We used two similarity metrics: Soft-TF-IDF and Levenshtein distance. Note that in Soft-TF-IDF, a higher score means higher similarity, while for Levenshtein distance, a higher distance means less similarity. Overall, we noticed Soft-TF-IDF is a better fit for our case, as shown in Tables 7 and 8. For the remaining part of the project, we used Soft-TF-IDF with a set threshold of 0.4.

| *PER* instance | Best Match Scrapped name | Similarity score |
|---|---|---|
| Langdon | Robert Langdon | 0.58 |
| Robert Hanssen | Robert Langdon | 0.34 |
| Sauniere | jacques saunière | 0.0 |

Table 7: Soft-TF-IDF similarity examples.

| *PER* instance | Best Match Scrapped name | Distance score |
|---|---|---|
| Langdon | Robert Langdon | 8 |
| Robert Hanssen | Robert Langdon | 4 |
| Sauniere | jacques saunière | 9 |

Table 8: Levenshtein distance examples.

## 2.4 Mapping Characters to Location

After our NER output is normalized, we map *PER* to *LOC* instances (characters to location). We use a brute-force naive approach, where for each *PER* instance, we look for the closest *LOC* instance in the text in both directions. While such a naive approach might be prone to error, we noticed that in most cases, this approach's outputs are correct.

## 2.5 Topic Modelling

We perform topic modeling with Latent Dirichlet Allocation and BERTopic. BERTopic uses Sentence-BERT to build 384 dimensional embeddings per document. We then apply UMAP for dimensionality reduction, mainly to make the clustering more efficient. For clustering, we employ HDBSCAN, a hierarchical density-based clustering algorithm. Lastly, we obtain our topic representations from clusters with c-TF-IDF, generating candidates by extracting class-specific words. To improve our topic representations, we finish the pipeline with Maximum Candidate Relevance. Note that while we follow the same pipeline as

Grootendorst [1], each component could be replaced by alternatives. For instance, one could employ PCA instead of UMAP for dimensionality reduction.

#### 2.5.1 Evaluation

The first clusters for BERTopic were generated by feeding each chapter as a document (we have a total of 105 chapters). Interestingly, only two clusters were outputted. We compared those two clusters by counting the occurrences of characters within each topic. The *hero* Robert Langdon and *villain* Silas had opposite frequencies among the topics (results shown in Table 9). For the first topic, Langdon occurred 27 times more than Silas, while for the second topic, Silas occurred 8 times more. More detailed results can be found in Appendix B, including the intertopic distance map for both methods (Figure 1 and 2 and the similarity matrix for BERTopic (Figure 3). In Appendix B, we documents are of length five sentences to obtain more topics.

| Character occurrences | Topic 1 | Topic 2 |
|---|---|---|
| Langdon | 3486 | 99 |
| Silas | 130 | 782 |

Table 9: How frequent are Silas and Robert across the two topics?

To evaluate the performance of both algorithms, we compare the coherence score in Table 10. In topic modeling, the coherence score is a measure that helps evaluate the quality of the topics generated by a topic model. It provides a quantitative assessment of how interpretable and coherent the topics are. The coherence score is calculated based on the degree of semantic similarity between the words within a topic. There are different methods to compute coherence scores. This report uses one of the most common approaches known as the *c_v coherence measure*. Higher coherence scores indicate more coherent and interpretable topics. Both models obtain a similar score, with a slightly higher score for BERTopic.

| | LDA | BERTopic |
|---|---|---|
| Coherence score | 0.451 | 0.459 |

Table 10: Coherence score

# 3 Discussion & Conclusion

## 3.1 So, where's Robert Langdon?

After running our complete NLP and Text Mining pipeline described in this report, we found around 500 locations mapping to Robert Langdon. It makes sense since our NER also picks up sublocation, such as *kitchen*, or *Pavillon*

*Dauphine* on top of general location as *Paris* or *Vatican City*. Here is an example of the first 10 locations visited by Robert Langdon according to our pipeline: *['Grand Gallery', 'Hotel Ritz Paris', 'Chartres Cathedral', ' Vatican', 'Paris', 'City of Lights', 'Pavillion Dauphine', 'Vatican', 'Louvre', 'Vatican City']*. The total output can be found in Appendix A.

# 4 Future work

A few points can be improved in our approach. First, the handling of negation. Not paying attention to the negation can cause several problems. For example, in the sentence *Robert Langdon is not going to Paris.*, the algorithm will not notice the negation and return that Robert visited Paris.

