Table of Contents

1. Introduction

In natural language processing (NLP), topic modeling is a technique used to uncover the underlying themes within a collection of documents. One of the most widely used libraries for NLP tasks is NLTK (Natural Language Toolkit), which offers various functionalities for text processing and analysis.

However, a growing challenge within NLP is how well models trained on contemporary data perform when applied to older texts. Language evolves, and documents from different time periods may reflect shifts in vocabulary, syntax, and style. This raises an important question: can topic modeling tools, like NLTK, maintain their effectiveness across texts from different eras, or does their performance degrade when analyzing older documents?

Based on this question the hypothesis of this project is: **"Topic modeling using NLTK preprocessing exhibits lower performance on older texts compared to newer texts."**

This hypothesis stems from the assumption that NLP models like NLTK are optimized for more recent language patterns, and may struggle with older documents where linguistic conventions differ.

This project aims to explore and quantify this hypothesis by applying topic modeling to a collection of German political council meeting documents spanning different 5-year-spans. The goal is to compare the quality of the topics extracted from older documents with those from more recent ones and determine if there is a notable decline in performance when analyzing older texts.

2. Scope of the Project

This project leverages various techniques and tools covered throughout the course, making it both relevant and appropriate for this class. Several key topics form the foundation of this project:

- **Pickle for Data Storage**: The processed results are stored in pickle files, allowing for efficient data saving and retrieval.
- **XML File Parsing**: The German council documents used in this project are provided in XML format. The ElementTree library is used to extract the raw text data from the .xmi files.
- **NLTK**: NLTK is used extensively for text preprocessing tasks such as tokenization, stopword removal, and preparing the data for topic modeling.
- **Preprocessing Data**: Data preprocessing steps like stopword removal, tokenization, and POS tagging were crucial to cleaning the text data for analysis.

The following tools and topics were not explicitly covered in the course, but were essential for the project:

- **Gensim**: This library was used for topic modeling. While Gensim wasn't covered, **Scikit-learn** was, and both libraries offer similar approaches to topic modeling, making it a natural extension of the course material.

- **Matplotlib**: Visualizations, such as word clouds and bar charts, were created using Matplotlib, which was discussed briefly in the course but not explored in detail.
- **OS Module**: Iterating through directories to process multiple documents was necessary to automate and scale the analysis efficiently.

The project is ambitious in that it tackles the complex task of comparing topic modeling performance across different time periods. However, it remains manageable due to the use of pre-existing NLP libraries such as **NLTK** and **Gensim**, as well as well-structured, preprocessed datasets.

By using established NLP libraries and predefined datasets, the project ensures that the focus remains on analysis and hypothesis testing, rather than reinventing lower-level processing tools. This balance makes the project feasible while still offering valuable insights into the effectiveness of topic modeling over time.

3. Data and Tools

3.1. Datasets

The primary dataset used for this project is the **GerParCor corpus**, specifically focusing on the **Bundesrat** protocols from **1949 to 2025**. The data is divided into five-year intervals, with each interval containing multiple .xmi files that represent the textual content of the council's proceedings.

Each Bundesrat file provides comprehensive records of the Bundesrat-Sitzungen (sessions of the German council). These protocols cover similar political and administrative topics over the decades, which helps create a consistent basis for topic modeling. The .xmi files include both the preprocessed text (tokenization, POS tagging, etc.) using Spacy, as well as the raw text data. Since the goal of this project is to evaluate topic modeling based on NLTK, I opted to parse the files for their raw text and implement topic modeling from scratch using NLTK's preprocessing tools.

The use of consistent subject matter between older and newer meetings allows for a more reliable and consistent comparison of topic modeling results across different time periods. This uniformity ensures that the evaluation of the topic modeling is focused on the model's performance rather than variations in the content itself.

3.2. Tools and Libraries

Several tools and libraries were used to preprocess, analyze, and visualize the data:

- **NLTK (Natural Language Toolkit)**: Used for text preprocessing tasks, such as tokenization, stopword removal, and lemmatization. NLTK played a central role in preparing the data for topic modeling.
- **Gensim**: Applied for **Latent Dirichlet Allocation (LDA)** topic modeling.
- **Matplotlib**: Used for creating visualizations such as word clouds and bar charts to visually represent the results of the topic modeling and lemma frequency analysis.

- **Pickle**: Employed for saving and loading intermediate results, ensuring that processed data and models could be reused without reprocessing the raw data each time.
- **OS Module**: Utilized for iterating through directories and automating the processing of large datasets split into multiple folders.

4. Methodology and Implementation

The main objective of this project is to evaluate whether topic modeling using NLTK preprocessing performs differently when applied to older texts compared to newer texts. To achieve this, I implemented a pipeline that handles the preprocessing, topic modeling, and evaluation phases in a structured manner. Below is a breakdown of each step and how it contributes to the overall goal of the project.

