Decoding Legislative Discourse: Transformer-Based Topic Modeling of U.S. Congressional Hearings

A Comparative Analysis of Standard and Zero-Shot BERTopic

Supervisor(s): Stephanie Tan[1], Edgar Gírones[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

June 22, 2025

Name of the student: Rebecca Andrei

Final project course: CSE3000 Research Project

Thesis committee: Stephanie Tan, Edgar Gírones, Odette Scharenborg

Abstract

Congressional hearings are at the center of legislation, yet their analysis is hindered by the volume and complexity of the transcripts. While recent advances in Natural Language Processing (NLP) have enabled political discourse analysis using automated tools, conventional topic modeling methods often struggle to produce semantically coherent topics due to their reliance on context-free word frequencies. This paper evaluates the performance of a new transformer-based topic modeling technique, focusing on its application to policy discussions through a detailed case study. Two variants of BERTopic are considered: (1) a parameter-tuned model and (2) a zero-shot variant, evaluated on U.S. congressional hearing transcripts from 2021 to 2024. The results demonstrate that the zero-shot version achieves competitive coherence with increased interpretability and stability, making it a useful resource for policymakers and researchers alike. This paper establishes a foundational methodological framework for automated legislative text analysis. It also outlines the trade-offs between unsupervised and semi-supervised topic modeling in political usage.

1 Introduction

Congressional hearings are important legislative policymaking tools, where stakeholders introduce evidence, debate policy implications, and shape legislation through structured deliberation (Congressional Research Service, 2010). However, analyzing these hearings poses distinct difficulties due to the complexity of the transcripts, which combine heterogeneous structures (e.g., testimonies, Q&A exchanges), domain-specific terminology, and strategic framing designed to advance political agendas. Additionally, the sheer volume of transcripts makes manual analysis labor-intensive, subjective, and impractical for large-scale or real-time policymaking.

Although recent developments in Natural Language Processing (NLP) have enabled patterns in such debates to be extracted and interpreted automatically through techniques like argument mining (Ruiz-Dolz et al., 2022) and stance analysis (Le et al., 2016), topic modeling, a foundational method for uncovering latent themes in text, remains underutilized in legislative settings. Previous studies have neglected to provide a systematic evaluation of whether the generated topics are semantically coherent, policy-relevant, or sensitive to variations in input data. This gap limits their utility for systematic analysis of legislative debates and shows the need for more sophisticated methods tailored to legislative texts.

This paper addresses these shortcomings by evaluating the performance of transformer-based approaches for topic modeling through a case study. This work aims to quantify the extent to which they are able to produce coherent topics in U.S. congressional hearings. In particular, we compare two variants of the BERTopic model (Grootendorst, 2022): (1) a parameter-optimized version designed to maximize coherence and distinctness, and (2) a hybrid zero-shot variant that combines unsupervised clustering with pre-defined labels to direct topic assignment. The objective is to quantify the effectiveness of these models through specific metrics and methodologies across four key areas: topic coherence, semantic distinctness, computational stability, and interoperability.

To investigate the performance of transformer-based topic modeling for political text analysis, the **primary research question** studied is the following: How does zero-shot BERTopic compare to parameter-optimized BERTopic in extracting coherent, distinct, and policy-relevant topics from congressional hearing transcripts (2021-2024)?

For a thorough analysis, we further consider the sub-questions:

1. What impact do key BERTopic parameters (e.g., minimum topic size, number of topics, embedding model, etc.) have on topic coherence and distinctiveness?
2. What are the differences between the coherence scores and the semantic distinctiveness of topics produced by the two models?
3. How stable are the topics produced by zero-shot and parameter-tuned BERTopic under random corpus resampling?
4. To what extent do the identified topics exhibit interpretable structure, as evaluated through qualitative inspection?

The rest of this paper is structured as follows. Section 2 covers background work on topic modeling. Section 3 describes the dataset, models, and performance metrics used, while Section 4 presents the experimental findings, and Section 5 discusses their significance. Section 6 offers reflections and recommendations for future work, and Section 7 addresses responsible research.

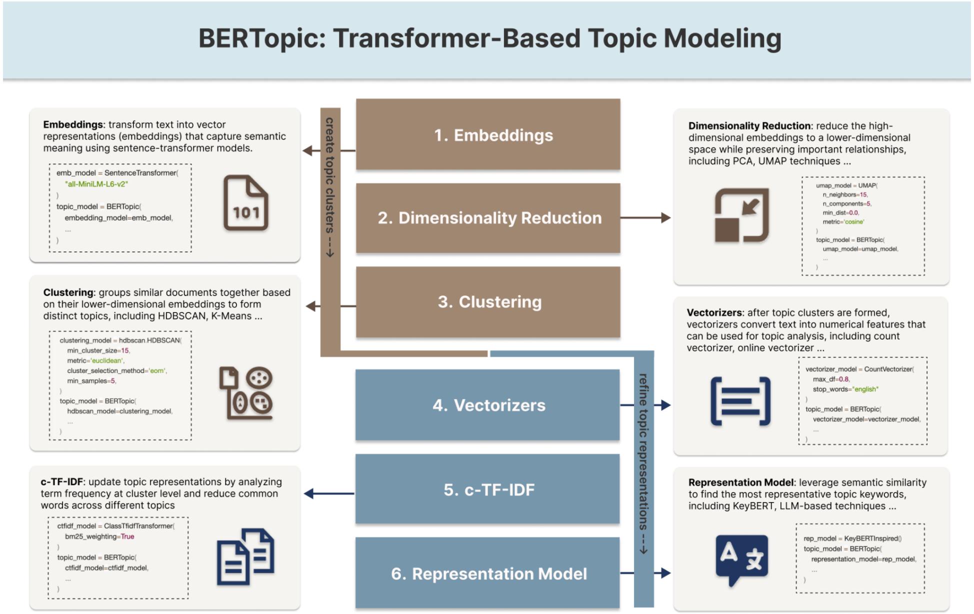
2 Background

2.1 Topic Modeling

Topic modeling is a set of NLP unsupervised machine learning techniques that identify and extract the latent topics of a large corpus of unstructured text documents (Blei, 2012). Classic models include probabilistic topic modeling, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), and dimensional reduction methods, like Non-Negative Matrix Factorization (NMF) (Lee & Seung, 1999). Both rely on word co-occurrences and bag-of-words representation to model topic distributions. These algorithms have been applied in many areas, with political text analysis being the most relevant to this paper (Bagozzi & Berliner, 2018; Greene & Cross, 2017; Quinn et al., 2010).

However, the traditional approaches possess some well-documented deficiencies. Their context-free word frequency application can generate incoherent subjects and often fails to detect embedded meaning in a corpus (Blair et al., 2020). This is particularly true in settings with expert jargon and high semantic subtlety, such as congressional hearings. LDA, for example, may consider "climate change" and "global warming" as distinct topics despite the semantic similarity between them. This indicates that they tend not to be able to identify the different subtleties in policy debates.

2.2 BERTopic


BERTopic: Transformer-Based Topic Modeling
1. Embeddings
2. Dimensionality Reduction
3. Clustering
4. Vectorizers
5. c-TF-IDF
6. Representation Model
create topic clusters --→
refine topic representations --→

Embeddings: transform text into vector representations (embeddings) that capture semantic meaning using sentence-transformer models.

```
emb_model = SentenceTransformer(
    "all-MiniLM-L6-v2"
)
topic_model = BERTopic(
    embedding_model=emb_model,
    ...
)
```

Dimensionality Reduction: reduce the high-dimensional embeddings to a lower-dimensional space while preserving important relationships, including PCA, UMAP techniques ...

```
umap_model = UMAP(
    n_neighbors=15,
    n_components=5,
    min_dist=0.0,
    metric='cosine'
)
topic_model = BERTopic(
    umap_model=umap_model,
    ...
)
```

Clustering: groups similar documents together based on their lower-dimensional embeddings to form distinct topics, including HDBSCAN, K-Means ...

```
clustering_model = hdbscan.HDBSCAN(
    min_cluster_size=15,
    metric='euclidean',
    cluster_selection_method='eom',
    min_samples=5,
)
topic_model = BERTopic(
    hdbscan_model=clustering_model,
    ...
)
```

Vectorizers: after topic clusters are formed, vectorizers convert text into numerical features that can be used for topic analysis, including count vectorizer, online vectorizer ...

```
vectorizer_model = CountVectorizer(
    max_df=0.8,
    stop_words="english"
)
topic_model = BERTopic(
    vectorizer_model=vectorizer_model,
    ...
)
```

c-TF-IDF: update topic representations by analyzing term frequency at cluster level and reduce common words across different topics

```
ctfidf_model = ClassTfidfTransformer(
    bm25_weighting=True
)
topic_model = BERTopic(
    ctfidf_model=ctfidf_model,
    ...
)
```

Representation Model: leverage semantic similarity to find the most representative topic keywords, including KeyBERT, LLM-based techniques ...

```
rep_model = KeyBERTInspired()
topic_model = BERTopic(
    representation_model=rep_model,
    ...
)
```

Figure 1: BERTopic Phases (Gong, 2025).

The current techniques in this field use contextual embeddings from pre-trained language models such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), which have denser semantic representations. BERTopic (Grootendorst, 2022) operates in six stages, as illustrated in Figure 1: 1) Semantic embedding generation using sentence transformers to transform text into vector embeddings. 2) Dimensionality reduction to reduce high-dimensional embeddings to a lower-dimensional space. 3) Density-based clustering to group similar documents together. 4) Document-term matrix construction via vectorization to convert text into numerical features for analysis. 5) Term importance scoring on a cluster level via class-based TF-IDF to reduce common words across different topics. 6) Semantic fine-tuning of topic representations. In doing so, it consistently performs better than both LDA and NMF in tasks requiring semantic understanding (Egger & Yu, 2022), meaning it performs better in context-heavy tasks such as political analysis.

Another innovation is topic modeling with zero-shot learning integration (Palatucci et al., 2009). Zero-shot BERTopic, and models like ZeroBERTo and BERTrend that use it (Alcoforado et al., 2022; Boutaleb et al., 2024), extend BERTopic by adding pre-defined topic labels. These models classify documents into possible topics based on embedding similarity instead of depending entirely on unsupervised clustering.

Such an integrated system is especially useful in policy analysis, as it makes it possible for researchers to both monitor known themes and also discover new ones simultaneously in complex legislative contexts.

Despite the breakthroughs in topic modeling techniques, there is a lack of systematic studies comparing zero-shot topic models with fine-tuned unsupervised baselines specifically in the legislative text domain. Most of the research performed so far relies on established metrics or detailed case studies with less focus on robustness, explainability, or policy relevance. They act as a complement to this study, which comparatively evaluates zero-shot and fine-tuned BERTopic on 2021-2024 U.S. congressional hearing transcripts against multiple quantitative and qualitative metrics.

3 Methodology

3.1 Data Collection

The data for this study consists of publicly accessible U.S. congressional hearing transcripts from the 117th and 118th Congresses, between January 2021 and November 2024. The hearings were manually retrieved from the official website *congress.gov* [1]. Only hearings dealing with explicit policy deliberations (health, environment, technology) were used, excluding mark-ups, ceremonial, and procedural meetings. This approach ensures thematic relevance and sufficient diversity to test model generalizability across policy domains. Furthermore, to ensure topical diversity, the hearings were drawn from an active House committee, namely the **House Committee on Foreign Affairs**. The Foreign Affairs corpus is semantically constrained, dealing with a comparatively smaller number of topics; however, these topics vary significantly (e.g., human trafficking, global security, etc.), making it topically distinct.

Transcripts were preprocessed to eliminate non-content items like bracketed text and special characters, along with procedural metadata (timestamps, speaker references, honorifics, etc.) and punctuation. The text was then tokenized to single words, which were further lowercased and filtered using a list of common English stop-words, retrieved from the Natural Language Toolkit platform, `nltk`[2]. An exception to this was words within a predefined list of domain-terms such as "security", "trade", "foreign", and so on, found in Appendix A. Also, tokens were only retained if they were more than three characters long and had no non-alphabetic characters. This was done to ensure that the final corpus had semantically significant content after noise removal.

3.2 Models

3.2.1 Parameter-Optimized BERTopic

The methodology used in this research to construct an optimally parameter-tuned BERTopic consists of four main phases: (1) data preprocessing, explained in Section 3.1, (2) chunk and embedding model selection, (3) parameter tuning, and (4) model evaluation, outlined in Section 3.3. Each step was chosen to be computationally efficient and maximize topic coherence.

Chunk Size and Embedding Model

Following the preprocessing, the next phase consists of splitting each document into smaller chunks of approximately equal length, with a minimum limit of 30 words per chunk. This step allowed even long documents to be uniformly represented during topic modeling without over-weighting the results for verbose transcripts. As a result, the output dataset is both noise-reduced and semantically rich.

The optimal document chunk size was determined through empirical testing. Five candidate chunk sizes (150, 250, 300, 400, and 500 tokens) were evaluated across five embedding models: `all-MiniLM-L6-v2`, `all-mpnet-base-v2`, `paraphrase-MiniLM-L6-v2`, `sentence-t5-base` (Reimers & Gurevych, 2019), and `intfloat/e5-base-v2` (L. Wang et al., 2022). This preliminary testing was selected as chunk size affects both the quality of embeddings and the resulting topic coherence. Smaller chunks risk losing contextual information, while larger chunks risk introducing noise by combining multiple topics in one document.

For determining which embedding model to use, the process involved extensive experimentation with sentence transformers. Five diverse HuggingFace embedding models were chosen to compare different architectural approaches and their impact on topic quality: MPNet (Song et al., 2020), MiniLM (W. Wang et al., 2020), T5 (Raffel et al., 2023), and E5 (L. Wang et al., 2022). These transformers were selected based on their performance in semantic textual similarity benchmarks and their varying capacity

[1] See https://www.congress.gov/house-hearing-transcripts for official hearing transcript access.
[2] https://www.nltk.org/

to capture document-level semantics. The embeddings were precomputed and cached for each chunk size to ensure reproducibility and efficiency during parameter tuning. This allowed for a fair comparison across different parameter settings while being computationally feasible.

Parameter Tuning

The parameter optimization phase involves a two-stage tuning process. First, parameters are tuned individually to identify promising ranges of values and shrink the possible parameter space before a more exhaustive tuning. Second, this screening is followed by a grid search that exhaustively compares combinations of eight impactful parameters: chunk size, embedding model, minimum topic size, number of topics, UMAP minimum distance, UMAP number of neighbors, UMAP minimum distance, HDBSCAN minimum cluster size, and HDBSCAN minimum number of samples. This large search space was designed based on BERTopic's official documentation [3], along with the individual tuning experiments that found these parameters most affected topic quality. The UMAP parameters (McInnes et al., 2020) control the dimensionality reduction step, affecting the cluster separation, and the HDBSCAN parameters (Campello et al., 2015) control the density-based clustering that defines BERTopic's topic identification.

The BERTopic implementation was customized in several ways to improve topic quality. A stopword list was created to remove general but uninformative words specific to the subject domain of congressional hearings, such as "chairman", "adjourned", "subcommittee", etc., found in Appendix A. The vectorizer model was initialized with an n-gram range of (1, 3) to capture relevant multi-word phrases and uses the stopword list to exclude overly common terms. Additionally, the representation model used KeyBERTInspired to generate more interpretable topic labels through transformer-based keyword extraction by measuring cosine similarity to determine which words are most semantically aligned with the corpus, thereby improving their representativeness for each topic. Topic filtering removed topics with fewer than three distinct top words so that only semantically coherent topics were kept for analysis.

3.2.2 Zero-shot BERTopic Implementation

The zero-shot BERTopic implementation enhances the parameter-optimized BERTopic pipeline by incorporating candidate topic labels from an initial base model run, which facilitates semi-supervised topic labeling and improves interpretability. It uses the same chunk size and hyperparameter setup as the regular BERTopic implementation for ease of comparison. The implementation process consists of three steps: (1) candidate topics generation, (2) setup of the zero-shot model, and (3) the fine-tuning and evaluation of the topics, which is the same as the basic BERTopic implementation. The method is intended to use predefined candidate topics and retain the flexibility to also learn new topics. As such, it is suitable for cases where partial prior knowledge of the topics exists, but where exhaustive pre-labeling is impractical.

The first step involves generating the candidate topics by initializing a parameter-optimized default BERTopic model. This base model processes the chunked input documents to find an initial set of topics. These are then filtered to exclude outliers and normalized to avoid noise. For each valid topic, the top four most frequent words are joined using underscores so that multi-word phrases are preserved (e.g., "health_care_reform_bill"), resulting in a candidate label. These labels are deduplicated to form the final list of candidate topics, which are the target classes for zero-shot classification.

The second step involves creating the zero-shot model configuration; candidate topics are added to a BERTopic instance that uses the optimized parameters previously found. The zero-shot configuration uses the same embedding model, vectorizer, representation model, UMAP dimensionality reduction parameters, and HDBSCAN clustering components as the parameter-optimized BERTopic, but adds zero-shot assignment to the unsupervised topic discovery. The most prominent differences are the `zeroshot_topic_list` parameter, which is used to append the candidate topics list, and the `zeroshot_min_similarity` parameter, which is used to set the minimum cosine similarity threshold when assigning a document embedding to a candidate topic centroid. This threshold is tuned to obtain the most promising value, with higher values offering more precision at the cost of coverage. The final phase is identical to the baseline BERTopic's post-processing step.

This approach is more interpretable than purely unsupervised methods but remains adaptable to unseen topics. The use of precomputed embeddings and the ability to configure a similarity threshold guarantees computational efficiency, which allows it to scale to the large document collections typical of congressional hearing transcripts or other domain-specific corpora.

[3] See this page for documentation related to hyperparameter tuning.

3.3 Evaluation Metrics

The evaluation employs a range of complementary metrics, including quantitative measures like topic coherence and semantic distinctiveness, as well as qualitative assessments through manual inspection, to provide a comprehensive overview of the model's performance. A dual approach was selected so that the model produces mathematically sound results and interpretable topic clusters for policy analysis.

3.3.1 Quantitative Metrics

Three dimensions are used to empirically calculate the model's performance: topic coherence, semantic distinctiveness, and computational stability.

Topic coherence is quantified in terms of the C_v and U_{Mass} scores used in the `gensim`[4] library. The resulting scores of both metrics are normalized to the [0,1] range for ease of interpretation in later comparisons. The C_v coherence score measures the normalized pointwise mutual information (NPMI) of the top-n words in each topic, and, based on the co-occurrence patterns of the topic words across the corpus, indicates how semantically connected they are. The metric was selected based on the findings of Röder et al., 2015, which showed it correlates well with human assessments of topic quality compared to other measures. The U_{Mass} coherence score (Mimno et al., 2011), also based on word co-occurrence, calculates how often two words, w_i and w_j, appear together in the corpus using a log-conditional probability formula. The final score aggregates pairwise values across all top-n topic words:

$$U_{\text{Mass}} = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} U_{\text{Mass}}(w_i, w_j) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \quad (1)$$

where:

- $P(w_i, w_j)$ is the probability of words co-occurring within a sliding window
- $P(w_j)$ is the marginal probability of word w_j
- ϵ is a small smoothing constant

Semantic Distinctiveness is computed via pairwise cosine distance of topic embeddings, derived from the c-TF-IDF representations of the model. It measures how separated topics are in the semantic space, with a lower average similarity score indicating that the topics are more clearly differentiated. The implementation makes use of `scikit-learn`'s [5] `cosine_similarity` method applied to the topic embedding matrix, excluding diagonal elements (self-similarity) to extract inter-topic relationships. This metric was selected because it directly measures to what extent meaningfully different pieces of the corpus are captured in distinct topics, which is an important aspect of good topic modeling.

Computational Stability is assessed through a resampling test. The data is split randomly a hundred times, topic modeling is run for every split, and topic stability is measured using the Adjusted Rand Index (for cluster assignment consistency) and top-n term overlap (for topic label stability). This approach assessed the model's robustness against variations in input data, which is a relevant concern in real-world applications where data characteristics may evolve over time.

3.3.2 Qualitative Metrics

The qualitative evaluation assesses interpretability through manual inspection of topic word lists and multidimensional visualizations. The visual assessment includes examinations of two main components.

First, **inter-topic distance maps** project topic embeddings into a two-dimensional space through UMAP dimensionality reduction, showing the global semantic relations between topics through spatial proximity and cluster formation patterns. Second, a **bar chart** illustrates the number of documents belonging to each cluster. This allows for an inspection of term relevance and semantic cohesion within each topic cluster.

[4] https://pypi.org/project/gensim/

[5] https://scikit-learn.org/stable/index.html

4 Results

4.1 Experimental Setup

4.1.1 Basic BERTopic

For the basic BERTopic model, the implementation begins with an extensive chunk size and embedding model optimization phase. A full accounting of the results may be found in Appendix B. Using a set of heuristically determined default parameters in Appendix C, five candidate chunk sizes (150, 250, 300, 400, and 500 tokens) are evaluated against five embedding models: `all-MiniLM-L6-v2`, `all-mpnet-base-v2`, `paraphrase-MiniLM-L6-v2`, `sentence-t5-base`, and `intfloat/e5-base-v2`.