Another limitation is the context. It is not easy with our approach to make the difference between a place actually visited and a place that is referenced in the text/dialogs of the book. For example, in the following sentence: *In Paris, Robert talks about the Vatican.* The algorithm might misbehave and return that Robert is in Paris and then in the Vatican.

## 4.1 Conclusion

In this project, we implemented and compared several NLP and Text Mining techniques and algorithms. It included coreference resolution with LingMess and SpanBERT, named entity recognition with BERT and RoBERTa, similarity measures with Soft-TF-IDF and Levenshtein Distance, and last but not least, Topic Modelling through LDA and BERTopic. Overall, our complete pipeline for the final output composed of SpanBERT, RoBERTa, and Soft-TF-IDF was selected given the evaluation we performed.

**Who wrote what:** Section 1 - Kristian and Arthur, Section 2.1 - Arthur, Section 2.1.1 - Arthur, Section 2.1.2 - Kristian, Section 2.1.3 - Arthur, Section 2.2 - Kristian, Section 2.2.1 - Arthur, Section 2.3 - Kristian, Section 2.4 - Kristian, Section 2.5 - Kristian, Section 2.5.1 - Kristian, Section 3 - Kristian, Section 4 - Kristian van Arthur

# References


[1] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[2] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[3] Shon Otmazgin, Arie Cattan, and Yoav Goldberg. "Lingmess: Linguistically informed multi expert scorers for coreference resolution". In: *arXiv preprint arXiv:2205.12644* (2022).

[4] Rhea Sukthanker et al. “Anaphora and coreference resolution: A review”. In: *Information Fusion* 59 (2020), pp. 139–162.

# A All the places visited by Robert Langdon

Grand Gallery, Hotel Ritz Paris, Chartres Cathedral, Vatican, Paris, City O Lights, Pavilion Dauphine, Vatican, Louvre, Vatican City, Rue La Bruyere, Opera House, Paris, Europe, Ritz, Eiffel Tower, Rome, Eiffel Tower, France, Tuileries, Tuileries Gardens, Seine, Ramses, Musee du Louvre, Louvre, Paris, Louvre, Paris, Louvre, DENON, Paris, Denon Wing, Louvre, Denon Wing, France, Louvre, Grand Gallery, Louvre, Murray Hill Place, Grand Gallery, Washington Monuments, Vatican Secret Archives, Rome, Louvre, United States, Venus, Hollywood, Church, Vatican, United States, Venus, Langdon, Louvre, Saint-Sulpice, Grand Gallery, Paris, Vatican, National Gallery, Paris, Capitaine, Britain, U.S., States, U.S., Paris, Mount Vesuvius, Church of Saint-Sulpice, U.S., Grand Gallery, Louvre, Paris, France, Paris, France, Place Saint-Sulpice, England, Louvre, Paris, Denon Wing, U.S., Louvre, Grand Gallery, Collet, Paris, Place du Carrousel, Paris, Place du Carrousel, Carrousel, Seine, Pont du Carrousel, Louvre, Grand Gallery, Louvre, Grand Gallery, Louvre, Parthenon, Louvre, Denon Wing, Mona Lisa, Grand Gallery, Cambridge, Mona Lisa, Wheeling, Saint- Sulpice, Salle des Etats, France, Priory, P.S., Priory, Mona Lisa, U.S., Seine, Paris, Salle des Etats, Essex County Penitentiary, Grand Gallery, Louvre, Paris, Salle des Etats, The Priory, Priory, Sion, Louvre, Saint-Sulpice, Mona Lisa, Salle des Etats, Louvre, Saint-Sulpice, Denon Wing, Carrousel du Louvre, Rue de Rivoli, Rivoli, Louvre, Champs- Elysees, Madonna of the Rocks, London, Champs-Elysees, Hotel de Crillon, Paris, Avenue Gabriel, Champs-Elysees, Madonna of, Priory of Sion, Priory, Ritz, Louvre, Champs-Ely sees, Champs-Elysees, Boulevard Malesherbes, Gare Saint-Lazare, Paris, U.S., Lyon, Rue de Clichy, Montmartre, Priory, Rue Haxo, Priory, Salle des Etats, Paris, Priory of Sion, Heaven, Priory of Sion, Priory, Sion, Holy Land, Pri, Europe, Priory of Sion, United Kingdom, Sangreal, Paris, Holy Grail, Priory of Sion, Holy Grail, New York, Holy Grail, Rue La Bruyere, Bois de Boulogne, Allee de Longchamp, Rue Haxo, Roland Garros, Priory, England, Holy Grail, Grail, Priory of Sion, Normandy, Roland Garros, Rue Haxo, Switzerland, Castel Gandolfo, Zurich, Louvre, Paris, Louvre, Priory of Sion, Louvre, Rosewood, Priory, St. Thurial, The Way, clo, Grail, Priory, Keystones, Priory, Holy Grail, Priory, Holy Grail, Priory, apreuve de merite, Priory, Priory of Sion, Priory, Paris, Priory, Priory of Sion, Pri, Priory, Castel Gandolfo, Rome, Priory, The Priory, of Sion, Holy Grail, Versailles, France, Priory of Sion, Teabing, Paris, Holy Grail, Paris, Holy Grail, Philadelphia, Priory of Sion, Holy Grail, Grail, Holy Grail, Grail, la Petite Versailles, British, Chateau Villette, Grail, Paris, Henley, Chateau Villette, Isis, Grand, Americaine, Oxford, Grail, Priory of Sion, Holy Grail, Priory of Sion, Grail, Holy Grail, Christianity, Roman Catholic, Qumran, Vatican, Holy Grail, Grail, Venus, Holy Grail, Grail, Holy Grail, Louvre, Versailles, Holy Grail, skitoma, Grail, Our Lady of Paris, Nag Hammadi, Saint Peter, Grail, David, Grail, Heavens, Holy Grail, Church, Priory, Venus, Priory of Sion, Sangreal, Priory, Sauniere, Louvre, Grail, Holy Grail, Grail, Garden of Eden, Holy City, The Priory, key, Paris, Priory, S, Priory, Holy Grail, Priory, Rome, Grail, Priory, Rome, Vatican, Priory, Sangreal, Priory,