4.1. Data Preprocessing

The first step was to prepare the text for topic modeling. Since the GerParCor corpus provides both preprocessed text and raw text data, I chose to work with the raw text in order to apply NLTK-based preprocessing.

This process ensures consistency in how older and newer documents are processed, allowing for a fair comparison. The preprocessing pipeline involved:

- **Lowercasing**: Standardizing the text to avoid issues with case sensitivity.
- **Removing punctuation and special characters**: Cleaning the text to focus on meaningful content.
- **Tokenization**: Splitting the raw text into individual words.
- **POS filtering**: Using NLTK's POS tagger to keep only nouns, verbs, and adjectives—categories that are typically most relevant in topic modeling.
- **Stopword removal**: Eliminating common words that do not contribute to identifying topics (e.g., "and," "the").
- **Lemmatization**: Reducing words to their base forms to ensure that different forms of the same word (e.g., "running" vs. "run") are treated as identical.

4.2. Topic Modeling

To perform topic modeling, I used Gensim's LDA (Latent Dirichlet Allocation). The model was trained on preprocessed text from both older and newer documents to extract topics. The following steps were taken:

- **Creating a Dictionary and Corpus**: A dictionary was created to map the unique words in the text to unique IDs. The corpus, representing the frequency of each word in each document, was then generated.
- **TF-IDF Transformation**: Term Frequency-Inverse Document Frequency (TF-IDF) was applied to weigh the importance of words based on how often they appear across documents.

- **LDA Model Training**: The LDA model was trained using a predefined number of topics. This step produced 15 topics for each set of documents, with each topic represented by its most frequent words.

The goal was to compare the topics extracted from older documents with those from newer documents to see if there is a difference in the model's ability to extract coherent and relevant topics across different time periods.

4.3. Evaluation Using Jaccard Similarity and Intersection Percentage

To evaluate the model's performance, I employed the following metrics:

- **Lemma Frequency Analysis**: For each document, the 10 most frequent lemmas were identified.
- **Jaccard Similarity Calculation**: The Jaccard similarity score was calculated by comparing the intersection and union of the top lemmas and the most frequent words in the LDA topics for each document.
- **Intersection Percentage**: In addition to Jaccard similarity, the percentage of overlapping words between the most frequent lemmas and the LDA topic words was calculated. This provided a more granular view of how much overlap there was between the frequent lemmas and the identified topics, offering a complementary perspective to Jaccard similarity.

4.3.1. Clarifying the Metrics

1. **Jaccard Similarity in the Context of This Project:**

   Jaccard similarity measures the overlap between the most frequent lemmas (key terms from the document) and the words generated by the LDA topic model. It shows how much common ground exists between the significant terms from the original text and the topic words identified by the model. Higher Jaccard similarity means a closer alignment between the model's output and the actual document content, while lower similarity indicates a mismatch.
2. **Topic Intersection Ratio:**

   This metric represents the percentage of overlap between the most frequent lemmas and the topic words generated by the LDA model. Unlike Jaccard similarity, which provides an overall ratio based on intersection and union, the topic intersection ratio offers a direct percentage of commonality, showing how well the original document's key terms align with the generated topics.
3. **Difference Between Jaccard Similarity and Topic Intersection Ratio:**

   While both metrics measure the overlap between frequent lemmas and topic words, they do so in different ways. Jaccard similarity considers the size of the intersection relative to the union of both sets, providing a balanced measure that accounts for disparities. In contrast, the topic intersection ratio is a straightforward percentage of overlap without considering the union. Essentially, Jaccard similarity offers a broader view of how well the model captures the original content, while the topic intersection ratio provides a more direct measure of commonality.

These metrics allowed me to assess how closely the topics generated by the LDA model aligned with the most frequent words in the text, providing a way to measure the model's performance on older versus newer documents.

4.4. Visualizations and Additional Insights

To provide further insights, Matplotlib was used to create visualizations, including:

- **Word Clouds**: These visualizations show the top words for each topic, offering a qualitative view of the model's results.
- **Bar Charts**: Bar charts were used to visualize lemma frequency in individual documents, providing a way to compare the relative importance of different words.
- **Interactive Normal Distributions**: Jaccard similarity scores across multiple time periods were visualized with normal distribution plots to show the spread of similarity scores and allowing comparison of document sets from different decades.
- **Topic Intersection Ratio Table**: A table summarizing the percentage of overlap between topic words and lemmas for different time periods was also generated, providing another quantitative measure of the model's performance.

These visualizations helped in manually inspecting the quality of the topics and how well they align with the most frequent words, especially across different time periods.

4.5. Implementation and Research Questions

Each step in the methodology directly supports the research question of whether topic modeling using NLTK preprocessing performs differently on older versus newer texts:

- **Preprocessing**: Ensures that older and newer documents are treated consistently, which is essential for a fair comparison.
- **Topic Modeling**: Extracts topics from both time periods, providing a basis for comparison.
- **Evaluation via Jaccard Similarity and Intersection Percentage**: Quantitatively measures the model's performance by assessing how well the topics align with the most frequent words, along with the percentage of overlapping words.
- **Visualizations**: Provide a qualitative assessment of the coherence and relevance of the topics extracted by the model.