This preliminary analysis used a **composite scoring metric**, calculated by weighting five results: number of meaningful topics (10%), topic coverage (20%), outlier ratio (20%), average topic size (10%), and the combined coherence-distinctiveness score (40%). The balanced metric prevents over-optimization on any single dimension and ensures the resulting topics are usable. Following this evaluation, the three best-performing embedding models and chunk sizes were chosen for further parameter tuning.

The **number of meaningful topics** metric is a straightforward count of the number of topics found by the model that are of a pre-defined quality, i.e., not outliers (`topic_id` $\neq -1$) and have a minimum of three distinct normalized words after preprocessing. The count is normalized, divided by the maximum observed value across configurations. This metric rewards models that are able to extract a reasonable number of interpretable topics without over-fragmenting the data. The **topic coverage** measures the number of documents assigned to non-outlier topics divided by the total number of documents. The metric punishes models with a high number of unassigned documents (high outlier ratio) and is normalized by rescaling the scores to the [0,1] range by dividing them by the maximum observed coverage. Conversely, the **outlier ratio** computes the percentage of documents that fall under no topic (outliers) as the number of documents in topic -1 divided by the total number of documents. The code inverses the percentage (1 − `outlier_ratio`) before normalizing so that it corresponds with the composite score's preference for higher values signifying better performance. Inverse scaling has the effect that models with fewer outliers have a larger positive impact on the composite score, preferring general topic assignment. The **average topic size** metric calculates the mean number of documents assigned to each non-outlier topic, and is normalized by dividing by the maximum observed topic size across all configurations.

Finally, the **combined score** is calculated as a weighted average of a normalized topic coherence score and a semantic distinctiveness score, where the factor $\alpha = 0.8$ controls the weight of the coherence; both are described in Section 3.3. On the one hand, the coherence component averages two metrics, namely C_v and U_{Mass}, which are normalized. The C_v score is used directly, while U_{Mass}, which is usually negative, is normalized using the formula $1 + \frac{U_{\text{MASS}}}{14}$ to map it to a [0,1] range. On the other hand, the distinctiveness score is computed by taking one minus the average cosine similarity between topic embeddings, such that lower average similarity (i.e., higher distinctiveness) results in a higher score. Consequently, the total combined score is calculated by multiplying the normalized coherence score by α and the distinctiveness score by $1 - \alpha$, and then adding them together. This returns a single scalar with higher values indicating better topic model quality according to the specified coherence-distinctiveness trade-off.

The parameter tuning phase used a two-stage approach to optimize the high-dimensional hyperparameter space. First, the hyperparameters that impact the model's performance the most were identified using BERTopic's official documentation[6]. For this paper, they were: the minimum topic size, the number of topics, the range of the n-grams used by BERTopic, along with the UMAP parameters: the number of neighbors, the number of components, the minimum distance between clusters, and the HDBSCAN parameters: the minimum size of the clusters and the minimum number of samples. Using the default values in Appendix C for the rest of the hyperparameters, each of the identified parameters was tested individually to identify a promising range. Second, along with the selected embedding models and chunk sizes, these ranges informed an exhaustive grid search that checked each unique parameter combination. The search space was constrained to values that were found to be stable in preliminary testing but still maintained meaningful variations in model behavior. The configurations were evaluated using the same combined coherence-distinctiveness metric as in the chunk analysis, with precomputed and cached embeddings for each chunk size to ensure fair comparisons.

After the grid search, several value combinations had the same highest combined score. As such, a qualitative evaluation of the possible results was used to determine the optimal hyperparameter values.

[6] See this page for documentation related to hyperparameter tuning.

4.1.2 Zero-shot BERTopic

The zero-shot BERTopic implementation followed a pipeline specific for its supervised learning characteristics. Candidate topics were extracted directly by a basic parameter-optimized BERTopic model and post-processed according to the procedure described in 3.2.2. The model was then configured with the chunk size and parameter values as the basic BERTopic to isolate and focus on architectural differences.

Finally, the only additional parameters were represented by the candidate topic list and the minimum similarity threshold for document assignment to a cluster. The minimum similarity threshold was individually tuned to determine the most promising value, quantified by the **combined coherence-distinctiveness score** as described in 4.1.1. The same evaluation criteria as for the basic BERTopic model (Section 3.3) were used for consistency.

4.2 Parameter Tuning

A complete accounting of the parameter tuning results may be found in Appendix B.

4.2.1 Basic BERTopic

First, the implementation examined possible chunk sizes and embedding models using the metrics explained in 4.1.1. This preliminary analysis revealed that smaller chunks (150 and 250 tokens) often fractured longer policy discussions regarding a single topic, while larger chunks, that is, more than 500 tokens, introduced noise by combining multiple topics (Figure 4).

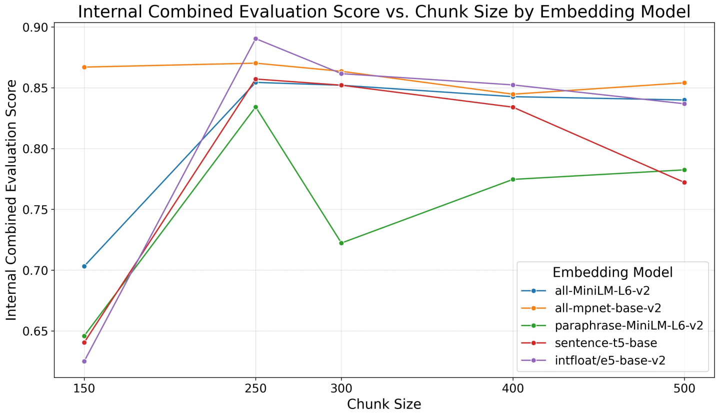


Figure 2: Combined Evaluation Score vs. Chunk Size by Embedding Model

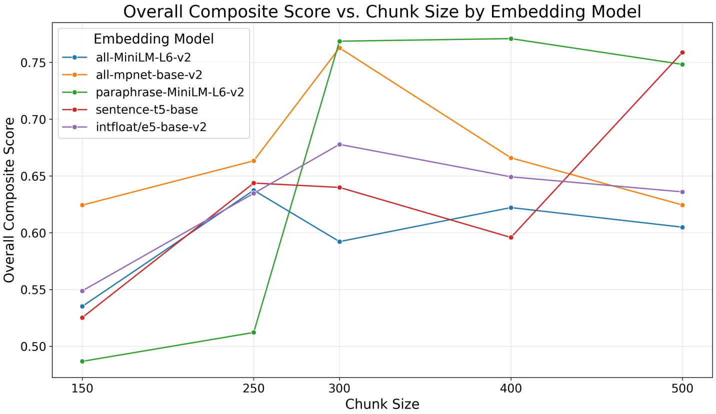


Figure 3: Overall Composite Score vs. Chunk Size by Embedding Model

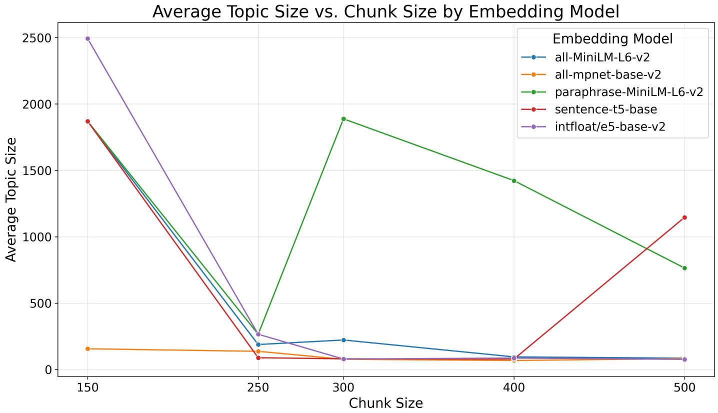


Figure 4: Average Topic Size vs. Chunk Size by Embedding Model

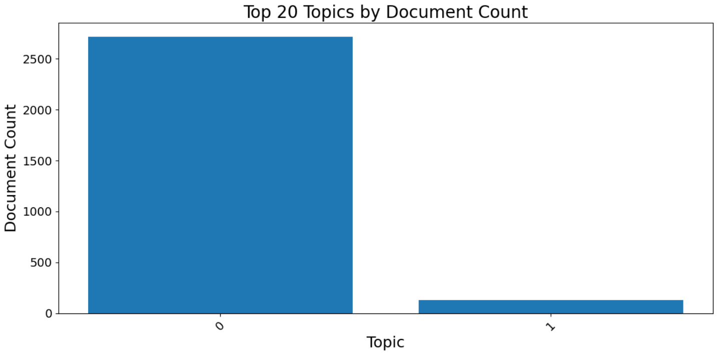


Figure 5: Topic Distribution for `paraphrase-MiniLM-L6-v2` model and 400 tokens

The evaluation showed that the `paraphrase-MiniLM-L6-v2` model with a chunk size of 400 yielded the highest overall composite score in Figure 3. However, upon closer examination, it was found that the model is an anomaly, generating only two topic clusters; one with 2,718 documents and another with

127 documents, along with an empty outlier cluster (-1) (Figure 5). Due to this lack of meaningful topic distribution, the model should not be considered further for the parameter tuning. As shown in Figures 2 and 3, the `all-mpnet-base-v2`, `infloat/e5-base-v2`, and `all-miniLM-L6-v2` models demonstrated the best scores for mid-range sizes (300, 400, and 500 tokens per chunk). As such, those will be the three models chosen for the grid tuning, along with the mid-range sizes (300, 400, and 500).

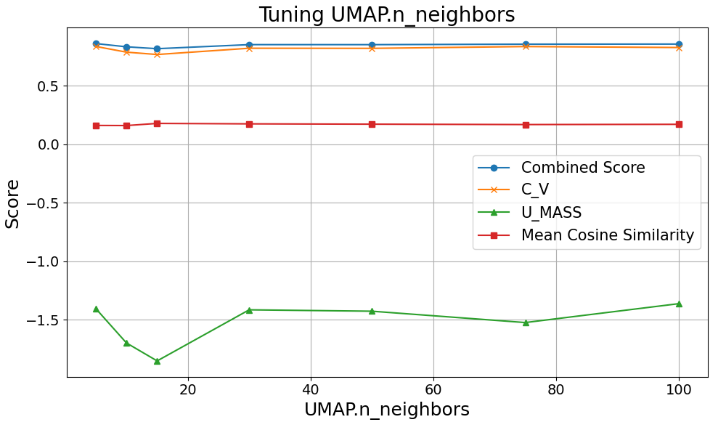


Figure 6: UMAP `n_neighbors` tuning for 300 tokens

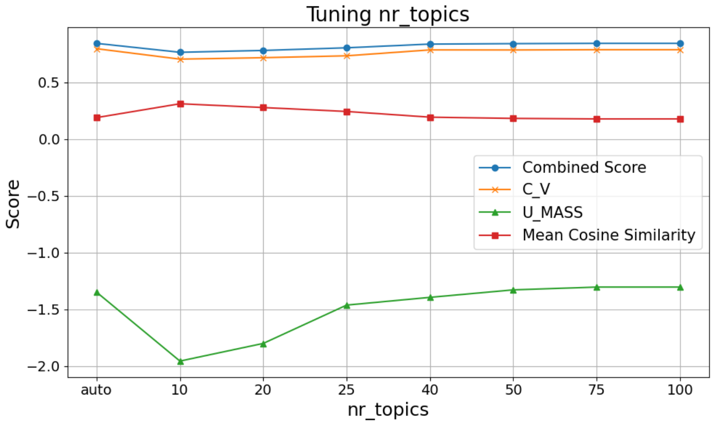


Figure 7: `nr_topics` tuning for 400 tokens

For each chunk size, individual tuning was run on the most important hyperparameters to determine a promising value range, using the default parameter values for the others. Only the three values with the most consistently high combined scores were selected based on plots such as in Figures 6 and 7, found in Appendix B. To summarize the results, `min_topic_size` has been reduced to [5, 10, 15], `nr_topics` to ['auto', 75], `min_dist` to [0.1, 0.3], `n_neighbors` to [10, 15], `min_cluster_size` to [10, 15], and `min_samples` to [5, 10].

	Embed.	**Min Topic Size**	**Nr topics**	**Min Dist.**	**Nr Neighbors**	**Min Cluster Size**	**Min Samples**	**CV**	**UMass**	**Mean Cosine Sim.**	**Comb. Score**
300	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	15	5	0.8673	-1.2638	0.1497	0.8808
300	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	5	0.8528	-1.5047	0.1286	0.8724
300	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	15	10	0.8475	-1.5985	0.1424	0.8648
500	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	5	0.8238	-1.2072	0.1560	0.8638
400	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	10	0.8230	-1.1674	0.1620	0.8634

Table 1: Grid Search Results for Topic Modeling Parameters

Following exhaustive tuning, several consistent trends emerge from the data. First, the combination of the model `intfloat/e5-base-v2` and the setting 'auto' for `nr_topics` yields the strongest performance, with the 96 highest scoring combinations using both. Second, a `chunk_size` of 300, along with `min_samples` set to 5, also scores better than their alternatives. Finally, `min_topic_size`, `min_dist`, and `nr_neighbors` appear to have minimal effect on any metric when this model is used, suggesting they have a negligible contribution for `intfloat/e5-base-v2`.

The final step involves a quantitative analysis of the top-scoring configurations so that the yielded results are useful. Following the conclusions drawn from Table 1, parameters `model_embedding` = `intfloat/e5-base-v2`, `nr_topics` = 'auto', `chunk_size` = 300, `min_cluster_size` = 15, `min_samples` = 5 are fixed. The only parameters left to verify are `min_topic_size`, `min_dist`, and `n_neighbors`.

The first observation to make is that using `min_dist` = 0.1 results in one or two "bigger" clusters, and many significantly smaller ones (Fig. 8). As this effect persists across variations of the `n_neighbors` (Fig. 9) and `min_topic_size` (Fig. 10) parameters, but disappears when `min_dist` is increased to 0.3 (Fig. 11), `min_dist` = 0.1 must cause this result. The distribution suggests that this configuration creates an overly dense representation where most of the documents are forced into a few dominant clusters while leaving many outlier documents poorly grouped. As such, `min_dist` = 0.3 will be fixed.

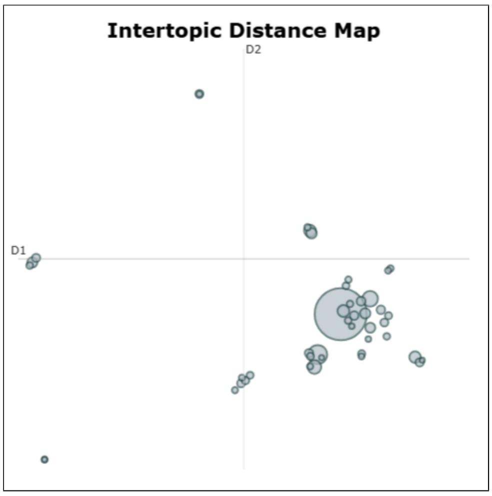


Figure 8: Inter-Topic Distance Map for the 5-0.1-5 configuration

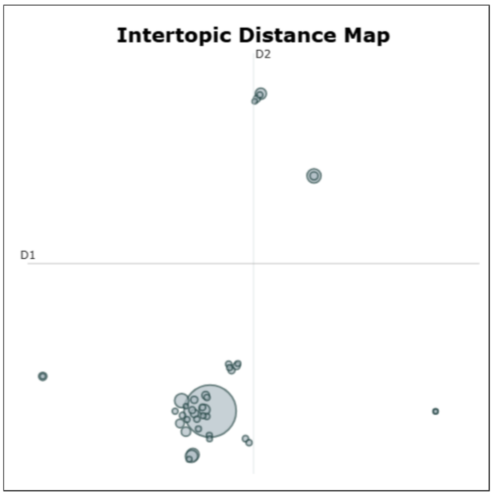


Figure 9: Inter-Topic Distance Map for the 5-0.1-10 configuration

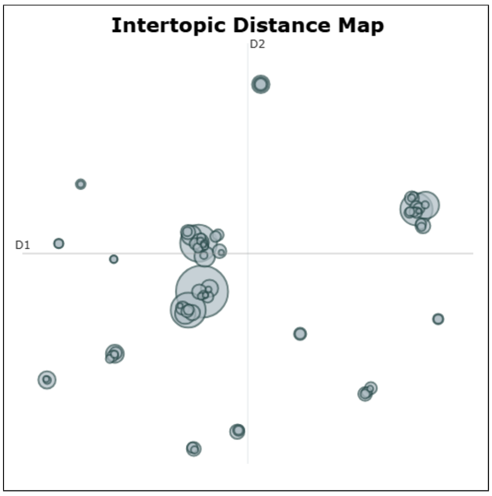


Figure 10: Inter-Topic Distance Map for the 10-0.1-5 configuration

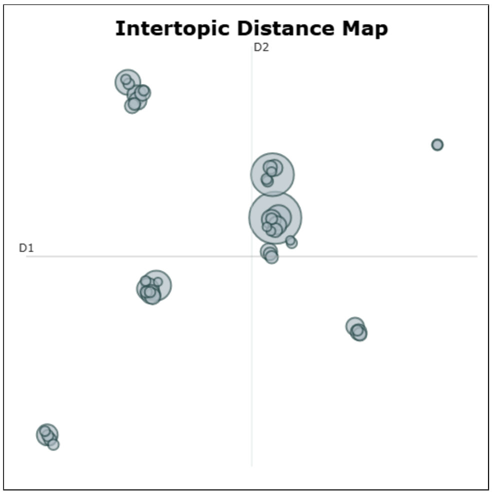


Figure 11: Inter-Topic Distance Map for the 5/10/15-0.3-5 configurations

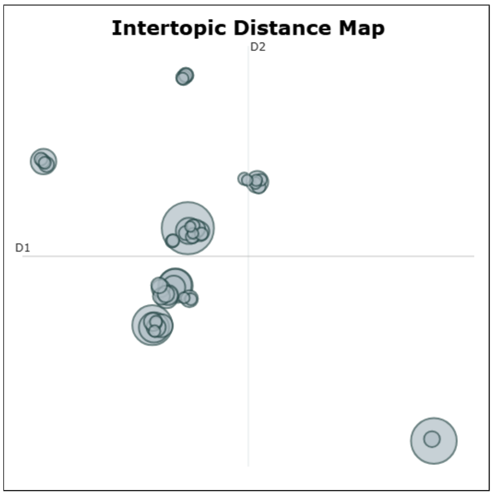


Figure 12: Inter-Topic Distance Map for the 10-0.3-10 configuration

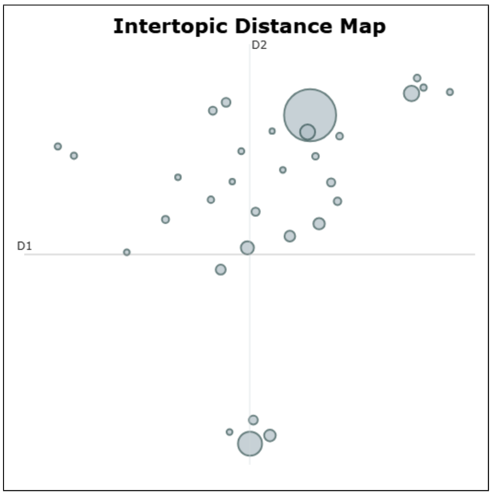


Figure 13: Inter-Topic Distance Map for the 10-0.3-15 configuration

Secondly, the analysis showed that variations in `min_topic_size`, tested at 5, 10, and 15, have almost no impact on the overall topic distribution; they produce identical inter-topic maps for all settings (Fig. 11). Such consistency prevails regardless of any `n_neighbors` parameter value; this implies that in this data, for these set values, topic separation and quality are more controlled by dimensionality reduction settings than cluster size control. However, `min_topic_size` might affect outlier rates and the granularity of small topics that are invisible at the inter-topic level. Let `min_topic_size` be set to 10 as a middle ground between sensitivity (lower values could capture more nuanced topics but risk fragmentation) and robustness (higher values promote stability but may overlook meaningful smaller clusters).

Lastly, the analysis revealed that as `n_neighbors` increases, the clusters become fewer and sparser, with one dominant cluster emerging (Fig. 13). This outcome led to the exclusion of `n_neighbors` = 15 because of its poor preservation of topic diversity. To differentiate between the candidate values of 5 and 10 (Fig. 11, 12), examining Fig. 14 and 15 yields two observations. First, `n_neighbors` = 5 produces fewer outlier documents overall, demonstrating better topic coverage. Second, there is a smaller size disparity between the most populous cluster and the -1 cluster, suggesting more balanced topic assignments. For these reasons, `n_neighbors` = 5 was chosen as the optimal parameter value. After finalizing the parameter tuning, the resulting configuration is detailed in Appendix C.