France, keystone, Chateau Villette, Hun, Tyrrhenian Sea, Paris, Grail, Madrid, Bene, Grail, keystone, Boston, Grail, Chateau Villette, Isles, France, U.S., Isle of Avalon, Britain, Grail, New York, Paris Louvre, Paris, Priory of Sion, Priory, Grail, Priory, Le Bourget Air, London, Le Bourget Airfield, Holy Grail, Priory, Holy Grail, Chateau Villette, Le Bourget Airfield, England, Sub Rosa, Nekkudot, Fogg Museum, Europe, Venus, Vatican, Abracadabra, Priory, Atbash, Holy Grail, Kent, Holy Grail, Hieros Gamos, Isis, Nirvana, Hieros Gamos, Temple, Harvard, Normandy, Hieros Gamos, Priory of Sion, English Channel, Hieros Gamos, Royal Holloway, Holloway, Sheshach, Sophia, Priory, Holy Grail, Britain, London, United Kingdom, England, London, England, Biggin Hill, Louvre, Biggin Hill Airport, Kent, Fleet Street, London, Tiber River, London, Temple Church, Fleet Street, Grail, Victoria Embankment, England, Holy Grail, Louvre, Thames, Sangreal, Inner Temple Lane, Rome, England., the Temple, Sangreal, London, Grail, London, Temple Church, London, Vatican, Priory, Chateau Villette, Temple Church, Chateau Villette, Temple, Grail, District and Circle Line, London, Grail, London, Temple Church, London, Chateau Villette, Horse Guards Parade, Heavens, London, Harvard, London, Rome, London, England, The Grail, London, Chateau Villette, St, Centre, London, Priory of Sion, Bayreuth, Grail, TAROT, London, Westminster Abbey, London, Priory of Sion, London, Westminster Abbey, Temple Church, London, Holy Grail, Westminster Abbey, Louvre, Poets' Corner, Grail, Pis, Madonna of the Rocks, Priory, Chapter House, College Garden, Great Britain, Chapter House, ANERY, Chapter House, Pyx Chamber, Chapter House, Sangreal, Priory, Sangreal, Priory, Chateau Villette, Chapter House, Temple Church, Sangreal, Chateau Villette, Grail, Priory, Chateau Villette, Temple Church, Saint-Graal, Grail, Chapter House, College Garden, Britain, Priory, Paris, Louvre, Grail, College Garden, Grail, Leigh Teabing, Chapter House, Grail, APPLE, Chapter House, Holy Grail, Kensington Gardens, Paris, Rosslyn, London, Holy Grail, Rosslyn, Priory, Rosslyn, Solomon's Temple, Grail, Rosslyn Chapel, ROSLIN, Rosslyn, Rosslyn Chapel, Rosslyn, Solomon's Temple, Rosslyn, Holy Grail, Rosslyn, Holy Grail, Grail, Rosslyn, Holy Grail, Priory, Grand-pere, Rosslyn, Priory, Rosslyn, Paris, Priory, Westminster Abbey, Grail, Sangreal, Rosslyn, Rosslyn Chapel, Star of Dav, Holy of Holies, Grail, Holy Grail, Grail, Rosslyn, Paris, Venus, Paris, Florence, Hotel Ritz Paris, Rue des Petits Champs, Rue Richelieu, Paris, Rue de Rivoli, Paris, Louvre, Rose Line, Holy Grail.