By combining these methods, this project is able to determine whether NLTK-based topic modeling struggles with older texts compared to newer ones, as hypothesized.

5. Code & Modules

The project consists of three core modules that work together to achieve the primary goal of comparing the performance of topic modeling on older and newer documents. The pipeline is designed in a modular fashion, allowing each step to handle a specific part of the analysis, from data preprocessing to final evaluation. Below is an in-depth breakdown of each module and its role in the project.

5.1. Data Preprocessing (lda_topic_modeling.py)

The first step in the pipeline is data preprocessing, where the raw text from the GerParCor dataset is parsed and prepared for topic modeling. This step ensures that both older and newer documents are processed uniformly, allowing for a consistent comparison. The key components of the preprocessing step include:

- **Text Normalization**:
    - Converts all text to lowercase to avoid case sensitivity issues.
    - Removes punctuation and special characters to reduce noise.
    - Tokenizes the text using NLTK’s word_tokenize function, splitting the raw text into individual words.
- **POS Tagging and Filtering**:
    - Applies Part-of-Speech (POS) tagging to categorize words into nouns, verbs, and adjectives, which are typically most relevant for topic modeling.
    - Filters out irrelevant parts of speech to focus on content-rich words.
- **Stopword Removal**:
    - Uses NLTK’s stopword list to remove common words (e.g., "the," "and") that do not contribute to the meaning of a topic.
- **Lemmatization**:
    - Reduces words to their base forms (lemmas) to ensure that different forms of the same word are treated as identical. This step helps improve the coherence of topics by treating "run," "running," and "ran" as the same word.

The output of this module is a set of cleaned and lemmatized tokens that are ready for topic modeling.

5.2. Topic Modeling (lda_topic_modeling.py)

After preprocessing, the next step is topic modeling, where Latent Dirichlet Allocation (LDA) is applied to extract topics from the preprocessed documents. The key steps involved in topic modeling are:

- **Dictionary and Corpus Creation**:
    - A dictionary is created using Gensim, which maps unique words in the documents to unique IDs.

    - A corpus is generated, representing the frequency of each word in each document.
- **TF-IDF Transformation**:
    - Term Frequency-Inverse Document Frequency (TF-IDF) is applied to give more weight to words that are frequent within a document but less frequent across the corpus. This helps reduce the influence of commonly used words that may not be meaningful in distinguishing topics.
- **LDA Model Training**:
    - Gensim's LDA is used to extract topics from the documents. Parameters such as the number of topics and topn (the number of top words per topic) are predefined. In this case, the model is trained with 15 topics and topn=25, meaning the 25 most relevant words per topic are extracted.
    - The choice of parameters, including the number of topics and the number of passes during training, was made based on the need to capture meaningful topics across both older and newer documents.

The output of this step is a list of topics with their corresponding top words, as well as the dominant topic for each document. This information is saved to pickle files for later analysis.

5.3. Analysis and Visualization (analyze_data.py)

After topic modeling, the next module focuses on analyzing the results, calculating similarity metrics, and visualizing the outputs. This step evaluates the model's performance using Jaccard similarity and intersection percentages, while also providing visual aids like word clouds. The key components of this module include:

- **Jaccard Similarity Calculation**:
    - Jaccard similarity is used to measure the overlap between the most frequent lemmas (words) in a document and the top words in the LDA-generated topics. This provides a quantitative measure of how well the extracted topics align with the most frequent content in the document.
    - For each document, the 10 most frequent lemmas are compared to the top words in the dominant topic, and the Jaccard similarity score is computed based on the intersection and union of the two sets.
- **Topic Intersection Percentage**:
    - In addition to Jaccard similarity, the percentage of topic words that overlap with the most frequent lemmas is calculated. This gives a more granular view of the similarity between the extracted topics and the document's most frequent words, complementing the Jaccard score.

- **Word Cloud Visualization**:
    - Word clouds are generated for each topic to provide a view of the top words within each topic. This helps in visualizing how topics are formed and whether they are coherent across different time periods.
- **Bar Charts**:
    - Bar charts can be used visualize the frequency of lemmas within individual documents, helping compare the relative importance of different words and their overlap with topic words.

The information generated with this module is saved to pickle files for further analysis.

5.4. Comparing Results (compare_results.py)

In addition to individual analysis, the project also allows for the comparison of results across multiple time periods. This module is particularly important for evaluating whether topic modeling performs differently on older versus newer documents. The key features of this module include:

- **Loading Jaccard Similarity Data**: This script loads the Jaccard similarity data and intersection percentages from multiple pickle files generated for each time period.
- **Visualizing Results**:
    - **Histograms**: The Jaccard similarities across different time periods can be displayed as histograms, making it easy to visually compare the distribution of similarity scores between older and newer documents.
    - **Interactive Normal Distribution Plots**: A normal distribution curve for each set of results is plotted, allowing for an intuitive comparison of the data distributions. Users can toggle visibility for different time periods using interactive checkboxes.
    - **Intersection Ratio Table**: This script also generates a table comparing the percentage of overlap between topic words and frequent lemmas for different time periods, providing a quantitative comparison of topic modeling effectiveness over time.