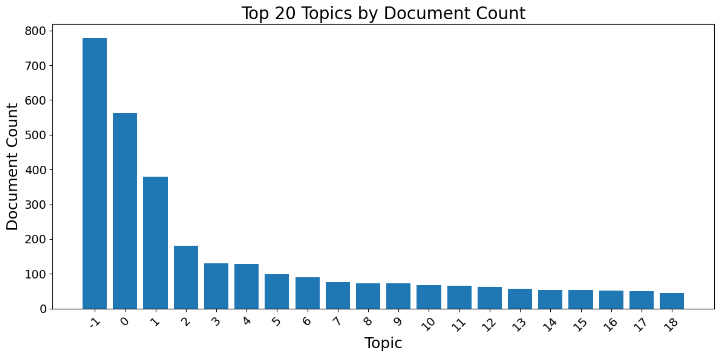


Figure 14: Document count per topic for the 10-0.3-5 configuration

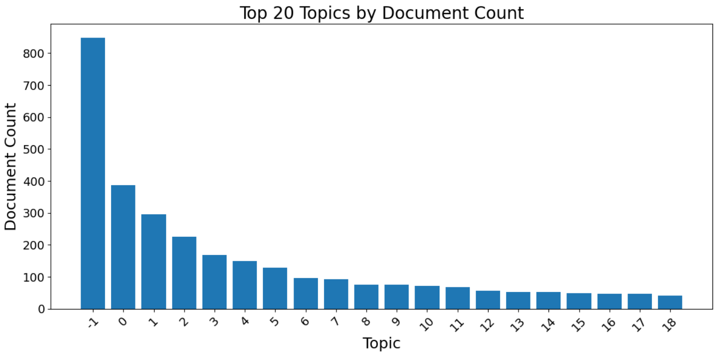


Figure 15: Document count per topic for the 10-0.3-10 configuration

4.2.2 Zero-shot BERTopic

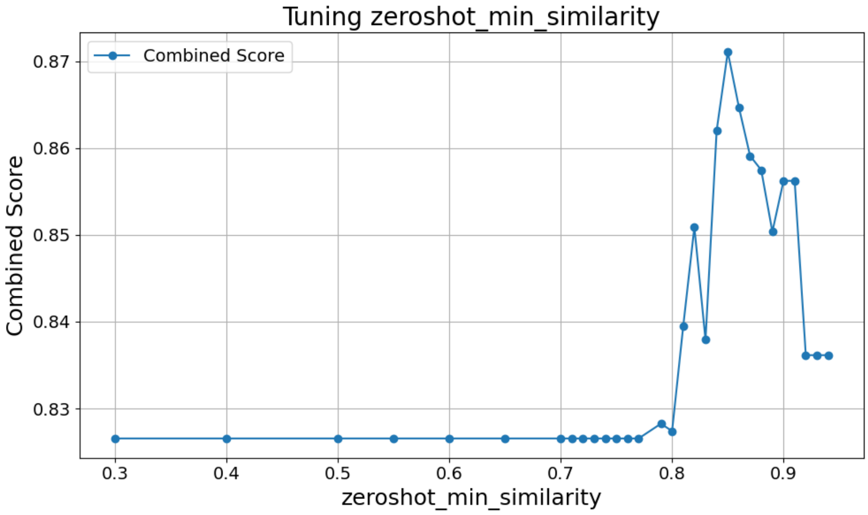


Figure 16: `zeroshot_min_similarity` tuning

The tuning phase for zero-shot BERTopic consists of optimizing the `zeroshot_min_similarity` parameter for the combined coherence-distinctiveness score (Section 4.1.1). This parameter controls the cosine similarity threshold necessary for a document embedding to be assigned to a topic label, with higher values offering more precision, but less coverage.

As can be observed from Figure 16, the maximum combined score of 0.8711 is achieved for 0.85. Using this value results in 98 topics, with C_v = 0.8404, $U_{\text{Mass}} = -1.5168$, and the mean cosine similarity = 0.1088.

4.3 Zero-shot Performance

Model	Number of Topics	C_v	U_{Mass}	Mean Cosine Similarity	Combined Score
Basic	46	0.8673	-1.2638	0.1497	0.8808
Zero-shot	98	0.8404	-1.5168	0.1088	0.8711

Table 2: Comparison of Results between Basic and Zero-shot BERTopic

The zero-shot BERTopic implementation demonstrated competitive performance in extracting coherent and policy-relevant topics from congressional hearing transcripts, as proven both by quantitative metrics and a qualitative evaluation. The combined coherence-distinctiveness score of the model was 0.8711, only slightly lower than basic BERTopic's 0.8808, consisting of $C_v = 0.8404$, $U_{\text{Mass}} = -1.5168$, indicating high semantic consistency across topics, and an average cosine similarity of 0.1088, showing a very high degree of semantic distinctiveness between the generated topics (Table 2).

The zero-shot model generated 98 topics, compared to the parameter-optimized version's 46, which can be attributed to the increased granularity introduced by predefined candidate labels. The model's similarity threshold of 0.85, used to assign a document to a cluster, eliminated much noise, guaranteeing that only semantically aligned documents were grouped together. The high coherence scores indicate the internal consistency of the topics, with top words being very closely related to the policy themes they were capturing. This is in line with the goal of producing policy-relevant outputs for legislative analysis.

4.4 Stability Analysis

The stability of the base BERTopic model was evaluated through a hundred repeated resampling tests, measuring the consistency of cluster assignments and topic label persistence. The Adjusted Rand Index (ARI) scores indicate a minimal consistency of cluster assignments between resampling iterations, with a mean ARI value of 0.000035 ($\sigma = 0.0042$), demonstrating negligible agreement in document-to-topic mappings. Conversely, the consistency of topic labels was far more robust, calculated by the top-n term overlap. The model achieved a mean overlap score of 0.6895 ($\sigma = 0.00916$), meaning that while document assignments to clusters were highly heterogeneous, the semantic content of the identified topics did not vary significantly across different data subsets. The standard deviation in overlap scores, with a value range between 0.3492 and 0.9021, indicates variation in topic preservation in certain policy domains.

Similarly, as seen in Figure 17, the zero-shot BERTopic model also has low stability of cluster labels, with a mean ARI score of -0.00003 ($\sigma = 0.00479$). However, the semantic content stability of its identified topics is significantly higher, achieving a mean top-n term overlap of 0.7957 ($\sigma = 0.0411$). These results align with expectations for semi-supervised approaches where predefined topic labels ensure thematic consistency across clusters, irrespective of which documents are assigned to which cluster.

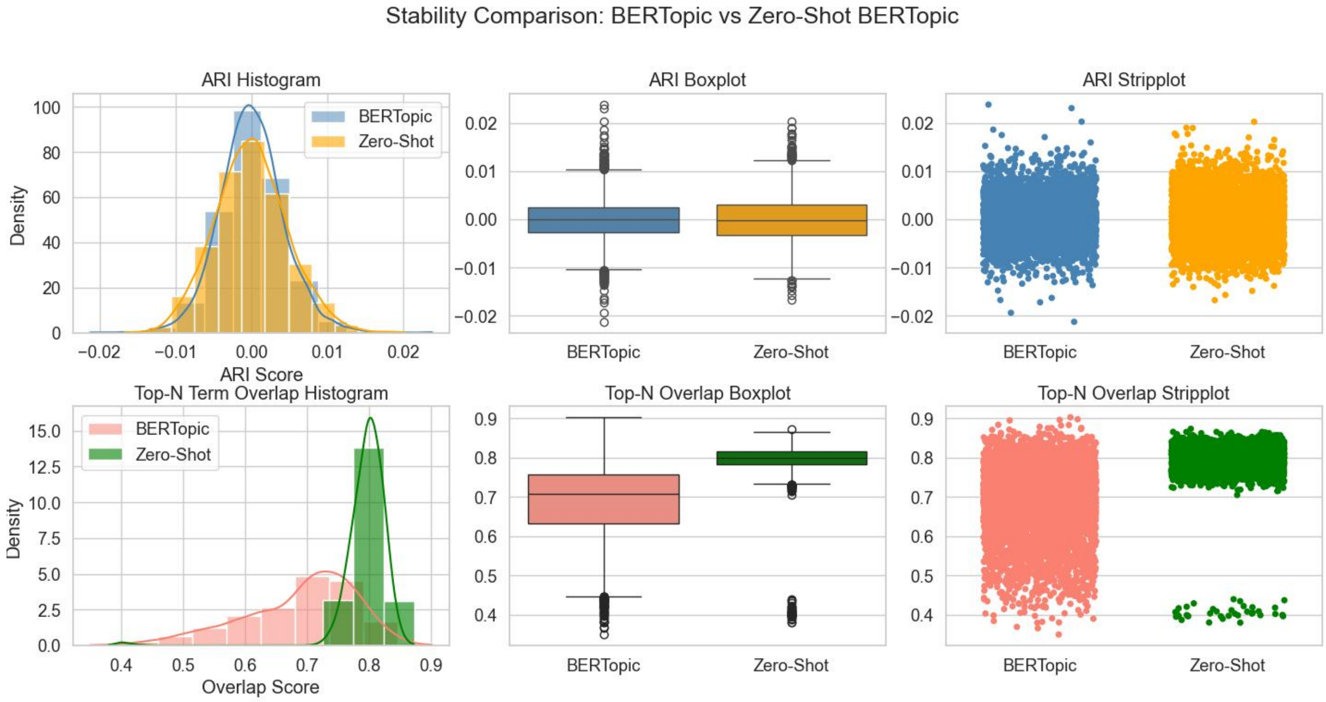


Figure 17: Stability Analysis Comparison between Baseline and Zero-shot BERTopic

4.5 Interpretability and Visualization

This section presents a visualization of the baseline and zero-shot BERTopic models through two figures: an inter-topic distance map (Fig. 11, 18), and a bar chart of the top-20 topics by size (Fig. 14, 19).

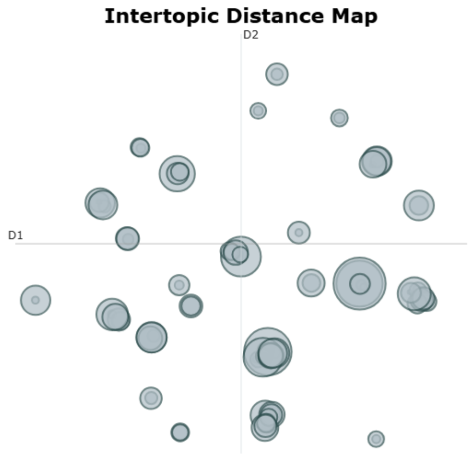


Figure 18: Inter-Topic Distance Map for Zero-shot BERTopic

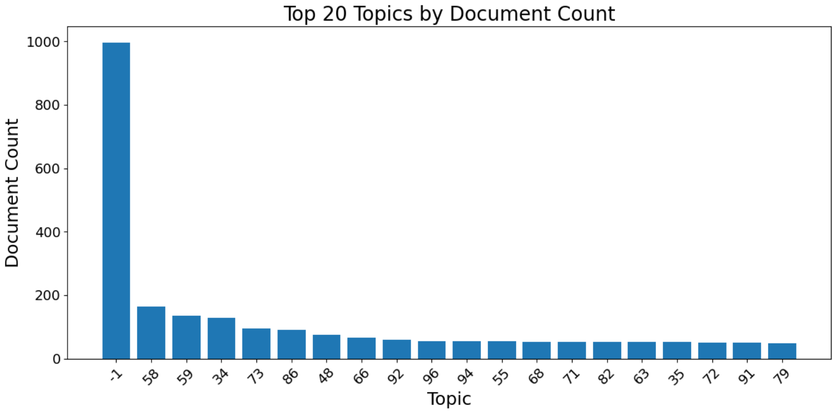


Figure 19: Top 20 Topics by Size for Zero-shot BERTopic

5 Discussion

5.1 Interpretation of Results

5.1.1 Quantitative Results

The experimental results show distinct performance trends for parameter-tuned and zero-shot BERTopic methods, both of which offer specific advantages in legislative text analysis. On the one hand, quantitative metrics indicate that the baseline BERTopic model had a marginally higher overall coherence-distinctiveness score (0.8808) compared to its zero-shot counterpart (0.8711), with a noticeably better C_v (0.8673 vs. 0.8404) and U_{Mass} (-1.2638 vs. -1.5168) score (Table 2). However, the zero-shot implementation achieved a substantially lower inter-topic similarity (0.1088 vs. 0.1497, a 27.3% decrease) and produced more than twice as many topics (98 vs. 46), indicating its ability to capture finer-grained policy distinctions through predefined topic labels. The analysis suggests that using a lower $\alpha = 0.8$ value when assigning the importance of the coherence metric (Section 4.1.1) could certainly improve the zero-shot model's overall performance.

The stability analysis revealed a difference between the behaviors of baseline and zero-shot BERTopic models. Both have comparable cluster assignment instability, with near-zero mean Adjusted Rand Index scores (BERTopic: 0.000035; zero-shot: -0.00003) and similar standard deviation (0.0042 vs. 0.00479), indicating that both are equally sensitive to input variations in document-to-topic assignments. This persistent volatility for both approaches suggests a fundamental limitation of density-based clustering in high-dimensional embedding spaces, regardless of supervision. Nevertheless, the zero-shot variant shows greater semantic stability, with a 15.41% higher mean top-n term overlap (0.7957 vs. 0.6895). Notably, the zero-shot model achieves this improved stability using the same input data, which suggests that its semi-supervised framework successfully restricts topic drift without compromising the model's ability to discover new topics. This means that while neither model achieves stable cluster assignments, meaning that document clusters are sensitive to variations in input data, both produce semantically coherent topics, with zero-shot outperforming BERTopic. They are therefore particularly suited for applications where maintaining consistent topic definitions across time or data subsets is important, such as thematic analysis, rather than for tasks requiring stable document-level clusterings.

5.1.2 Qualitative Results

On the other hand, a qualitative examination of the intertopic distribution maps (Fig. 11, 18) showed that the zero-shot version generates clusters with less overlap and of a more uniform size distribution, indicating superior semantic segregation of policy topics, in line with our quantitative results. Overlapping clusters are also theme-coherent (e.g., "situation haiti" and "counterterrorism"), and spatially proximate topics address similar policy areas (e.g., "afghanistan" and "aumf", "border security" and "human trafficking").

Conversely, the baseline BERTopic model shows more heterogeneity in cluster size and more diffuse overlaps and, consequently, less precise topic boundaries. However, the zero-shot model also has an interesting flaw in outlier control, with its outlier document cluster (-1) containing nearly 1,000 transcripts, almost five times larger than its biggest topic (Fig. 19). This contrasts with the regular model's more balanced split, where the outlier cluster (approx. 800 documents) is only slightly larger than the biggest topic (approx. 600 documents) (Fig. 14). The gap suggests that the zero-shot model's strict similarity threshold (0.85), although improving topic purity, may exclude documents that are relevant to a lesser degree but that could still enrich policy analysis.

5.2 Limitations

While this work demonstrates BERTopic's capacity for modeling legislative text, it is important to consider specific limitations that could impact its applicability. To start with, the dataset was restricted to the transcripts in the House Committee on Foreign Affairs, so the results may not apply to other congressional committees. Healthcare, technology, or environmental policy debates may have specific linguistic patterns, leading to varying levels of BERTopic performance. All such methods must be tested in follow-up research across a larger set of committees to determine robustness. Second, deploying pre-trained embedding models introduces the risk of biases while training the corpora for such models. Although the `intfloat/e5-base-v2` model was sufficient for the task at hand, it could unwittingly inherit biases from its training data and so create biased topic representations. Although the research eschewed this risk with quantitative analysis for semantic distinctiveness, a more explicit investigation of bias, e.g., topic label auditing for fairness or debiasing methods, would be an added strength to the methodology.

Third, the computational demands of transformer-based topic modeling pose practical challenges. Although optimized, the hyperparameter grid search and stability analysis were computationally intensive, and applying the methodology on larger datasets may require further optimization. Techniques such as model distillation or the use of more effective models could alleviate such a limitation. Lastly, the qualitative evaluation of topic interpretability, though valuable, was subjective. Although coherence scores provided quantitative verification, human bias played a rather strong role in deciding on topic relevance. Systematic expert polls or crowd-sourced judgments should also be incorporated in future studies.

These limitations indicate areas of improvement and highlight considerations for future study, but by no means reduce the value of this research. Mitigating these problems in future research will serve to further enhance the applicability and value of automated topic modeling to legislative research.

6 Conclusion

6.1 Reflection

6.1.1 Summary of Findings

This paper presents valuable research contributing to the application of transformer-based topic modeling techniques to legislative discourse analysis. Through rigorous experimentation on the performance of parameter-tuned and zero-shot BERTopic models on U.S. congressional hearing transcripts, the paper supplements the existing body of work in this area with a study of semantic coherence, policy relevance, and stability under data variation. The results show that the two BERTopic models have different strengths. The parameter-tuned model generates higher coherence values, while the zero-shot model achieves better semantic distinctiveness and is more interpretable, making it suitable for cases requiring fine-grained distinctions. Additionally, the zero-shot variant also shows greater stability in topic label consistency despite comparable instability in document-level assignments, which recommends it for applications requiring reproducible topic definitions.

The findings illustrate the trade-offs between the semi-supervised and unsupervised approaches in legislative text analysis. The unsupervised technique emphasizes semantic coherence, while the semi-supervised implementation facilitates more subtle policy distinctions using pre-defined labels. The results also demonstrate the potential of state-of-the-art NLP techniques to enhance the objectivity and scalability of legislative text analysis, which has traditionally relied on labor-intensive manual evaluation methods, prone to human bias.

6.1.2 Methodological Contributions

One of the most significant contributions of this paper is the development of a general methodological framework for evaluating topic models in political settings. The main research question of this work investigated the difference between parameter-tuned BERTopic and zero-shot BERTopic in identifying topics from U.S. congressional hearings transcripts (2021–2024) based on topic coherence, semantic distinctiveness, computational stability, and topic representation.

The use of quantitative metrics, as well as a qualitative analysis, ensures a thorough analysis of the model's performance. This approach validates both the mathematical accuracy of the generated topics and their practical utility for politicians and researchers. The stability testing, in particular, evaluates the consistency of the models across varying data conditions, which is important when applied to real-world contexts where agendas evolve with time. The good label stability and performance under iterative resampling of the zero-shot model demonstrate its suitability for longitudinal studies as well as cross-corpus analyses.

6.1.3 Implications of the Research

The potential impact of this research is vast. For policy-makers, automatic extraction of coherent and distinct topics from large volumes of legislative text can help evidence-based decision-making achieve better efficiency and transparency. Researchers can use these methods to identify developing policy trends and track changes in legislative debates. Using BERTopic in combination with other NLP methods, such as sentiment analysis or argument mining, can further enrich the analysis by providing a broader picture of political communication. This paper presents a case study as a proof of concept, showing how automated transformer-based topic extraction can be a useful resource for political analysis.

In conclusion, this study makes a contribution to the field of legislative text analysis by quantifying the efficacy of transformer-based topic modeling in capturing the semantic complexity of congressional hearings, as the evaluation of parameter-adjusted and zero-shot BERTopic models provides actionable insights for researchers. While challenges persist, notably those of generalizability, computational cost, and some ethical considerations, the findings indicate potential for future advances in this field. As techniques evolve, academics can find new ways of studying legislative issues, promoting more informed and inclusive policymaking. This research serves as a solid foundation from which to proceed, both as a model for methodology and as an open call for further research in the intersection of political science and NLP.

6.2 Future Work

This paper proposes several directions for future work in automated legislative topic modeling. One possibility is to expand the dataset to include other congressional committees and longer periods of time. While the current study was focused on the House Committee of Foreign Affairs, broadening the corpus to include other committees, such as healthcare or technology, would allow for a more thorough evaluation of BERTopic across various policy domains. In addition, using transcripts from previous Congresses would also aid researchers in observing the evolution of policy topics over time and recording emerging trends or changes in legislative agendas. This would also help create better baselines for evaluating topic modeling techniques in political settings.

Technical innovations in the BERTopic pipeline must also be considered. This paper used the `intfloat/e5-base-v2` embedding model, but the high development rate of transformer-based architectures means that future models could capture even more benefits. Comparison to current state-of-the-art representations, e.g., from Large Language Models such as GPT-4 or Claude, may also improve topic discriminativeness and coherence. In addition, comparisons with other clustering algorithms will help to reduce the shortcomings of the current implementation, e.g., parameter sensitivity or problems with highly overlapping topics. Improvements in computational efficiency, such as distributed training or quantization, would enable these methods to be implemented in low-resource settings.

A final opportunity is topic modeling of multilingual legislative documents. As political discourse grows more globalized, the ability to handle hearings and debates across numerous languages would provide important insight into the cross-national direction of policy and international cooperation. Future research could push BERTopic to multilingual corpora using the application of multilingual embeddings or translation-based pipelines, while attempting to avoid the particular challenge of linguistic and cultural variation in political communication.