# B Topic Modelling

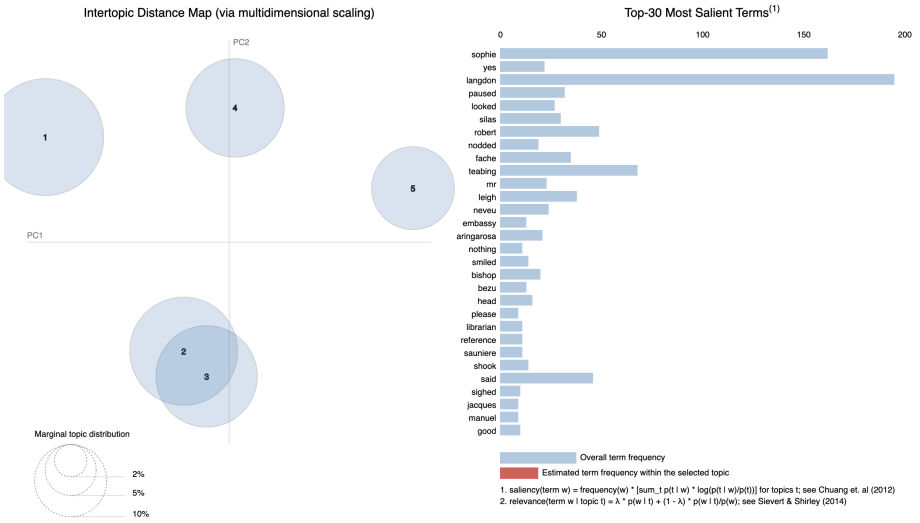


Figure 1: LDA Intertopic Distance Map

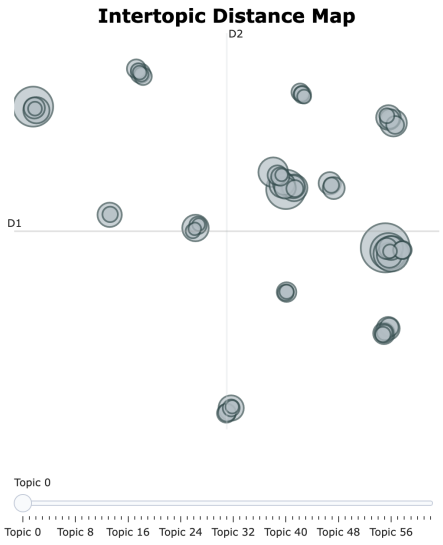


Figure 2: BERTopic Intertopic Distance Map for documents of size five sentences

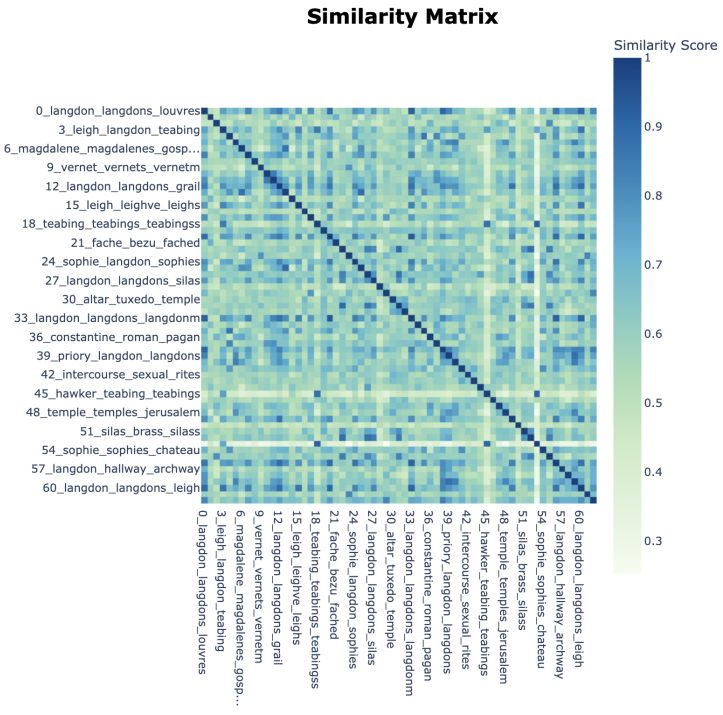


Figure 3: Similarity Matrix BERTopic