By comparing these results, the project can measure whether topic modeling is more effective for newer texts than older texts, based on both Jaccard similarity scores and intersection percentages.

5.5. How Each Module Contributes to the Research Question

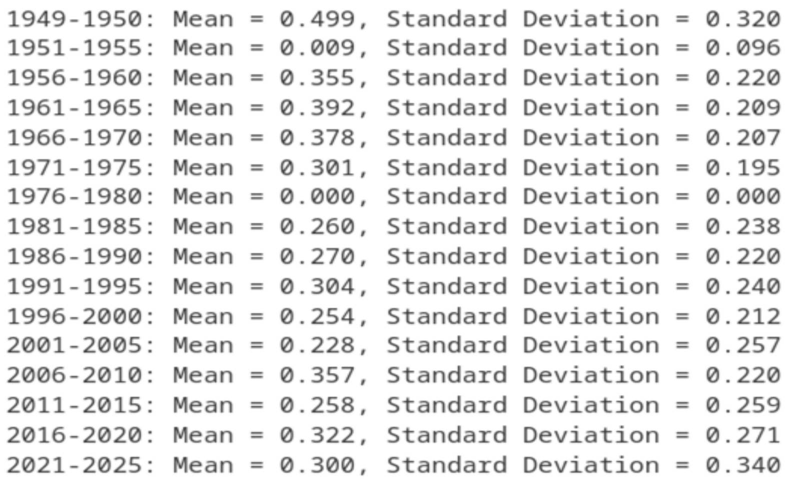
- **Preprocessing** ensures that documents from different time periods are treated consistently, removing potential biases while NLTK forms the basis of the research question.
- **Topic Modeling** extracts key topics for each set of documents, providing a foundation for comparison.

- **Evaluation** through Jaccard similarity and intersection percentage provides both a quantitative and qualitative way to assess the model's performance.
- **Visualizations** make it easier to interpret the results and compare the effectiveness of the model on older vs. newer texts.

By aligning each module with the overarching research goal, the project provides a thorough investigation of how well topic modeling using NLTK and Gensim performs on documents from different time periods.

6. Analysis

6.1. Jaccard Similarity Analysis



```
1949-1950: Mean = 0.499, Standard Deviation = 0.320
1951-1955: Mean = 0.009, Standard Deviation = 0.096
1956-1960: Mean = 0.355, Standard Deviation = 0.220
1961-1965: Mean = 0.392, Standard Deviation = 0.209
1966-1970: Mean = 0.378, Standard Deviation = 0.207
1971-1975: Mean = 0.301, Standard Deviation = 0.195
1976-1980: Mean = 0.000, Standard Deviation = 0.000
1981-1985: Mean = 0.260, Standard Deviation = 0.238
1986-1990: Mean = 0.270, Standard Deviation = 0.220
1991-1995: Mean = 0.304, Standard Deviation = 0.240
1996-2000: Mean = 0.254, Standard Deviation = 0.212
2001-2005: Mean = 0.228, Standard Deviation = 0.257
2006-2010: Mean = 0.357, Standard Deviation = 0.220
2011-2015: Mean = 0.258, Standard Deviation = 0.259
2016-2020: Mean = 0.322, Standard Deviation = 0.271
2021-2025: Mean = 0.300, Standard Deviation = 0.340
```

Figure 1: Overview of the Jaccard Similarity mean and standard deviation for each time span