7 Responsible Research

7.1 Reproducibility

Reproducibility is an important part of any research process. For that reason, all elements of this study, i.e., data, code, and experimental conditions, are documented and made publicly available. The dataset consists of publicly available transcripts of U.S. congressional hearings, collected and preprocessed as described in 3.1. The preprocessing code is available in the project repository[7], along with a comprehensive list of domain-specific stop-words and tokenization rules in Appendix A.

The BERTopic implementation uses open-source packages, such as `sentence-transformers` for embeddings, `umap` for dimensionality reduction, and `hdbscan` for clustering, all of which are listed in Appendix D. All hyperparameters, such as the chunk sizes, embedding models, and clustering threshold, were systematically tuned and documented, with results available in Appendix B, and default and final configurations in Appendix C. Random seeds were fixed to guarantee deterministic results (e.g., `random_state` = 42 for UMAP). Additionally, GPU and CPU configurations were explicitly noted in Appendix B. The evaluation metrics, namely topic coherence (C_v, U_{Mass}), semantic distinctiveness (pairwise cosine similarity), and computational stability (adjusted Rand index), were calculated using standardized libraries, and replication scripts were made available in the project repository.

7.2 Ethical Considerations

The ethical implications of automated legislative text analysis are multifaceted. While congressional hearings are publicly available documents, the research anonymizes who is speaking by removing personal identifiers (e.g., honorifics, timestamps) during preprocessing. This aligns with best practices in working with publicly available but sensitive textual information. Second, there is a risk of algorithmic bias arising from embedding models and topic labeling. For instance, transformer-based embeddings can inherit biases from their training datasets and distort topic representations towards larger policy narratives. To mitigate this, the study quantitatively evaluates semantic distinctiveness and coherence to ensure that topics are accurately represented and not influenced by underlying biases.

More broadly, concerns arise about the dual-use potential of this work. Congressional hearing transcripts, while publicly accessible, are recorded statements of individual views and policy positions that may be misrepresented if taken out of context. Although one of the purposes of this research is to improve legislative openness, the automated processing of sensitive political discourse risks accidental profiling, distorted policy narratives, or other discriminatory outcomes, especially if the results are interpreted without context. As a counterbalance, the paper explicitly discourages such applications in its public documentation and encourages human-in-the-loop verification in policymaking settings. The project's repository also includes a "Responsible Use" guideline, emphasizing that results should support, but not substitute for, expert judgment.

References

Alcoforado, A., Ferraz, T. P., Gerber, R., Bustos, E., Oliveira, A. S., Veloso, B. M., & Costa, A. H. R. (2022). Zeroberto: Leveraging zero-shot text classification by topic modeling. *Proceedings of the International Conference on Computational Processing of the Portuguese Language*, 125–136.

Bagozzi, B. E., & Berliner, D. (2018). The politics of scrutiny in human rights monitoring: Evidence from structural topic models of us state department human rights reports. *Political Science Research and Methods*, *6*(4), 661–677. https://doi.org/10.1017/psrm.2016.44

Blair, S. J., Bi, Y., & Mulvenna, M. D. (2020). Aggregated topic models for increasing social media topic coherence. *Applied Intelligence*, *50*(1), 138–156. https://doi.org/10.1007/s10489-019-01438-z

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84. https://doi.org/10.1145/2133806.2133826

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation (J. Lafferty, Ed.). *Journal of Machine Learning Research*, *3*(4–5), 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993

Boutaleb, A., Picault, J., & Grosjean, G. (2024). Bertrend: Neural topic modeling for emerging trends detection. *arXiv preprint arXiv:2411.05930*.

[7] The project repository may be found here: https://doi.org/10.5281/zenodo.15715846.

Campello, R. J. G. B., Moulavi, D., Zimek, A., & Sander, J. (2015). Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Transactions on Knowledge Discovery from Data*, *10*(1), 1–51. https://doi.org/10.1145/2733381

Congressional Research Service. (2010, March). *Hearings in the u.s. senate: A guide for preparation and procedure* (tech. rep. No. RL30548) (Analyst on Congress and the Legislative Process). Congressional Research Service. %7Bhttps://www.everycrsreport.com/reports/RL30548.html%7D

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://arxiv.org/abs/1810.04805

Egger, R., & Yu, J. (2022). A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Frontiers in Sociology*, *7*, 886498. https://doi.org/10.3389/fsoc.2022.886498

Gong, D. (2025). *A practical guide to bertopic for transformer-based topic modeling* [Towards Data Science. Accessed: 2025-05-31]. https://towardsdatascience.com/a-practical-guide-to-bertopic-for-transformer-based-topic-modeling/

Greene, D., & Cross, J. P. (2017). Exploring the political agenda of the european parliament using a dynamic topic modeling approach. *Political Analysis*, *25*(1), 77–94. https://doi.org/10.1017/pan.2016.7

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Le, D. T., Vu, N. T., & Blessing, A. (2016). Towards a text analysis system for political debates. *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 134–139.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791. https://doi.org/10.1038/44565

McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. https://arxiv.org/abs/1802.03426

Mimno, D., Wallach, H., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. *Proceedings of the 2011 conference on empirical methods in natural language processing*, 262–272.

Palatucci, M., Pomerleau, D., Hinton, G. E., & Mitchell, T. M. (2009). Zero-shot learning with semantic output codes. *Advances in Neural Information Processing Systems (NeurIPS)*, *22*, 1410–1418.

Quinn, K. M., Monroe, B. L., Colaresi, M., Crespin, M. H., & Radev, D. R. (2010). How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science*, *54*(1), 209–228. https://doi.org/10.1111/j.1540-5907.2009.00427.x

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2023). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://doi.org/10.48550/arXiv.1910.10683

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. http://arxiv.org/abs/1908.10084

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the Space of Topic Coherence Measures, 399–408. https://doi.org/10.1145/2684822.2685324

Ruiz-Dolz, R., Heras, S., & Garcia-Fornes, A. (2022). Automatic debate evaluation with argumentation semantics and natural language argument graph networks. https://arxiv.org/abs/2203.14647

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2020). MPNet: Masked and Permuted Pre-training for Language Understanding. https://doi.org/10.48550/arXiv.2004.09297

Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., & Wei, F. (2022). Text Embeddings by Weakly-Supervised Contrastive Pre-training. https://doi.org/10.48550/arXiv.2212.03533

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MINILM: deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Proceedings of the 34th International Conference on Neural Information Processing Systems*. https://dl.acm.org/doi/10.1145/3677389.3702603

A List of Stop and Keep Words

A.1 Stop Words

- **Common stopwords:** the, and, to, of, a, in, that, is, it, for, time, today, really, much, important, yes, no, many
- **Congressional-specific stopwords:** thank, know, think, chair, gentleman, also, want, committee, subcommittee, people, chairman, united, states, would, could, will, like, going, go, get, one, well, work, back, should, question, questions, adjourn, adjourned, appendix, submit, member, members, nil, objection, objections, statement, statements, prepared statement, director, new, secretary, www, http, house, gov
- **Verb forms:** said, say, says, ask, asked, want, needs, need, make
- **Pronouns:** i, you, he, she, it, we, they, my, mine, your, yours, his, hers, their, theirs, our, ours, us, him, her