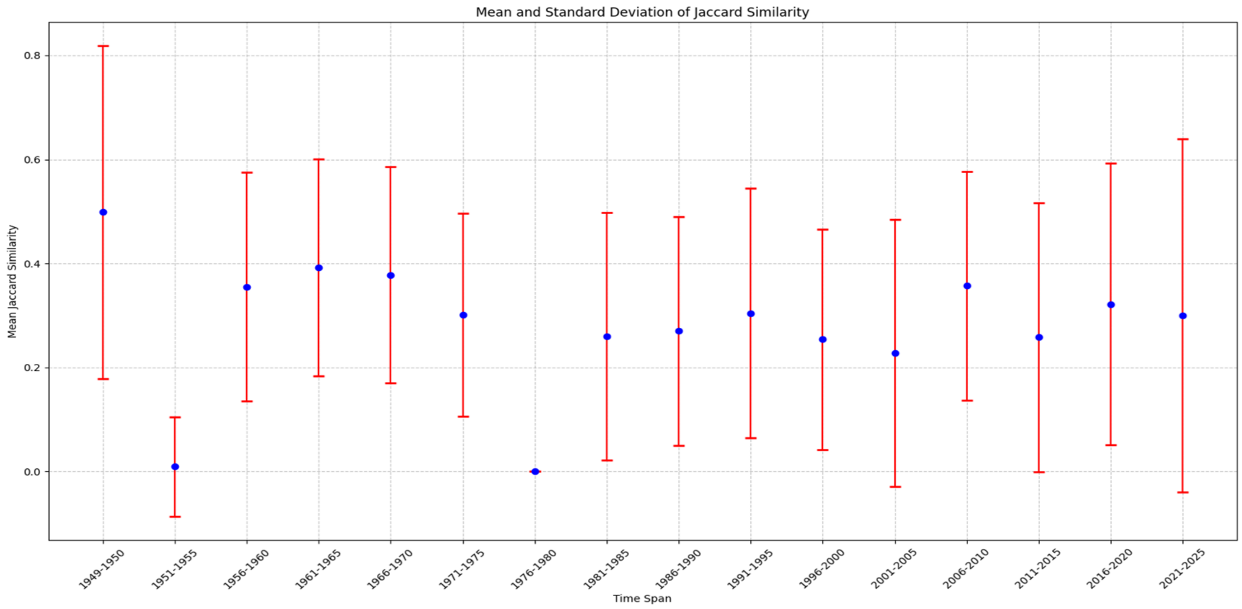


Figure 2: Distribution of the mean and standard deviation for Jaccard Similarity

The Jaccard similarity measures the overlap between the most frequent lemmas in the documents and the words generated by the LDA model for each topic. The overall mean Jaccard similarity is highest for the earliest time frame (1949-1950) and gradually decreases in more recent decades (see Figures 1 and 2), indicating that the topic modeling was more effective on older documents. However, the outliers significantly impact this pattern, with the most prominent being the 1951-1955 and 1976-1980 time spans.

Outliers in Jaccard Similarity:

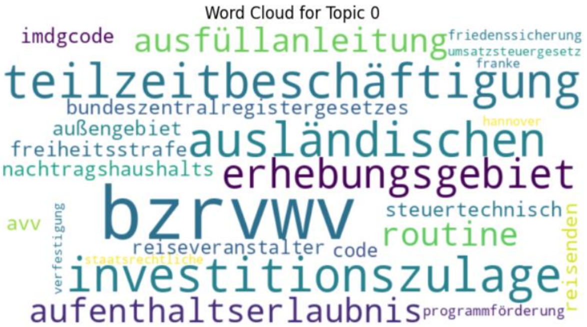


Figure 1: Wordcloud topic example 2 - 1951-1955

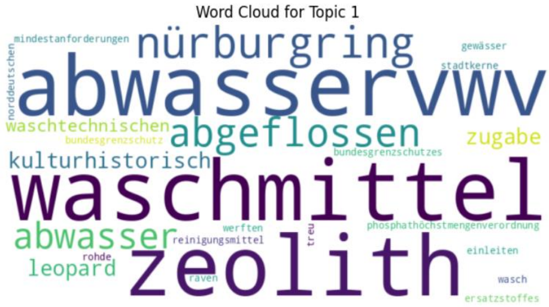


Figure 2: Wordcloud topic example 1 - 1951-1955

- **Outlier 1951-1955:** Despite having a variety of topics, topic words, and weights (see Figures 3 and 4) , almost all Jaccard similarities are recorded as 0. This suggests a potential data processing issue for this period, resulting in it being an unreliable representation of the topic modeling effectiveness.

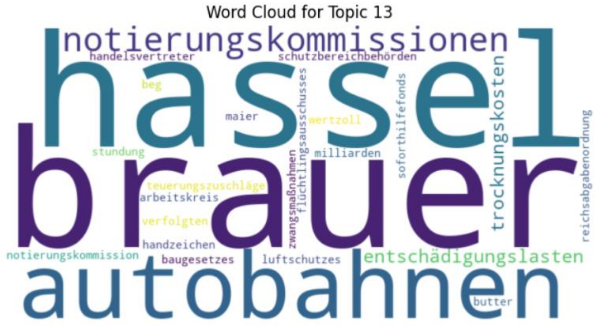


Figure 3: Wordcloud topic example 1 - 1976-1980

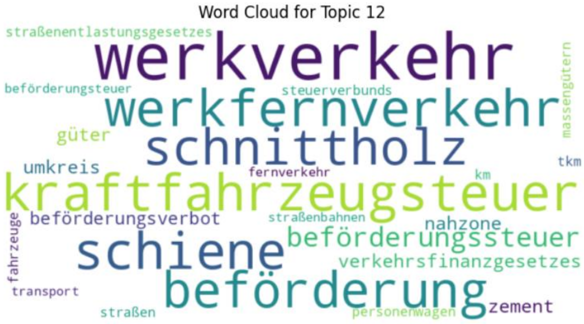


Figure 4: Wordcloud topic example 2 - 1976-1980

- **Outlier 1976-1980:** Similarly, the data for this period shows all Jaccard similarities as 0, indicating that there might have been an error during the data analysis phase.

Therefore, these two periods will be disregarded when evaluating the hypothesis, as they do not accurately reflect the model's performance. The remaining time spans a gradual decrease in similarity before stabilizing, meaning that topic modeling becomes less effective over time, especially from the 1970s onwards.

Standard Deviation Insights

The standard deviation generally trends downward from the 1940s up to the 1990s, then increases again in recent decades. This pattern suggests that the model's effectiveness became more consistent over time, with less variation in similarity scores between individual documents. However, this stability begins to wane in the more recent decades, indicating that the model's performance is less predictable in modern texts.

6.2. Normal Distribution Analysis

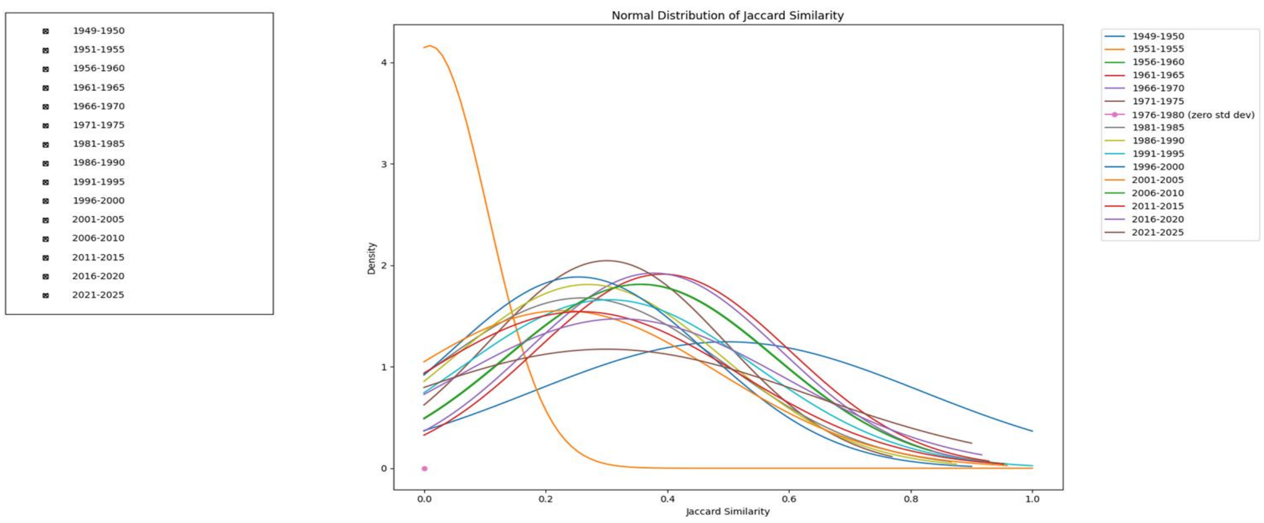


Figure 5: Normal Distribution for each time span of the dataset

The normal distribution graph visualizes the density of the Jaccard similarities, showing how frequently different similarity values occur. The density value represents how often a specific Jaccard similarity value is observed (see Figure 7).

Starting from 1971-1975, there is a noticeable leftward shift, indicating that the topic modeling results tend to have lower Jaccard similarity values as we approach more recent documents. This reinforces the idea that older documents are better represented by the topic model compared to newer ones.

Outliers in Normal Distribution (see Figure 9):