A.2 Keep Words

climate, energy, security, trade, commerce, policy, economic, foreign, affairs

B Parameter Tuning

Chunk Size and Embedding Model

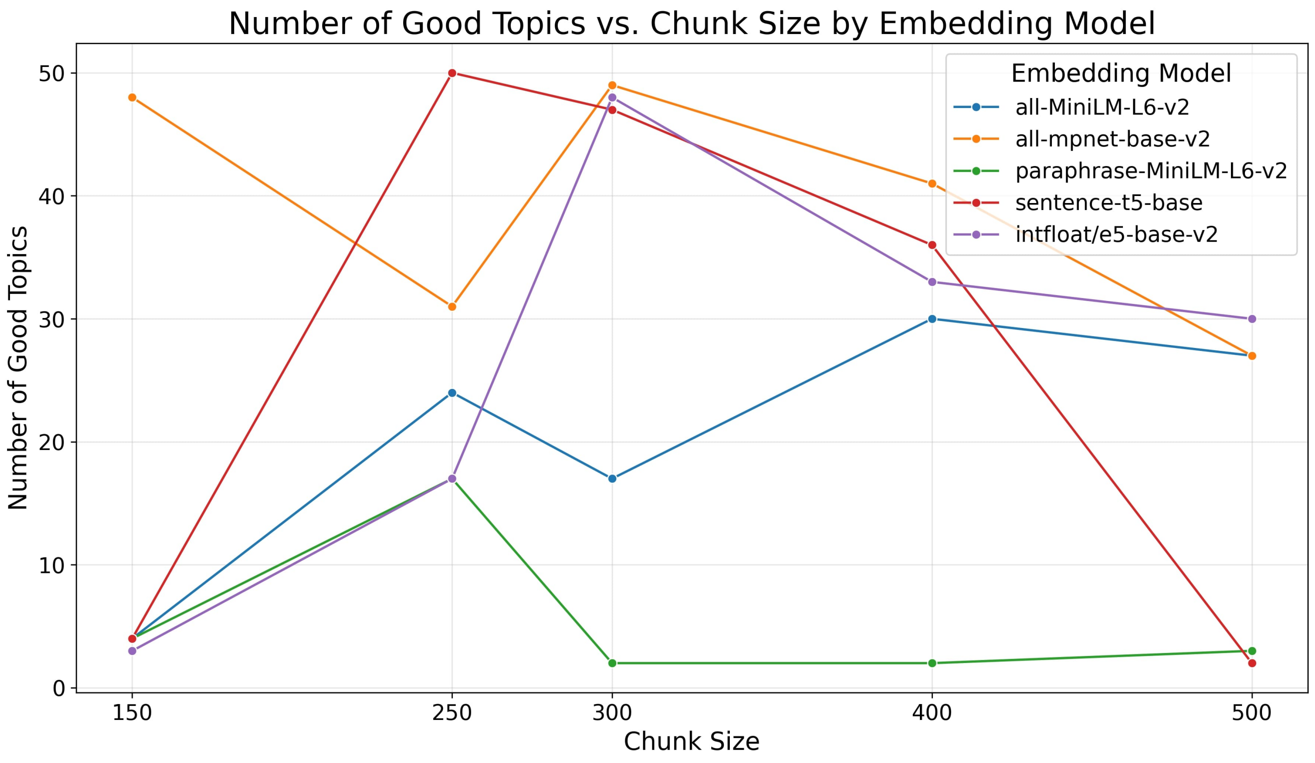


Figure 20: Number of Good Topics vs. Chunk Size by Embedding Model

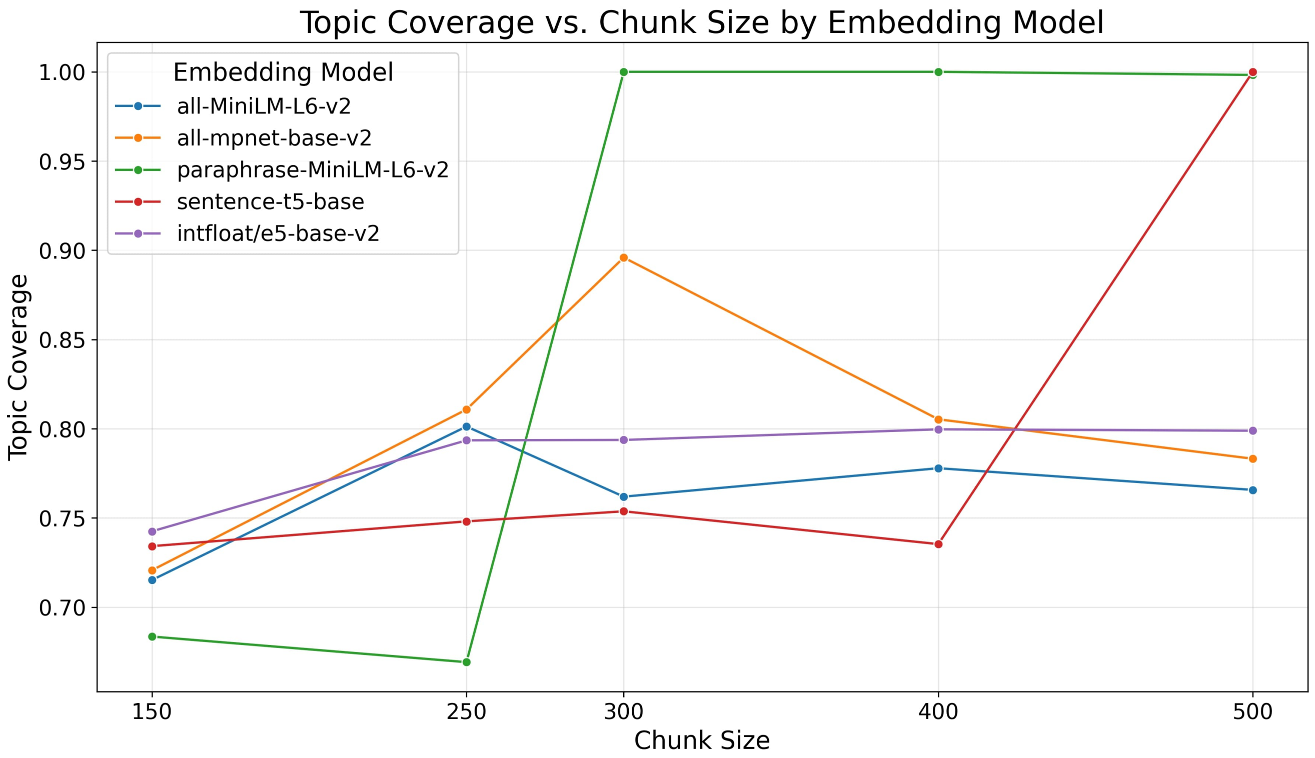


Figure 21: Topic Coverage vs. Chunk Size by Embedding Model

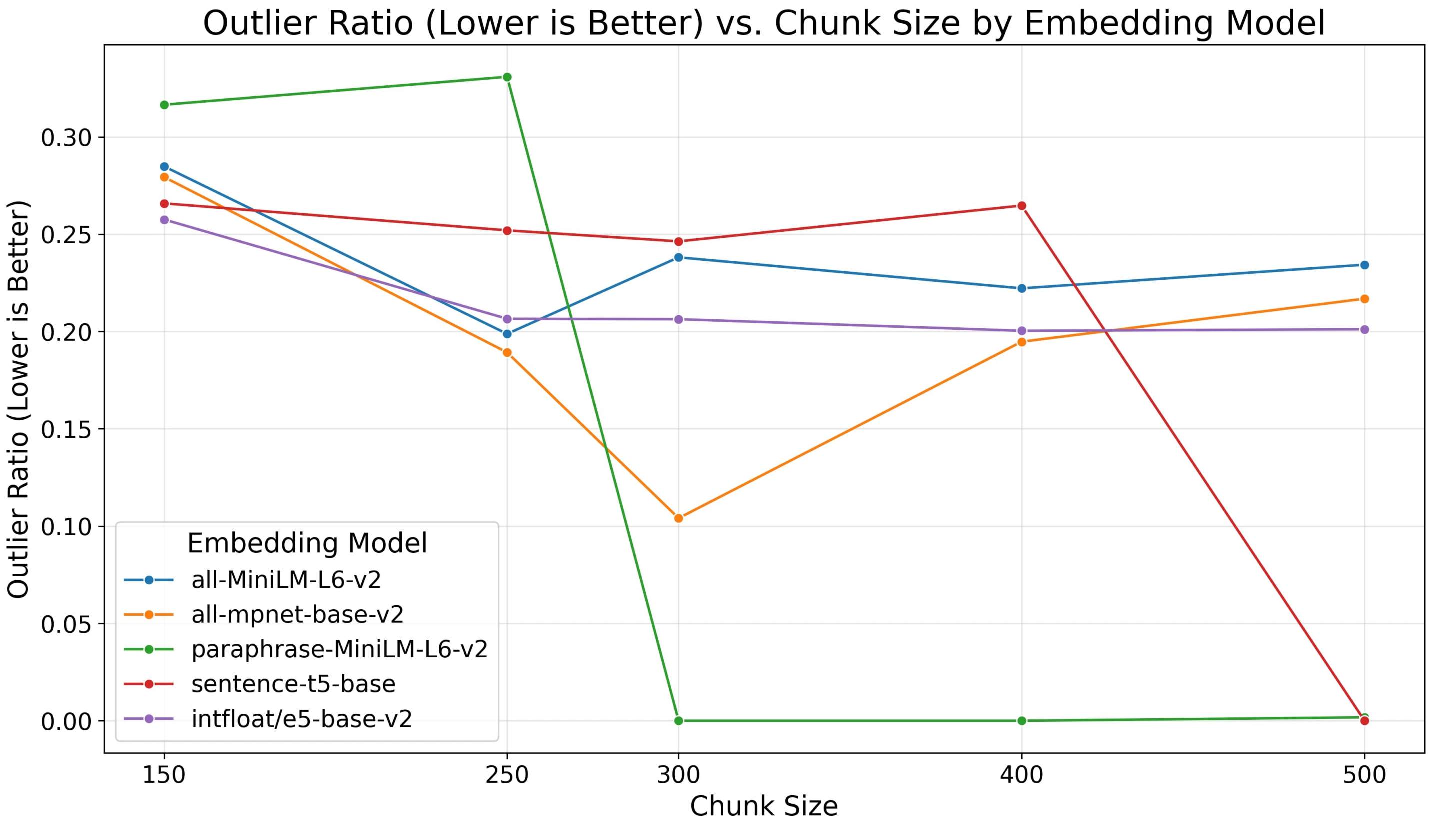


Figure 22: Outlier Ratio vs. Chunk Size by Embedding Model

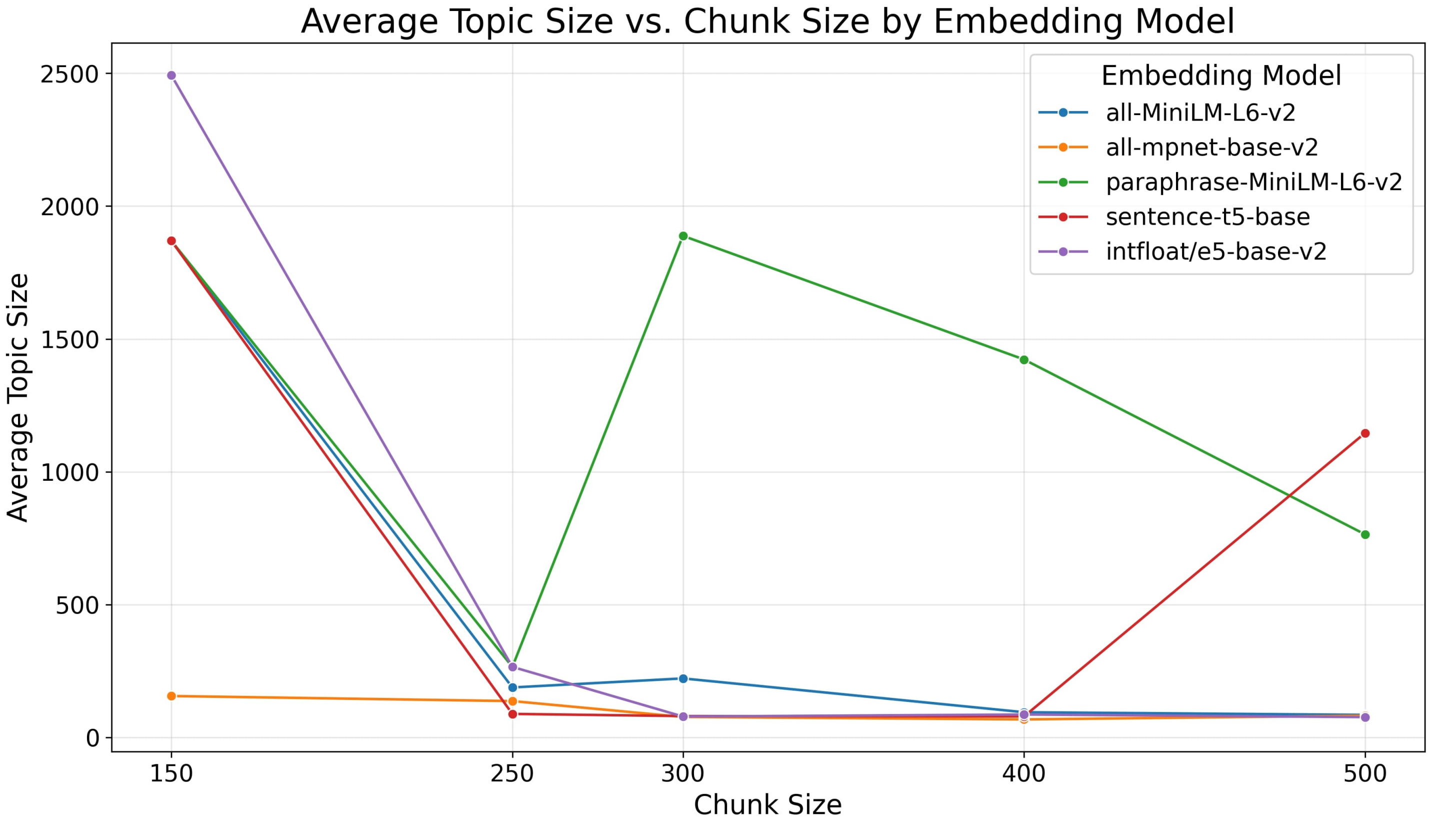


Figure 23: Average Topic Size vs. Chunk Size by Embedding Model

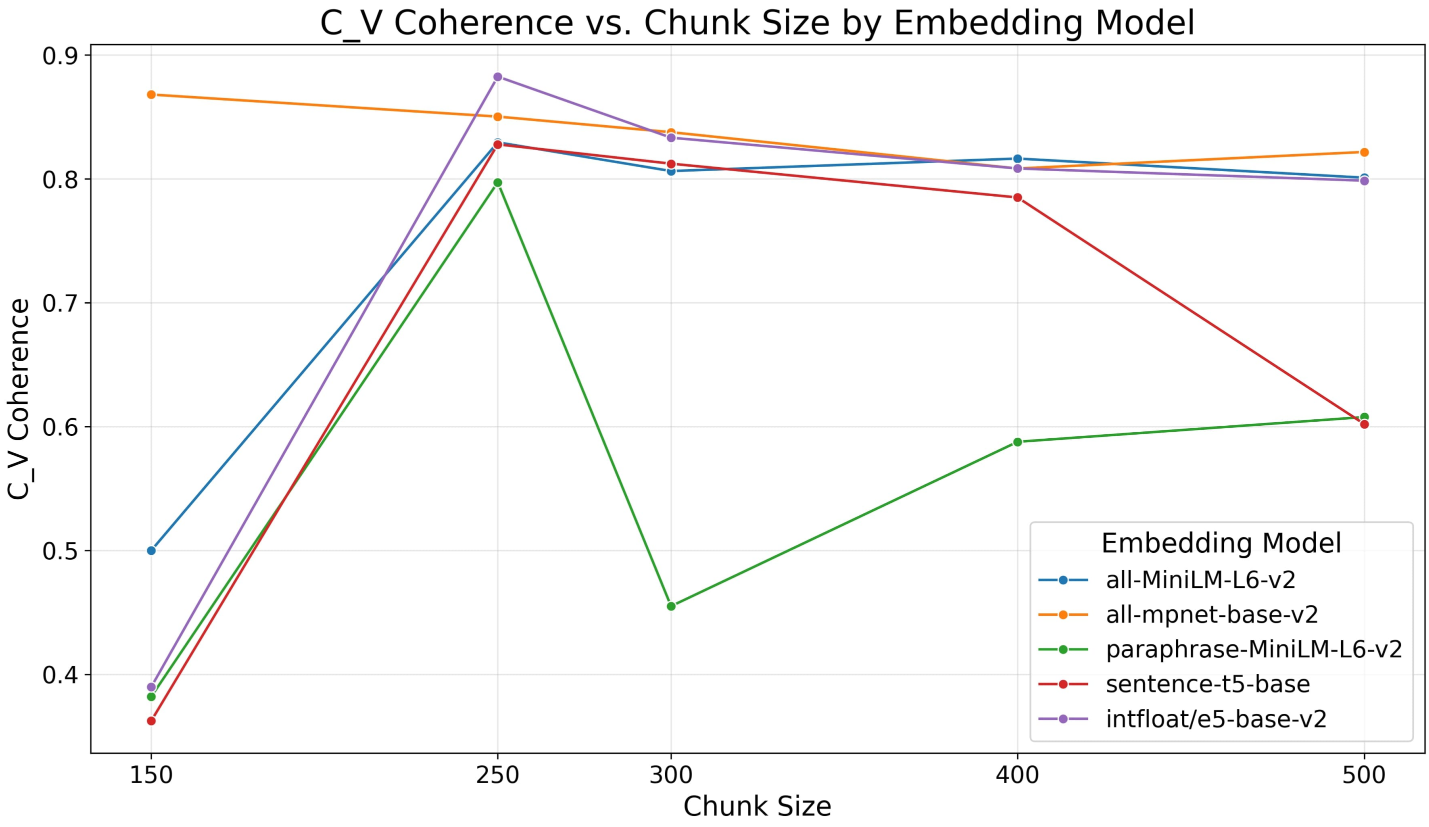


Figure 24: C_v Coherence vs. Chunk Size by Embedding Model

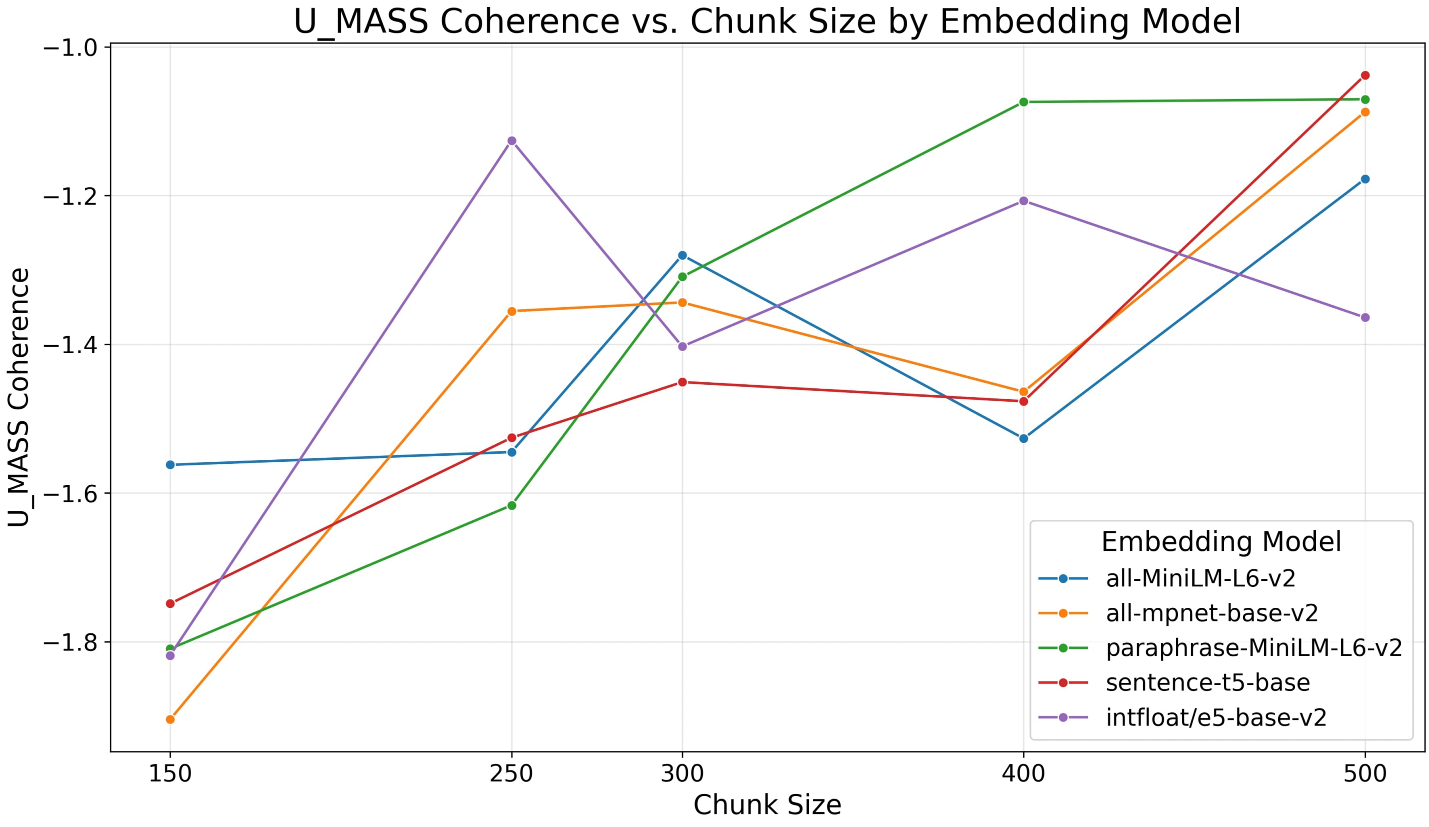


Figure 25: U_{Mass} Coherence vs. Chunk Size by Embedding Model

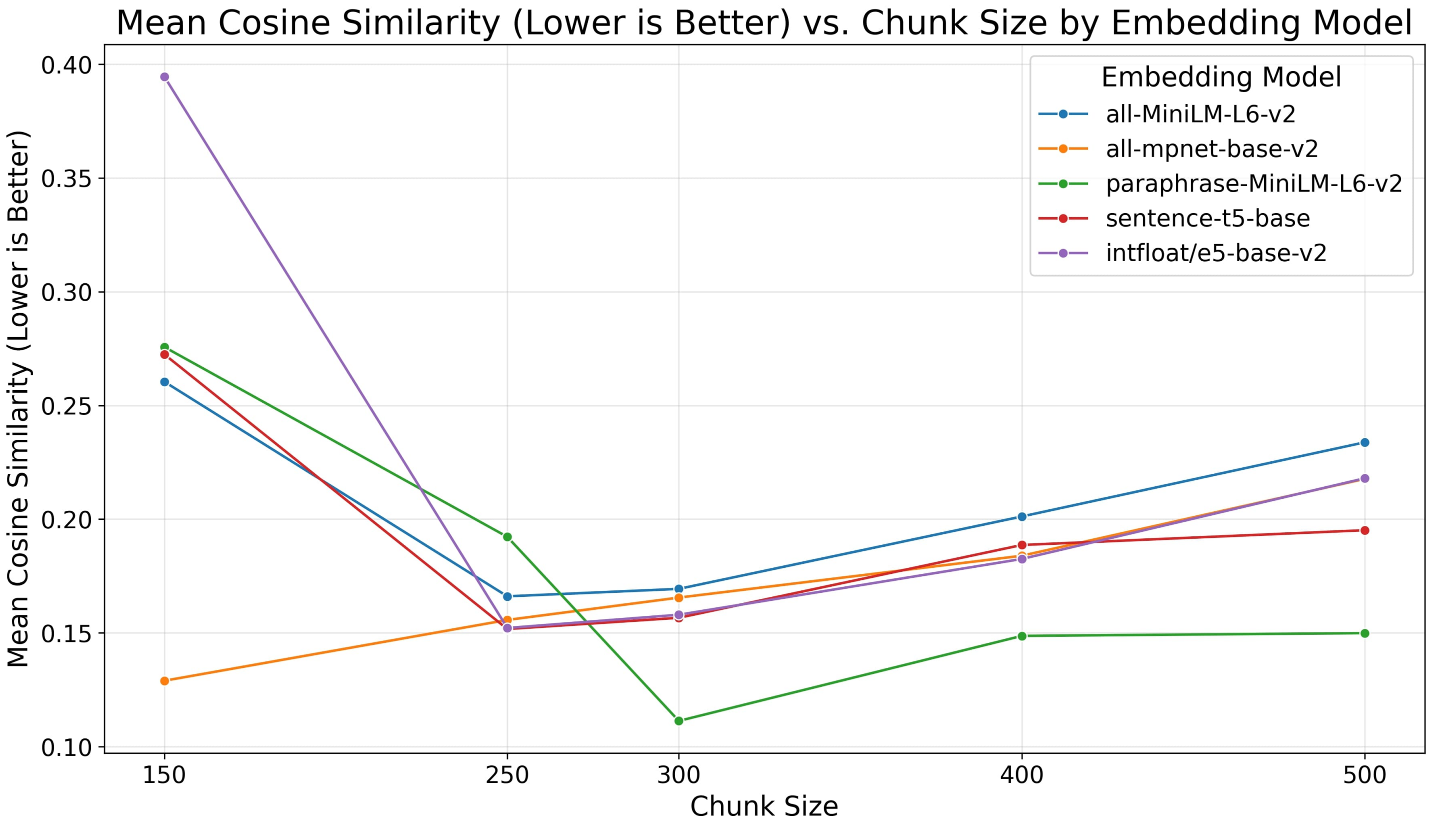


Figure 26: Mean Cosine Similarity vs. Chunk Size by Embedding Model

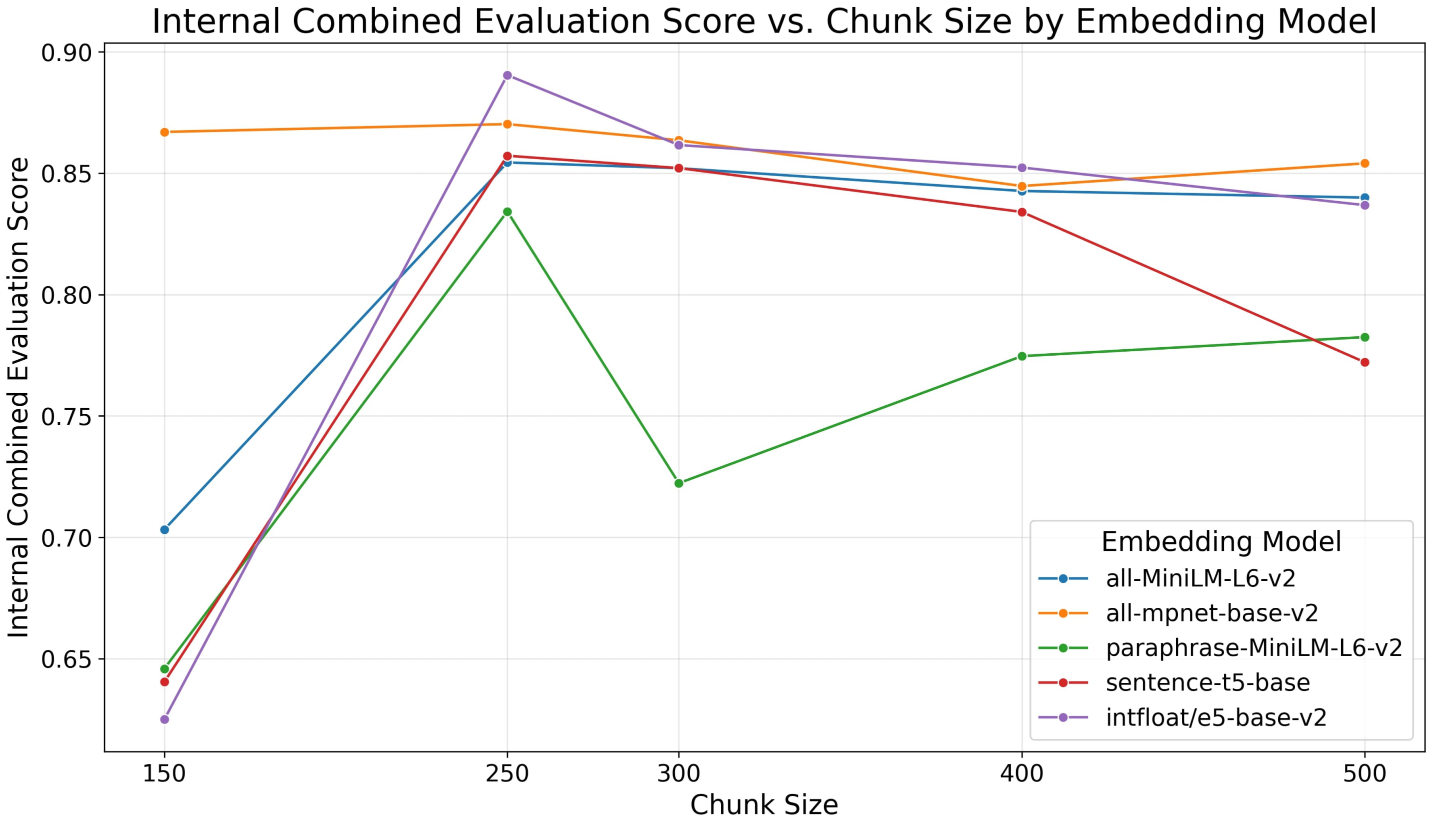


Figure 27: Combined Evaluation Score vs. Chunk Size by Embedding Model

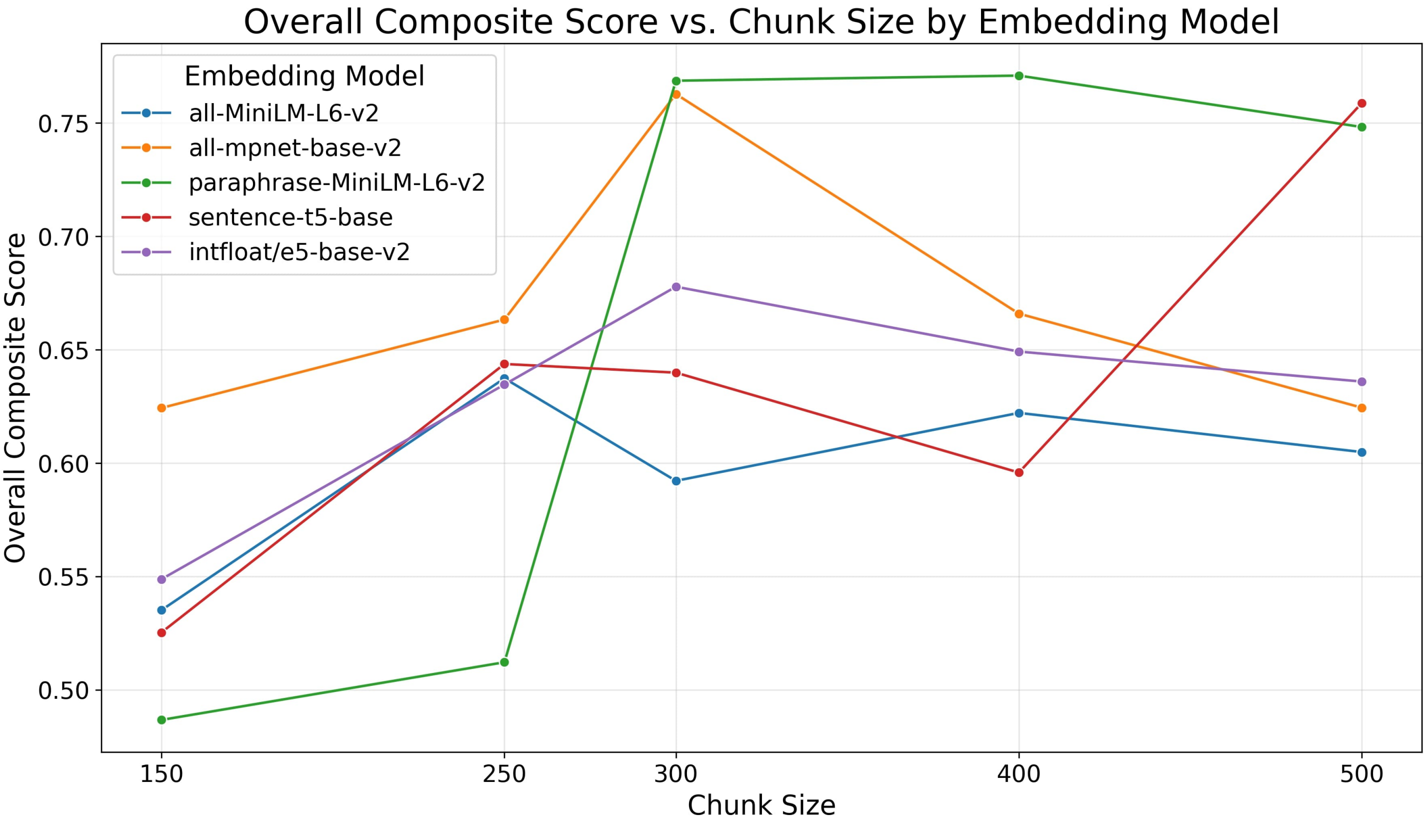


Figure 28: Composite Score vs. Chunk Size by Embedding Model

Basic BERTopic Parameters

Individual Tuning

300 Tokens

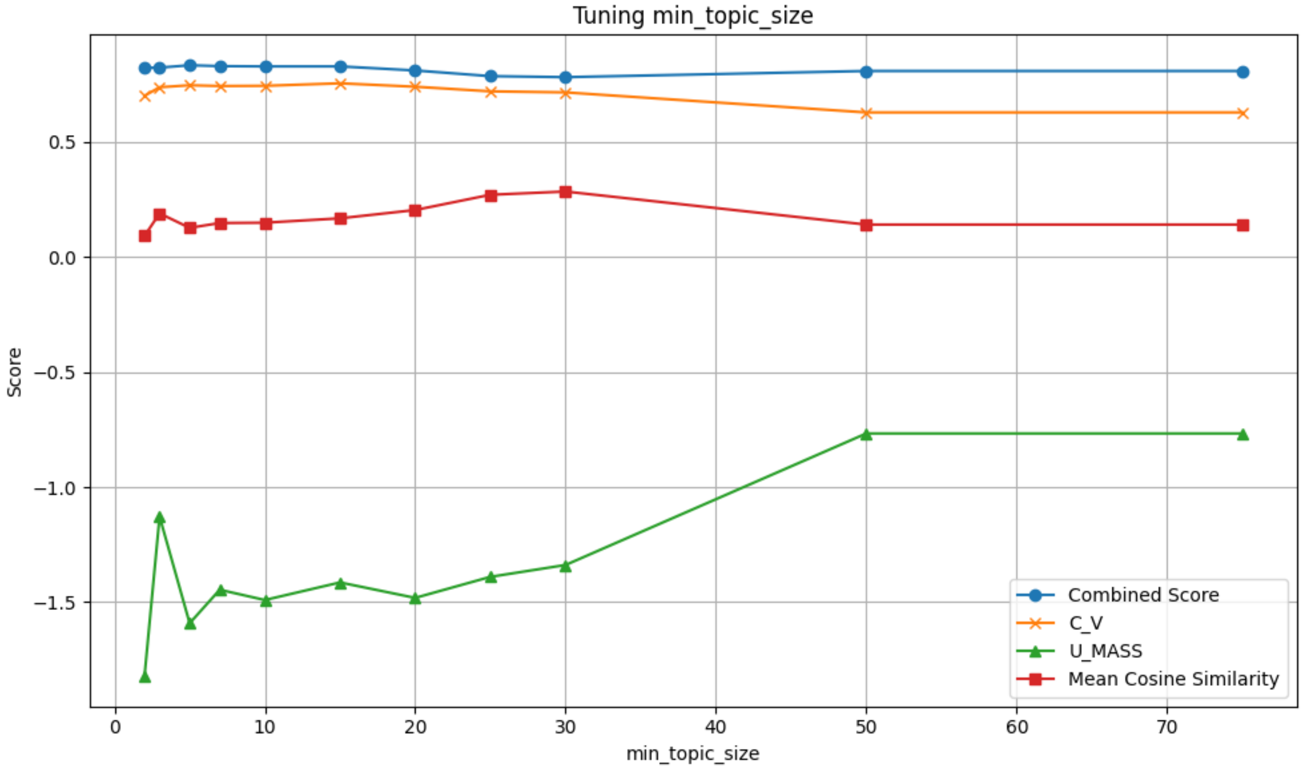


Figure 29: `min_topic_size` tuning for 300 tokens

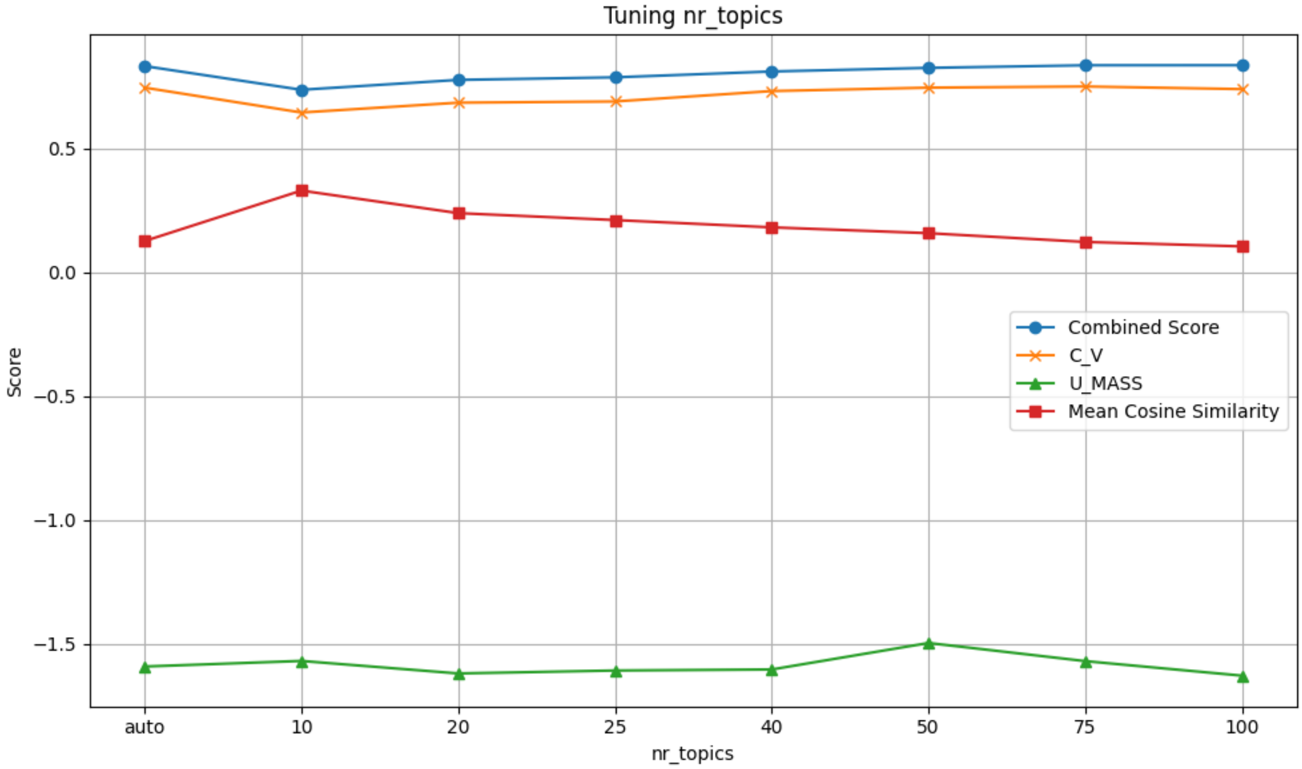


Figure 30: `nr_topics_tuning` tuning for 300 tokens

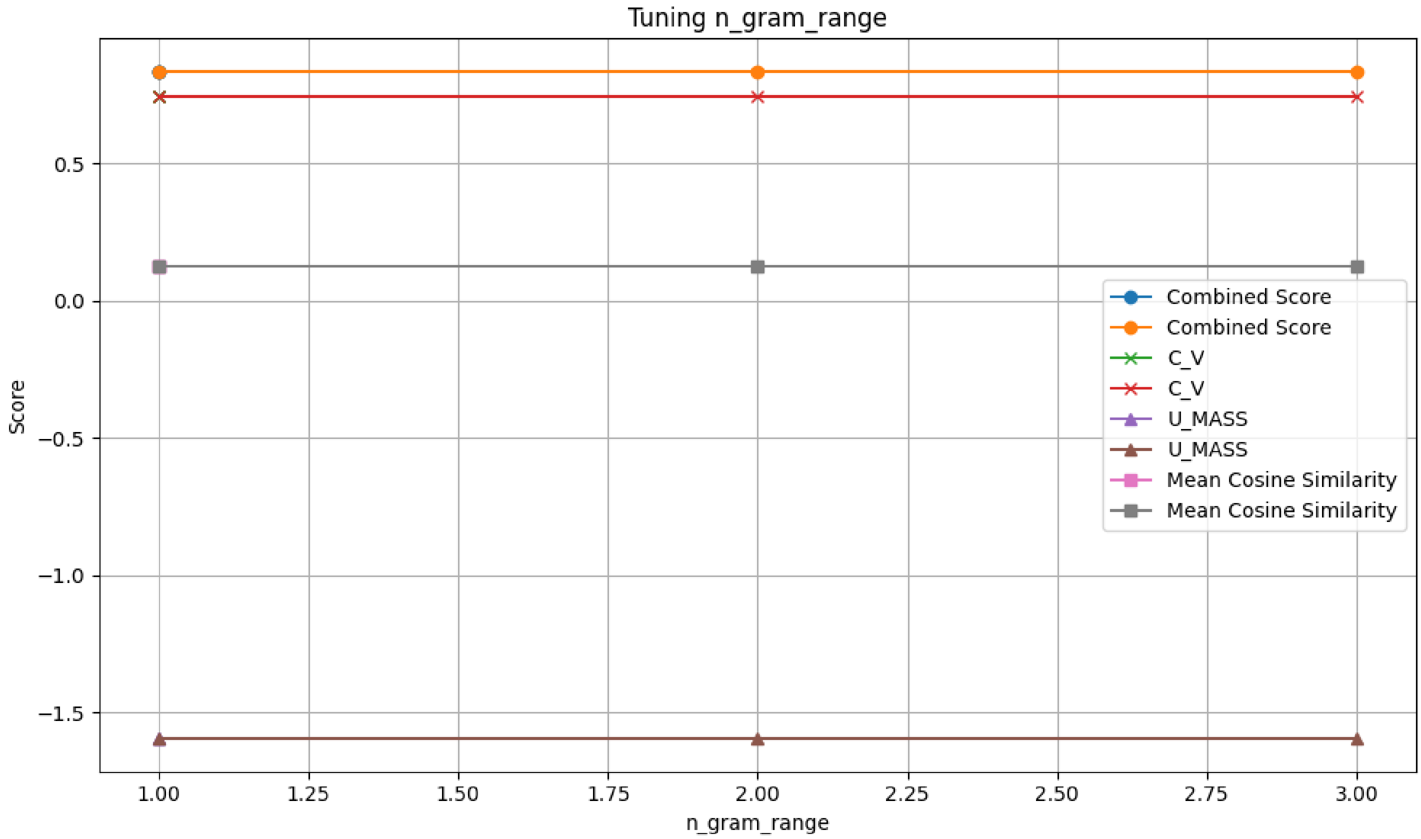


Figure 31: `n_gram_range` tuning for 300 tokens