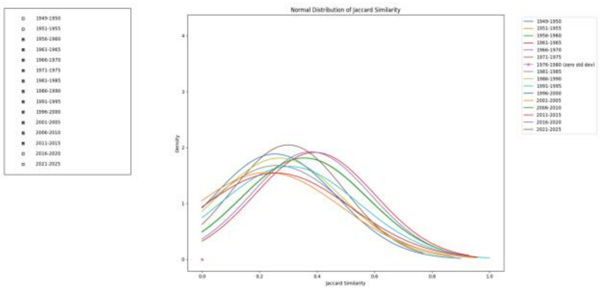


Figure 6: Normal Distribution without outliers

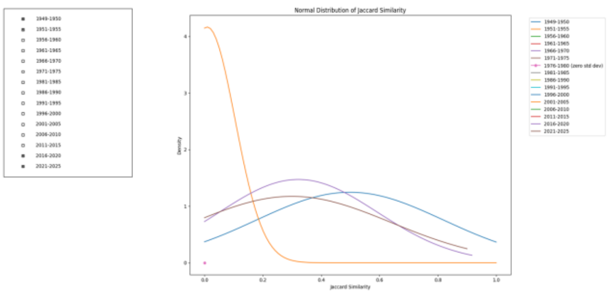
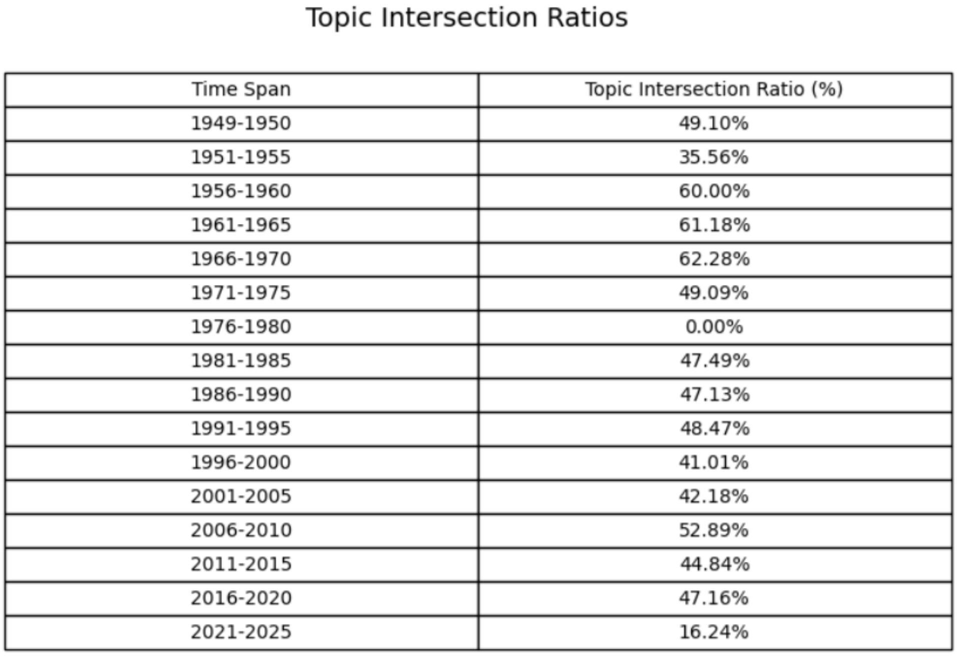


Figure 7: Normal Distribution of outliers

- **1949-1959:** This period is an outlier with a distinctly different distribution, likely because the timeframe of this dataset is smaller (only two years of data), resulting in more homogenous topics.
- **1951-1955:** As previously mentioned, data quality issues make this period an outlier.
- **2016-2020 and 2021-2025:** Both show a wide variety of Jaccard similarity values, suggesting more diverse topics or potential inconsistencies in how topic modeling was applied. The low number of documents in the 2021-2025 period likely contributes to its deviation from other periods.

6.3. Topic Intersection Ratio Analysis



Topic Intersection Ratios

Time Span	Topic Intersection Ratio (%)
1949-1950	49.10%
1951-1955	35.56%
1956-1960	60.00%
1961-1965	61.18%
1966-1970	62.28%
1971-1975	49.09%
1976-1980	0.00%
1981-1985	47.49%
1986-1990	47.13%
1991-1995	48.47%
1996-2000	41.01%
2001-2005	42.18%
2006-2010	52.89%
2011-2015	44.84%
2016-2020	47.16%
2021-2025	16.24%

Figure 8: Table of Topic Intersection Ratios

The topic intersection ratio measures the percentage overlap between topic words and the most frequent lemmas across the entire dataset. This provides another lens through which to evaluate how well the topic model captures the main themes of the documents.

As a general trend the topic intersection ratio was relatively high from 1949 to 1975, ranging around 50-60%. However, after 1975, the ratio dropped by about 10-15% and remained relatively stable at this lower level (see Figure 10).

Outliers in Topic Intersection Ratio:

- **1951-1955 and 1976-1980:** These time spans show the expected issues, with abnormally low ratios matching the low Jaccard similarity scores.
- **2006-2010:** Despite being a more recent period, this era shows an unexpectedly high intersection ratio, suggesting that topics may have been more consistently represented in this timeframe.

- **2021-2025:** This period has fewer documents than others, which likely explains its lower intersection ratio, but it still aligns with the overall trend of declining topic intersection.

7. Conclusion

7.1. Answer to the Research Question based on the Analysis

Hypothesis: "Topic modeling using NLTK preprocessing has lower performance on older texts than on newer texts."

Based on the analysis of Jaccard similarity and the trend in topic intersection ratios, the results indicate that topic modeling actually performs better on older texts. From 1949-1950 up to the 1970s, there is a clear trend of higher Jaccard similarity values and a more favorable topic intersection ratio, suggesting that the model aligns more closely with the core topics in older documents. After 1971, there is a slight decline in performance, as indicated by the leftward shift in the normal distribution of Jaccard similarities and the drop in intersection ratios. However, this decline remains relatively stable over time.

Therefore, the data effectively falsifies the original hypothesis, demonstrating that topic modeling performs better on older documents. This conclusion is consistently supported by both the Jaccard similarity values and the topic intersection ratio data, indicating a more robust alignment with older texts.