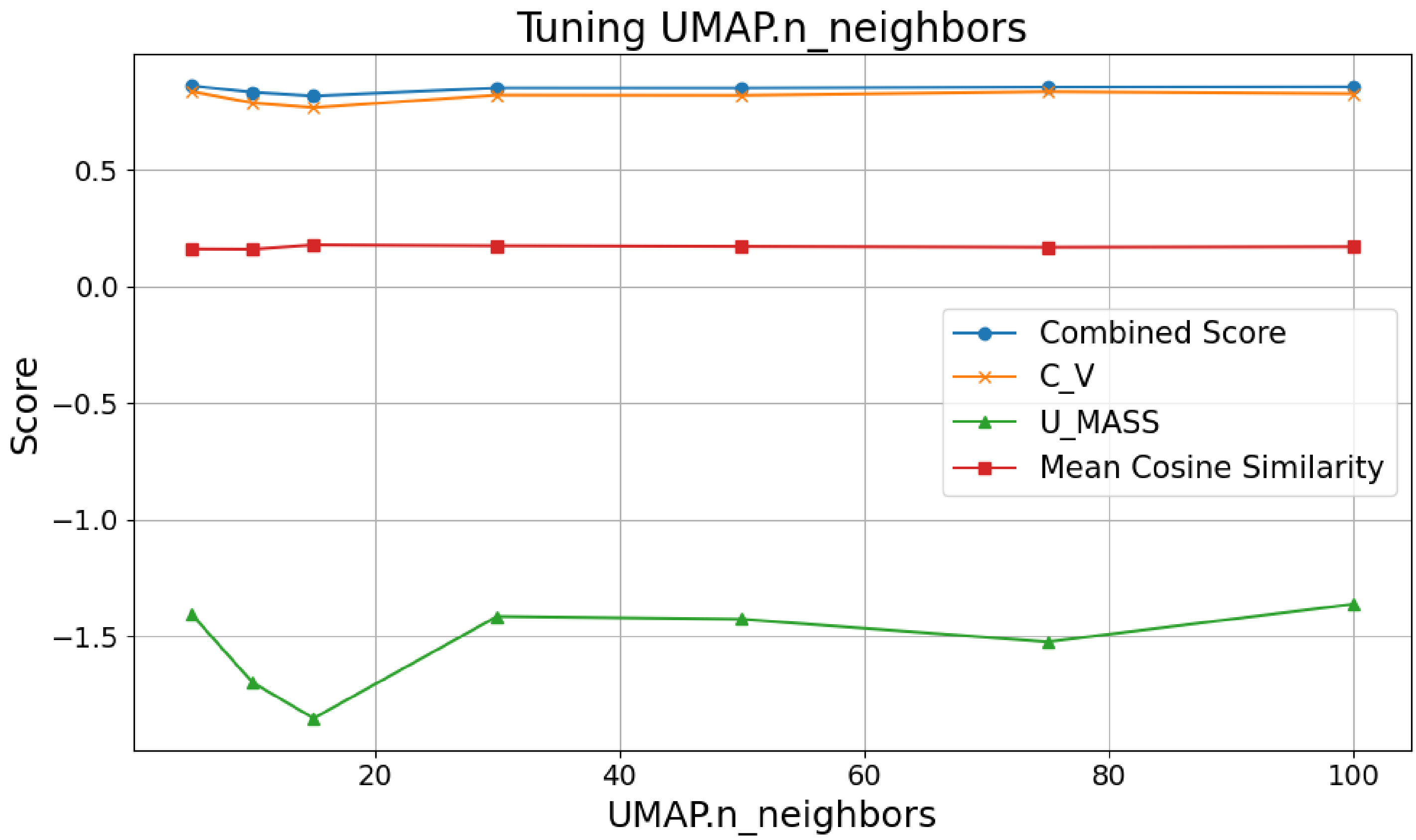


Figure 32: UMAP `n_neighbors` tuning for 300 tokens

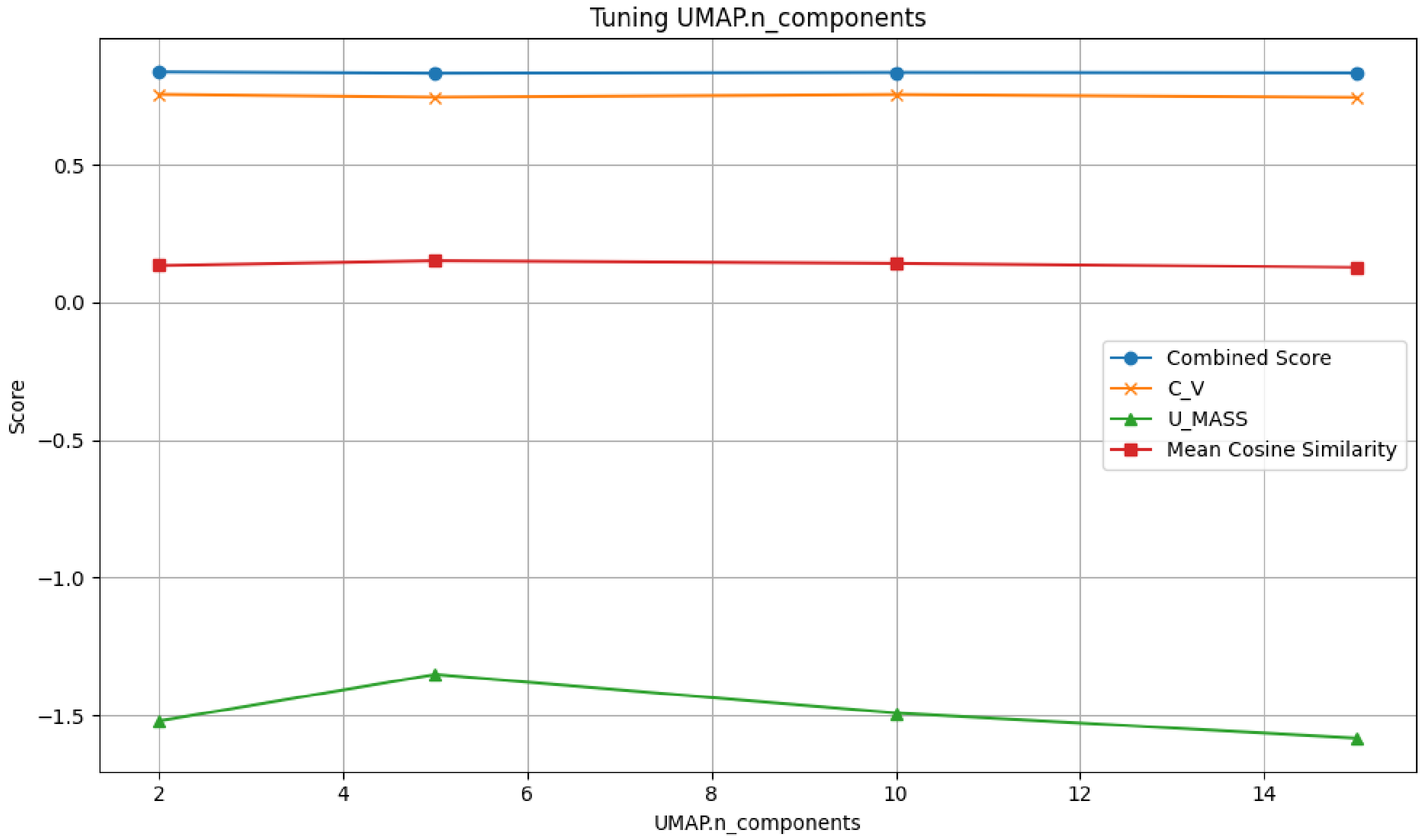


Figure 33: UMAP `n_components` tuning for 300 tokens

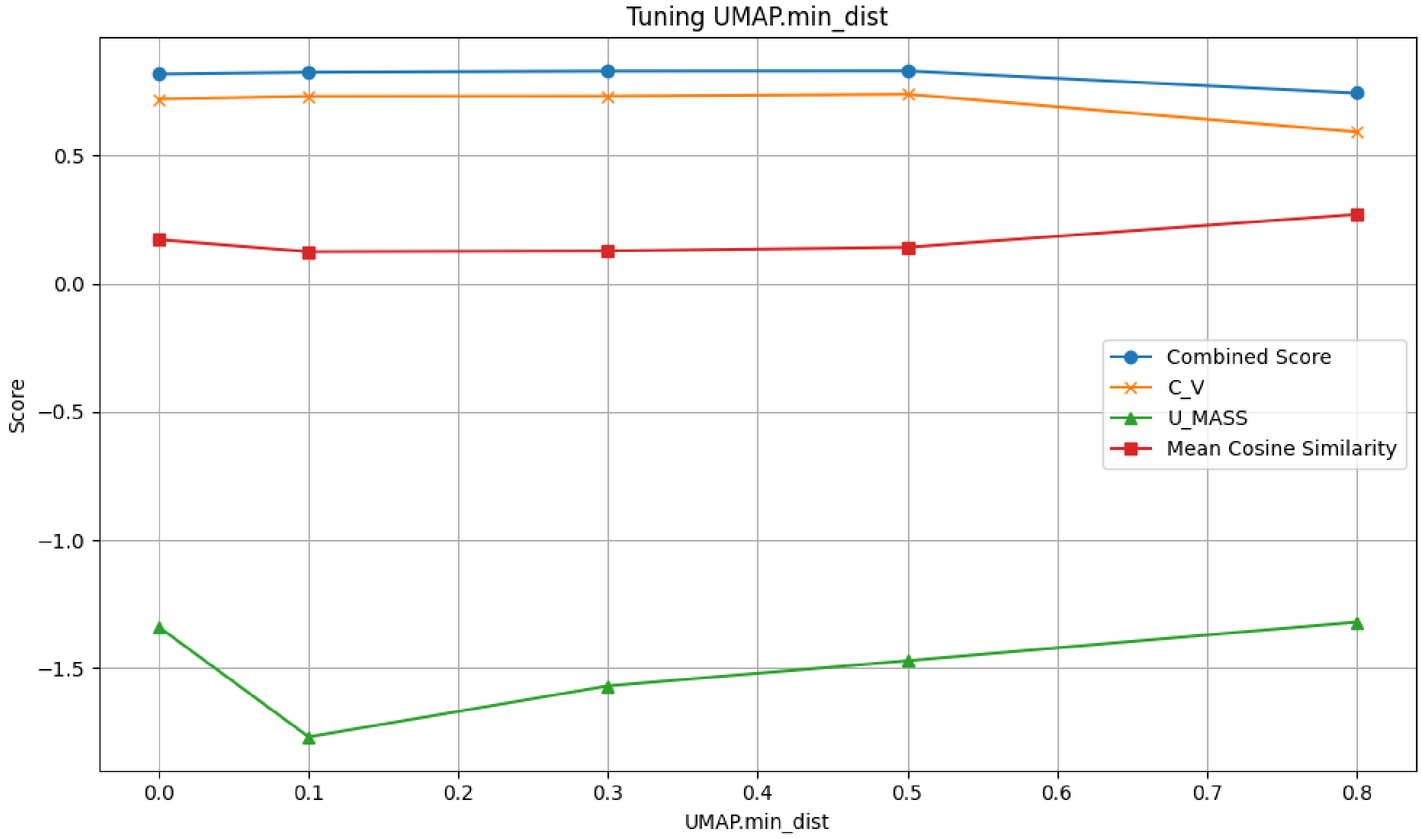


Figure 34: UMAP `min_dist` tuning for 300 tokens

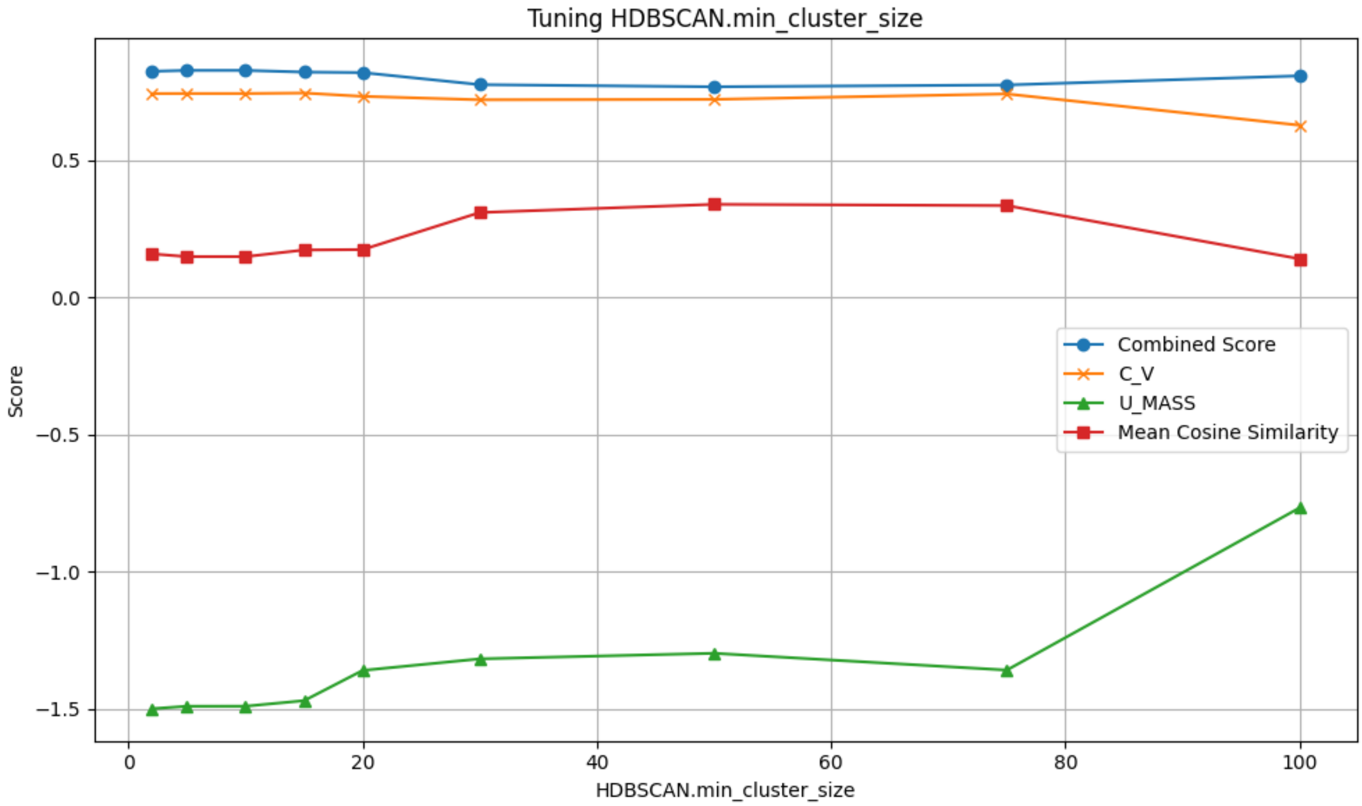


Figure 35: HDBSCAN `min_cluster_size` tuning for 300 tokens

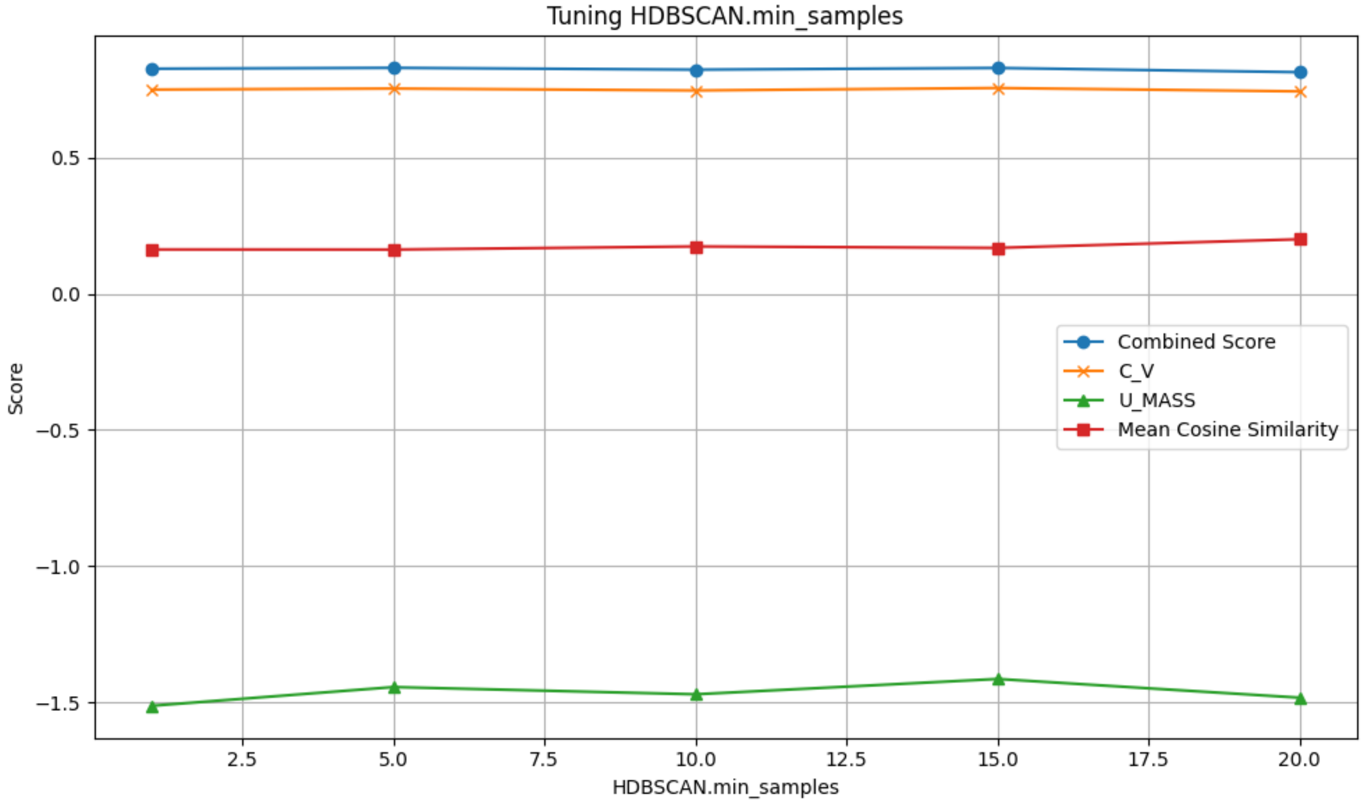


Figure 36: HDBSCAN `min_samples` tuning for 300 tokens

400 Tokens

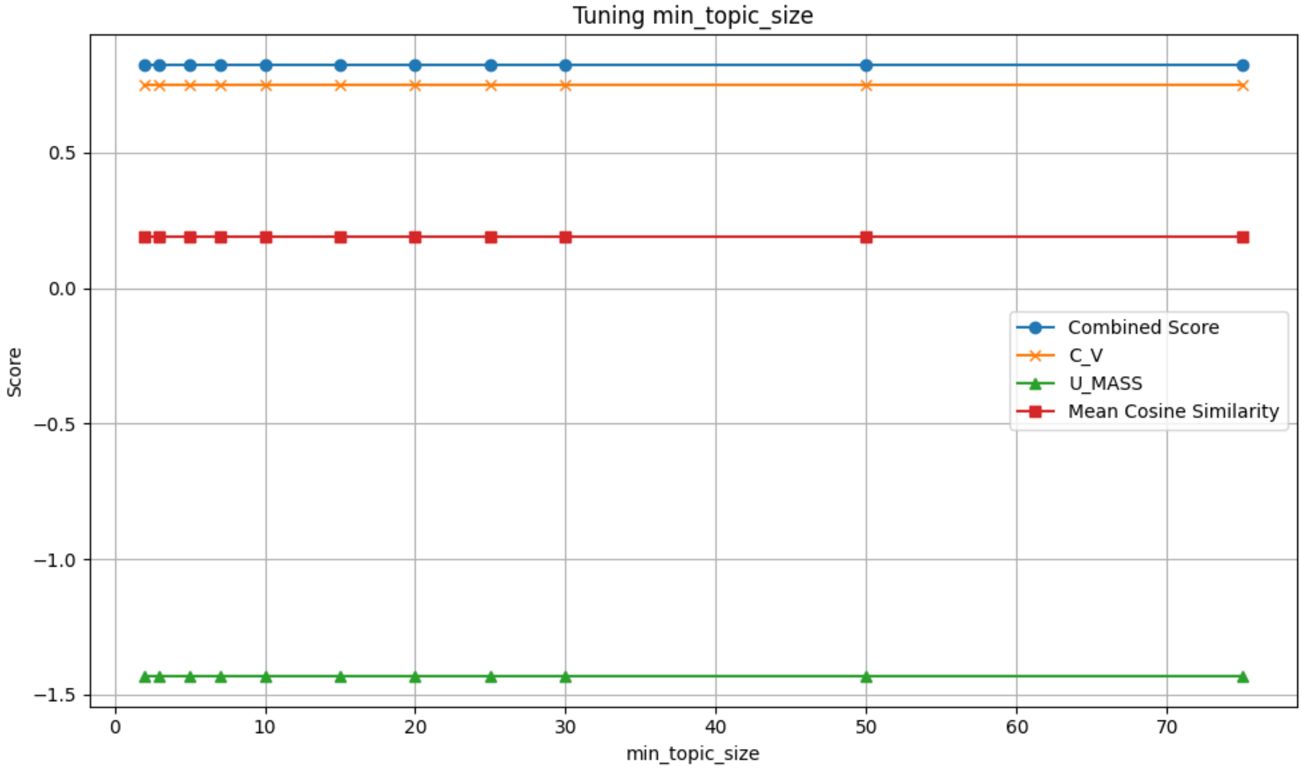


Figure 37: `min_topic_size` tuning for 400 tokens

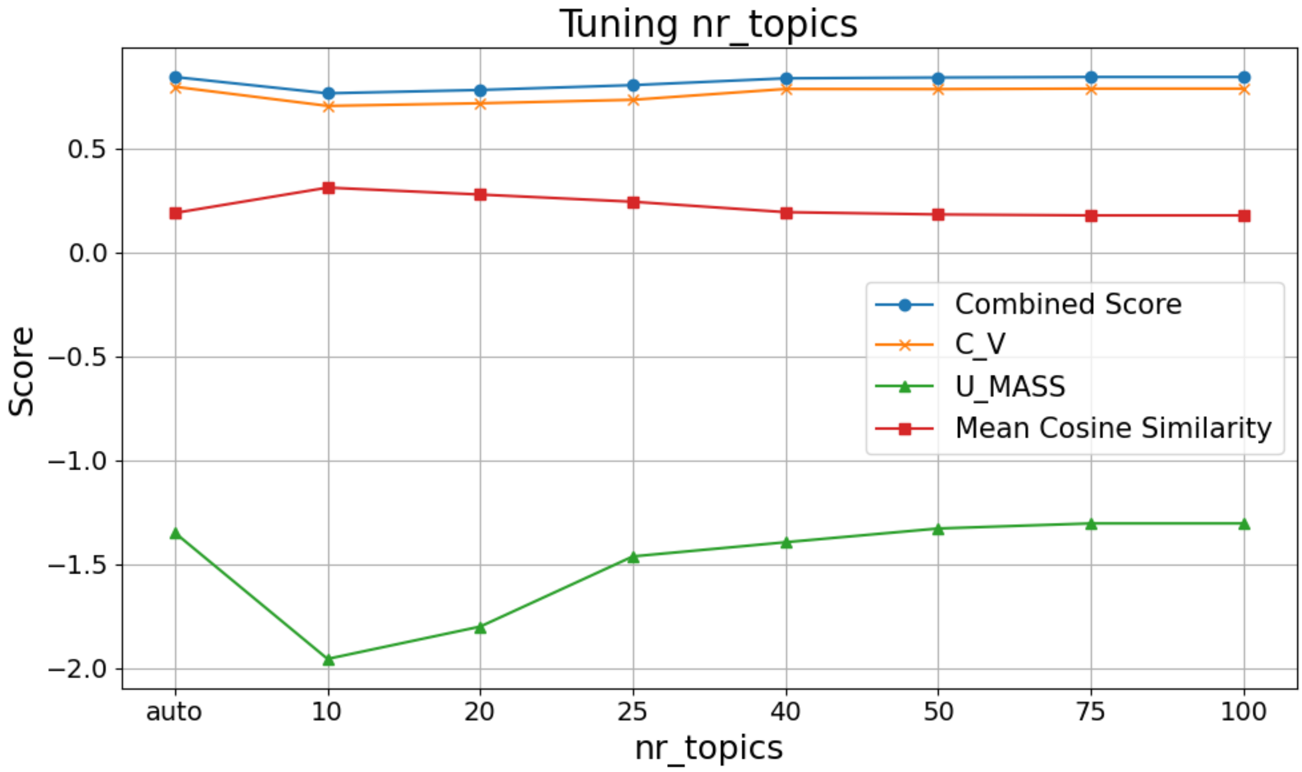


Figure 38: `nr_topics_tuning` tuning for 400 tokens

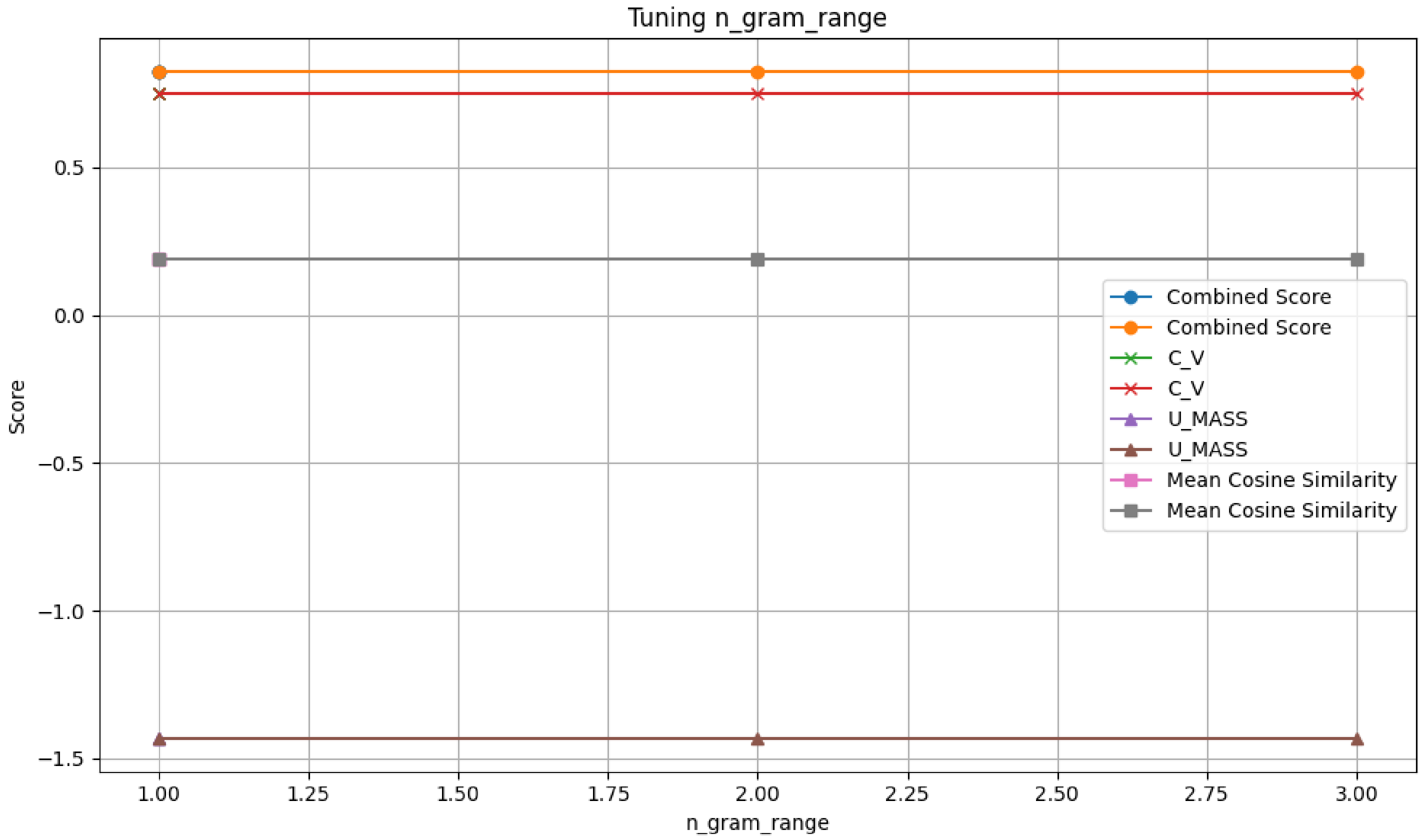