7.2. Implications of the Findings

1. **Effectiveness on Historical Texts:** The better performance of topic modeling on older documents suggests that historical texts use more consistent or structured language, making topic identification easier. This highlights the strength of topic modeling for analyzing well-defined themes in historical data.
2. **Challenges with Modern Texts:** The decline in performance on newer texts indicates that modern language is more diverse and evolving, making it harder for models to identify clear topics. This suggests that contemporary texts may require more advanced or hybrid topic modeling techniques.
3. **Utility in Political and Historical Research:** Topic modeling proves valuable for historical political documents, but the lower alignment in newer texts suggests researchers should consider supplementary methods for more recent material.
4. **Impact on NLP Research:** This project shows that while NLTK-based topic modeling works well for older texts, modern texts might benefit from integrating more advanced NLP methods, emphasizing the need for ongoing model adaptation.

In summary, these findings underscore the importance of considering the age of the text when applying topic modeling techniques, as well as the need to adapt or supplement these methods for newer documents.

7.3. Future Work

While the project provides valuable insights into the performance of topic modeling on older versus newer texts, there are several areas for future exploration:

1. **Fine-tuning the LDA model**: Experimenting with different LDA parameters, such as the number of topics or the learning rate, could help improve performance on older texts.
2. **Analyzing Specific Time Periods in More Detail**: Given the observed outliers, a more in-depth analysis of specific decades—especially 1951-1955 and 1976-1980—could reveal why topic modeling struggled during these periods. This might involve investigating changes in language use, external events, or anomalies in the dataset.
3. **Applying to Different Domains**: Extending this analysis to datasets from different domains, such as legal or medical texts, would be valuable to see if the trends observed in this project hold true across other fields.
4. **Alternative Modeling Approaches**: Exploring other topic modeling techniques, such as Non-Negative Matrix Factorization (NMF) or neural-based models like BERTopic, could provide alternative ways to analyze older documents more effectively.
5. **Broader Dataset**: Expanding the dataset to include documents from additional time periods or other types of historical texts could help validate the findings and provide a more comprehensive view of how NLP models perform on older texts.
6. **Alternative Preprocessing Pipelines:** Using other modern NLP libraries, such as SpaCy or Stanza, which have dvanced language models and tokenization methods that may handle the peculiarities of older texts differently. A comparison between the results of using NLTK, SpaCy, and Stanza could offer further insights into how preprocessing choices influence topic modeling outcomes.

7.4. Challenges

1. **Evaluation and Interpretation of Results:**
    - Measuring the effectiveness of topic modeling using Jaccard similarity posed its own challenges. While Jaccard similarity provides a quantitative measure of overlap, it does not fully capture the quality or relevance of topics. Some documents showed very low overlap between the most frequent lemmas and the topic words, making it difficult to draw definitive conclusions about the model's effectiveness for those documents.
    - The intersection percentage between lemmas and topic words was introduced as an additional measure, but even this metric had limitations, especially for documents with very specific jargon or procedural language.
2. **Coherence Score:**

- The project initially planned to use coherence scores as an additional evaluation metric. However, due to the low term probabilities in the LDA output, coherence scores could not be reliably calculated. This might stem from the smaller dataset paired with a high variation of lemmas. Future iterations of this project could experiment with different LDA implementations or use a coherence score metric more suited to sparse probability distributions.

2. **Outliers and Data Gaps:**
   - Certain time periods, such as 1951-1955 and 1976-1980, showed inconsistencies or issues with topic modeling and similarity calculations. This required extra steps to identify, analyze, and exclude these outliers from parts of the analysis.
3. **Scaling the Project:**
   - As the size of the dataset grew, particularly when processing multiple decades of documents, issues related to performance and data handling arose. The computational cost of running the LDA model on large datasets, combined with the need to store and visualize the results, became a bottleneck. While this was managed by splitting the dataset and saving intermediate results, it highlighted the need for more efficient processing and storage solutions, especially for larger projects.
4. **Hyperparameter Tuning:**
   - Adjusting hyperparameters like the number of topics, topn, alpha, and eta required extensive trial and error to achieve coherent topics. While the final values chosen were effective, they may not be optimal for all time periods. Future work could explore automated hyperparameter optimization techniques to streamline this process.

7.5. Final Thoughts

In conclusion, this project provided valuable insights into the effectiveness of topic modeling on political texts across different time periods. Contrary to the initial hypothesis, the analysis revealed that topic modeling performed better on older texts, indicating that the model captures themes more effectively from earlier decades. This suggests that historical documents might exhibit clearer or more consistent language patterns, making them more suitable for topic extraction.

8. Sources

G. Abrami, M. Bagci, L. Hammerla, and A. Mehler, “German Parliamentary Corpus (GerParCor),” in Proceedings of the Language Resources and Evaluation Conference, Marseille, France, 2022, pp. 1900-1906.

G. Abrami, M. Bagci and A. Mehler, “German Parliamentary Corpus (GerParCor) Reloaded,” in Proceedings of the 2024 Joint International Conference on Computational Linguistics, (LREC-COLING 2024), Torino, Italy, 2024, pp. 7707-7716.

K. Millie, „Natural Language Processing in Python for Text Analysis“ 2024.