Figure 39: `n_gram_range` tuning for 400 tokens

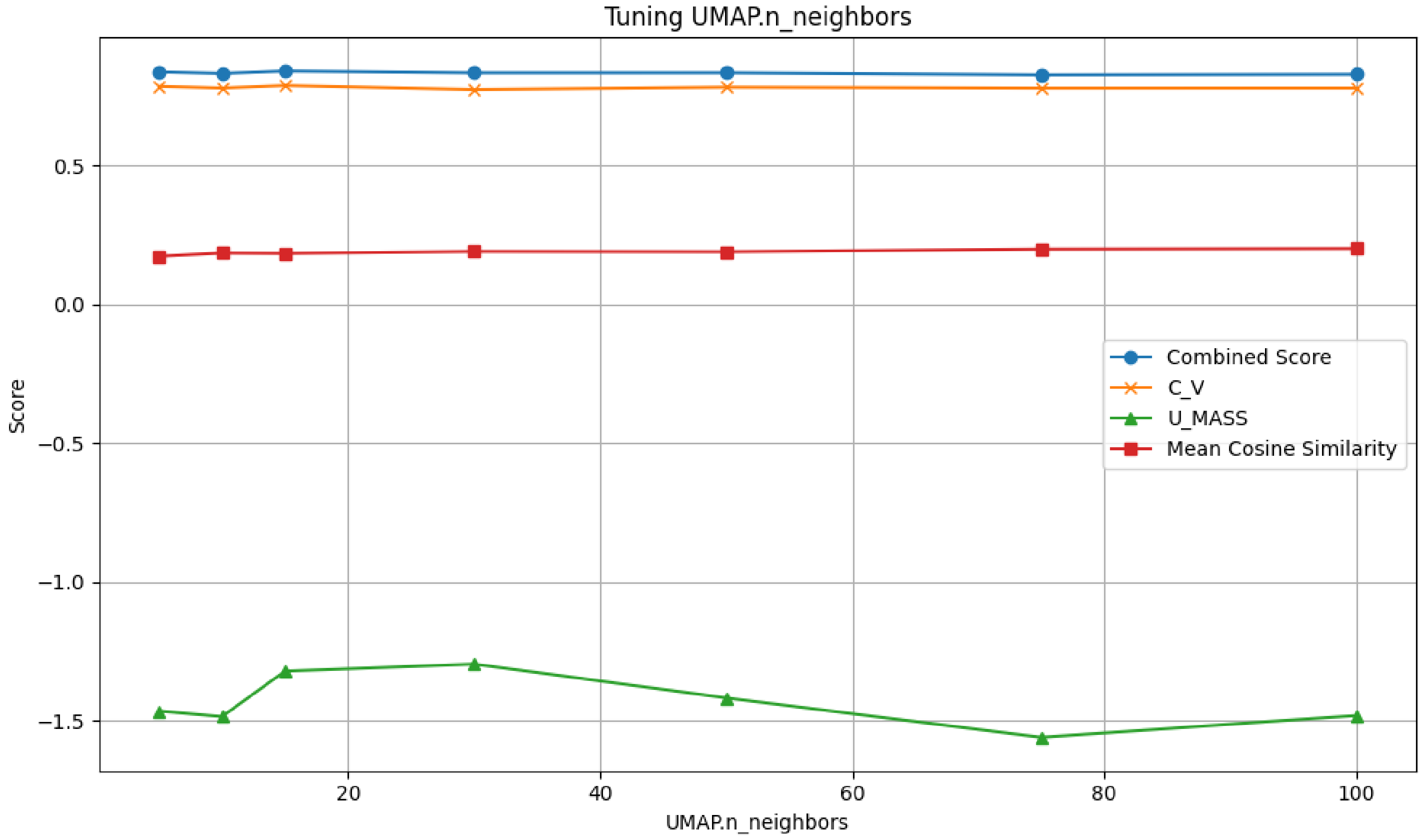


Figure 40: UMAP `n_neighbors` tuning for 400 tokens

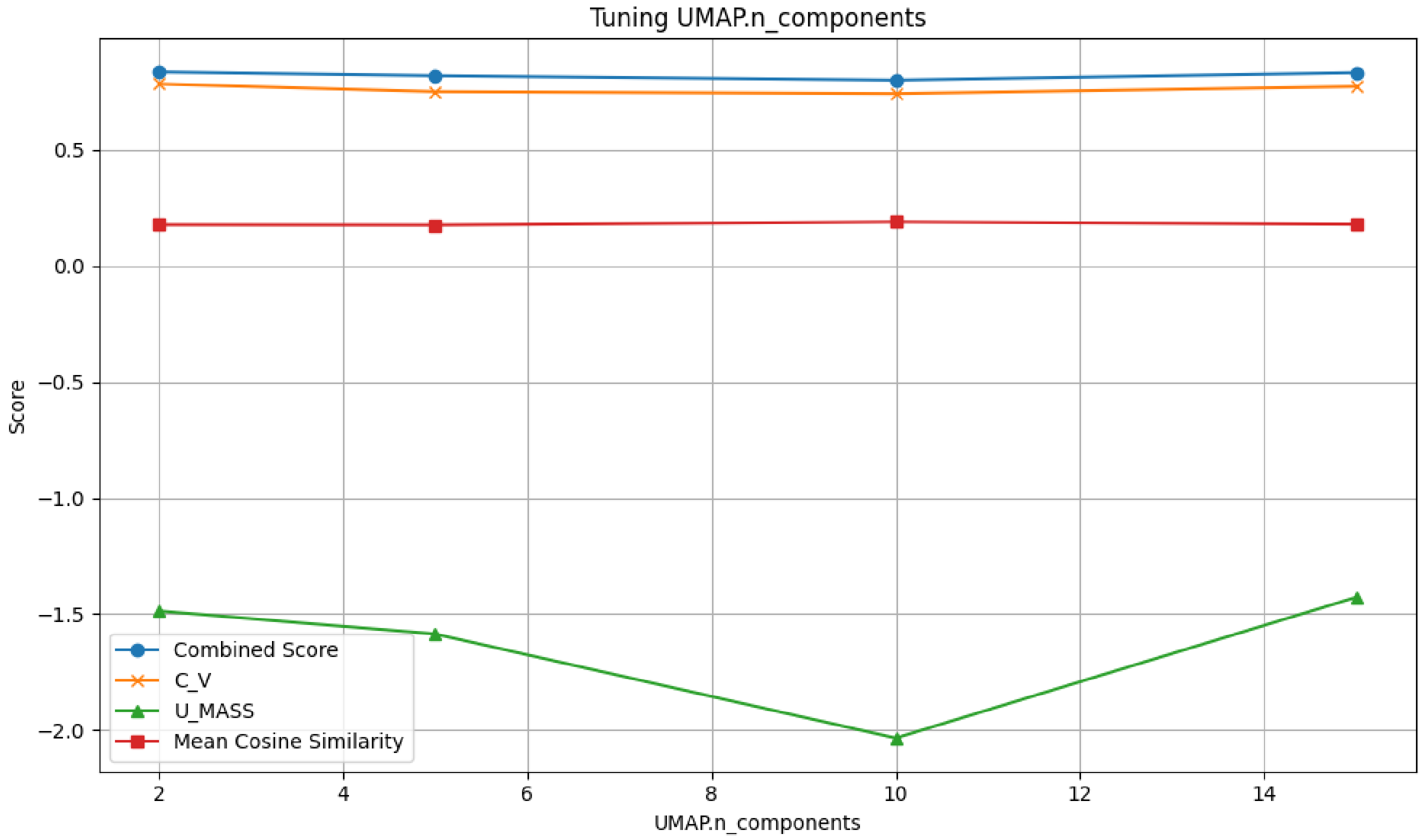


Figure 41: UMAP `n_components` tuning for 400 tokens

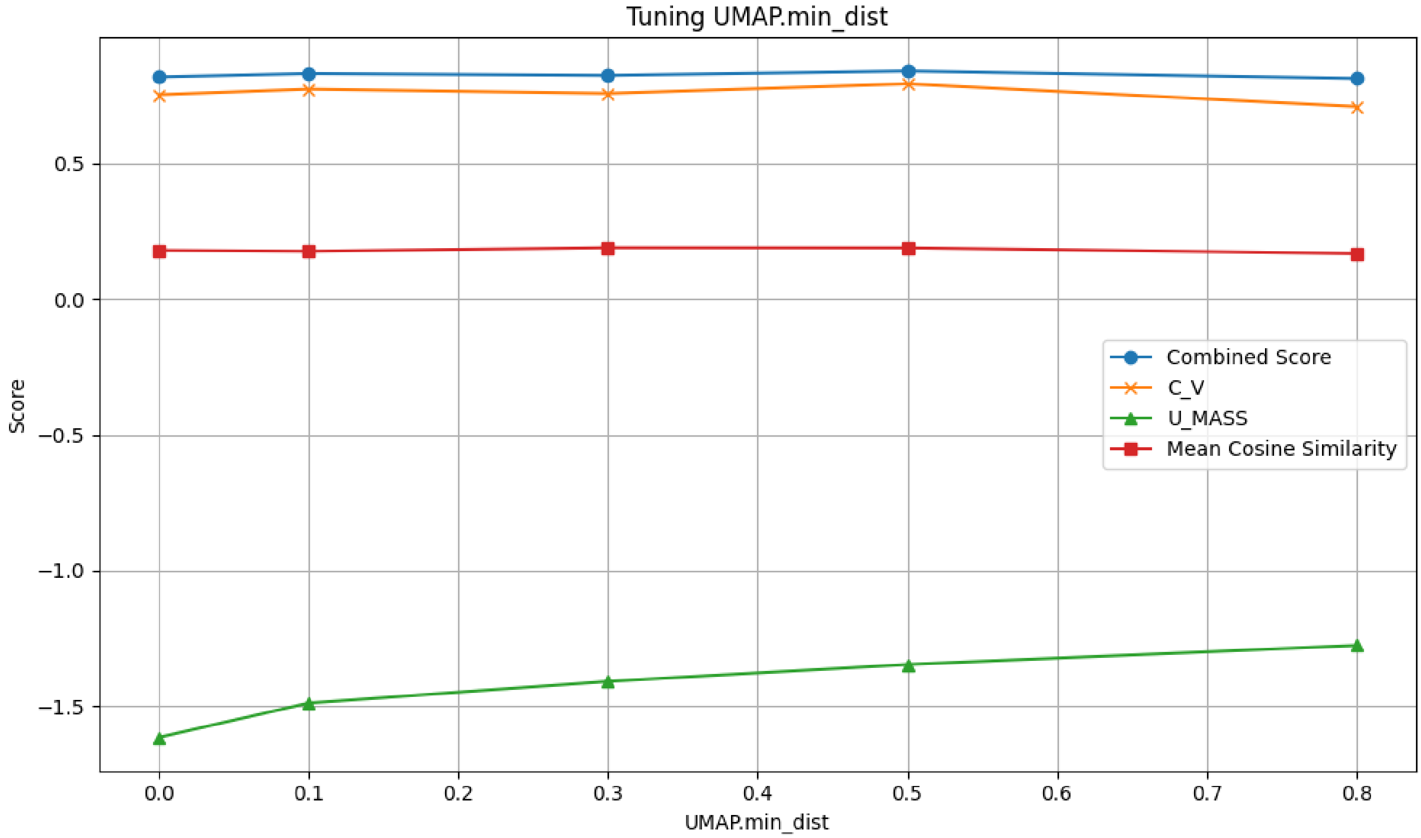


Figure 42: UMAP `min_dist` tuning for 400 tokens

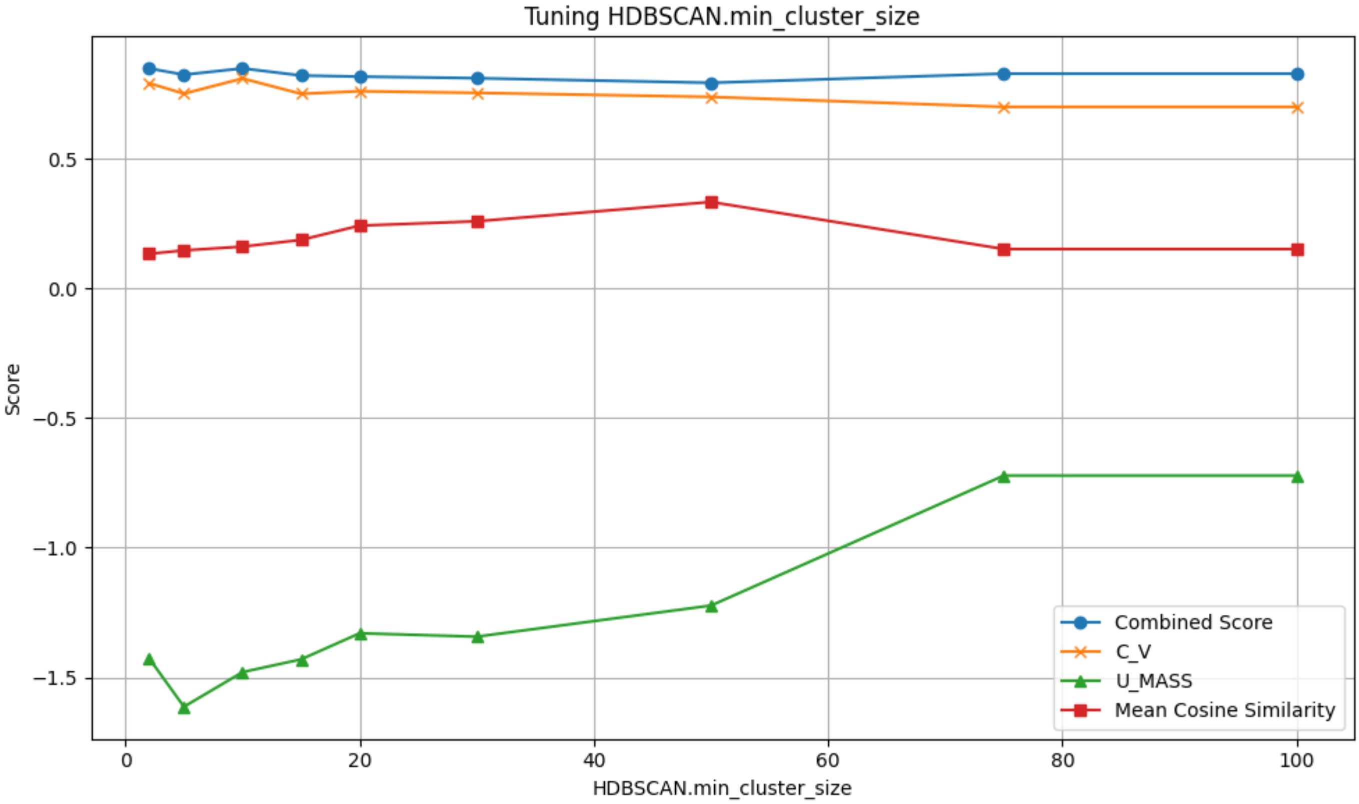


Figure 43: HDBSCAN `min_cluster_size` tuning for 400 tokens

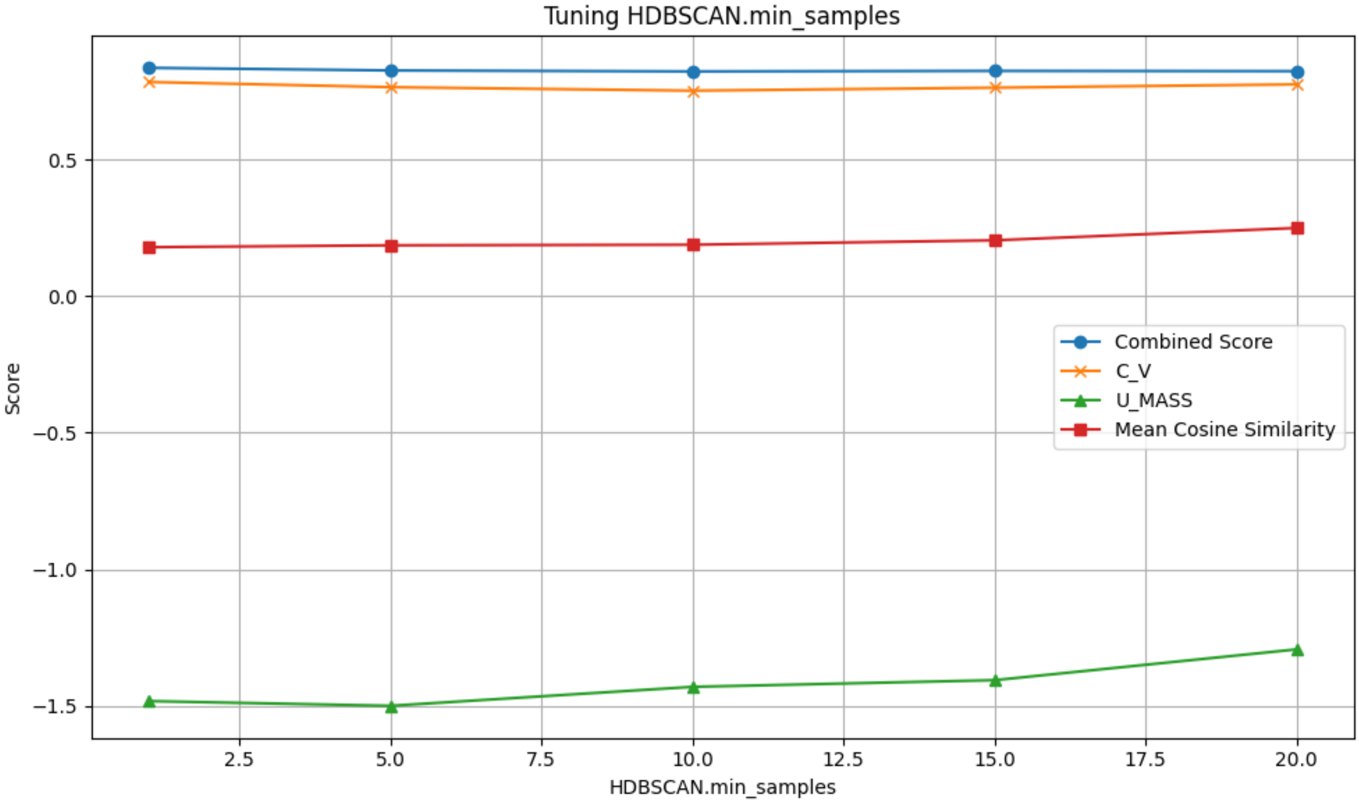


Figure 44: HDBSCAN `min_samples` tuning for 400 tokens

500 Tokens

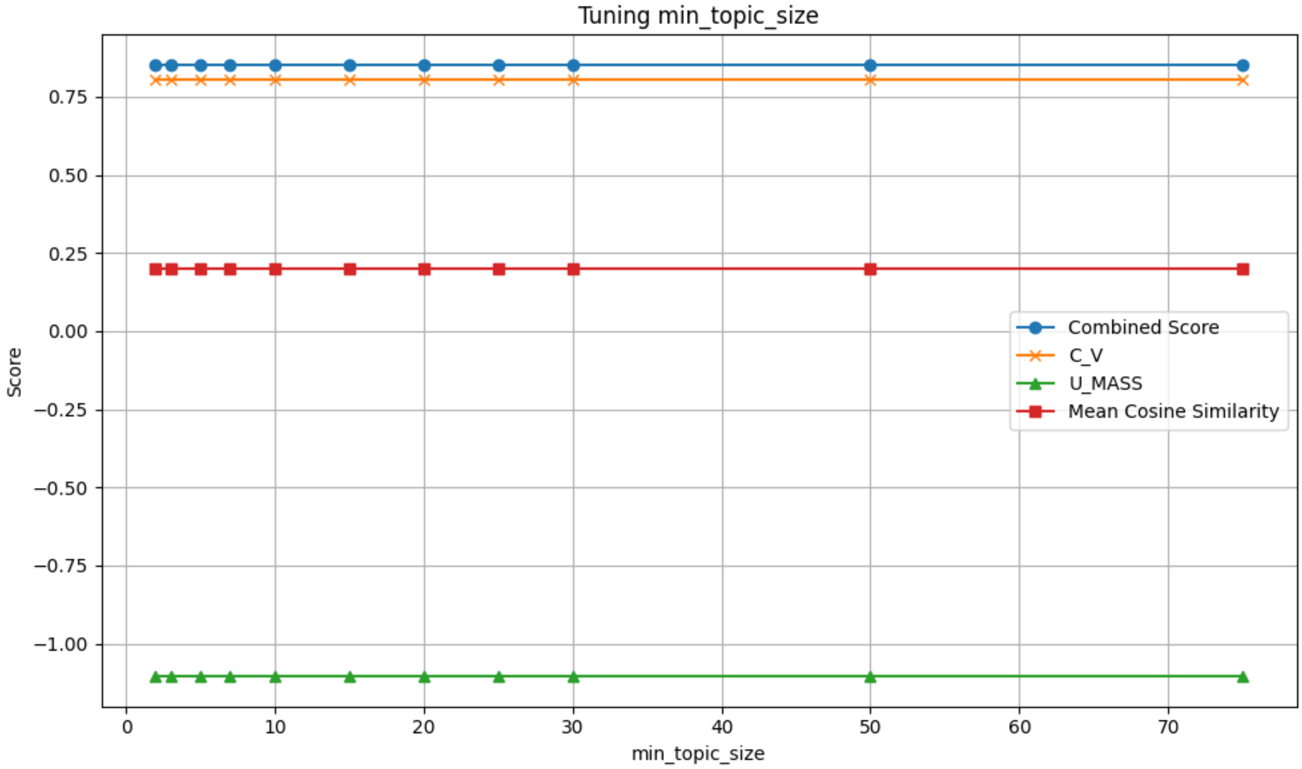


Figure 45: `min_topic_size` tuning for 500 tokens

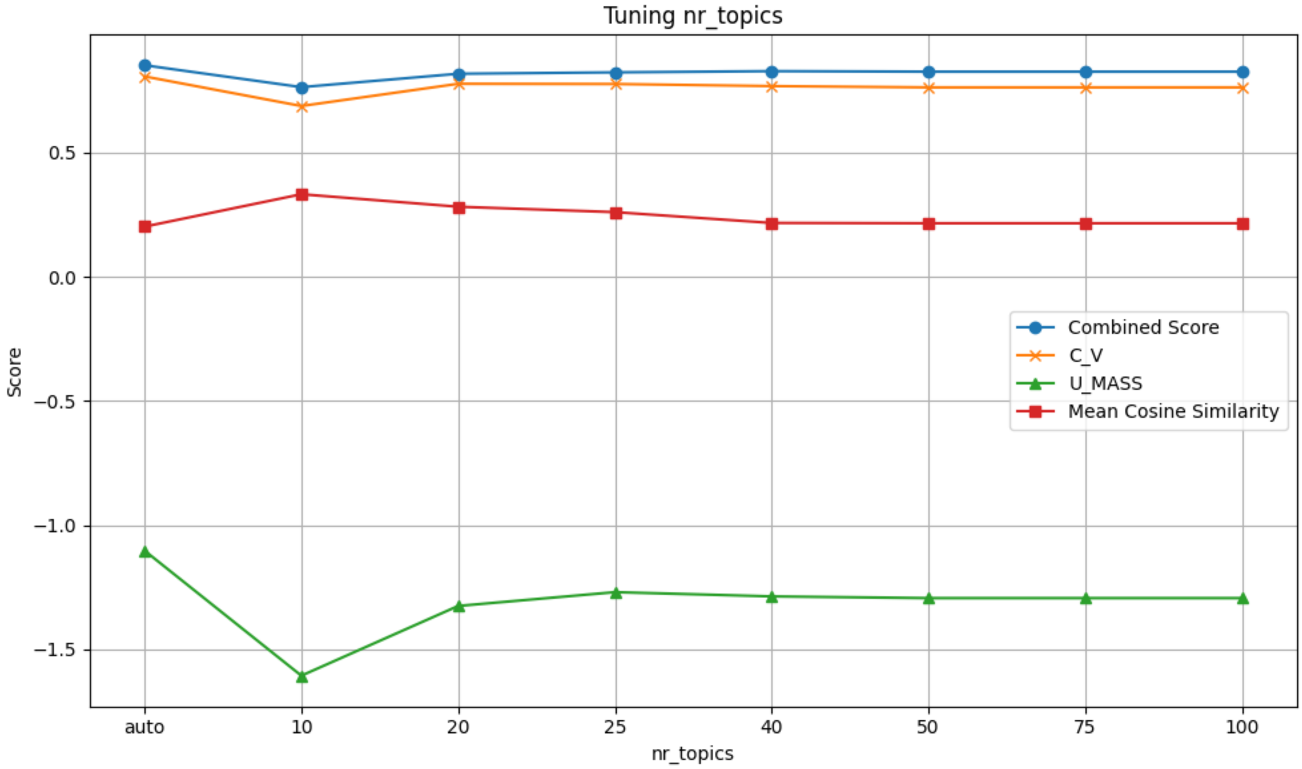


Figure 46: `nr_topics_tuning` tuning for 500 tokens

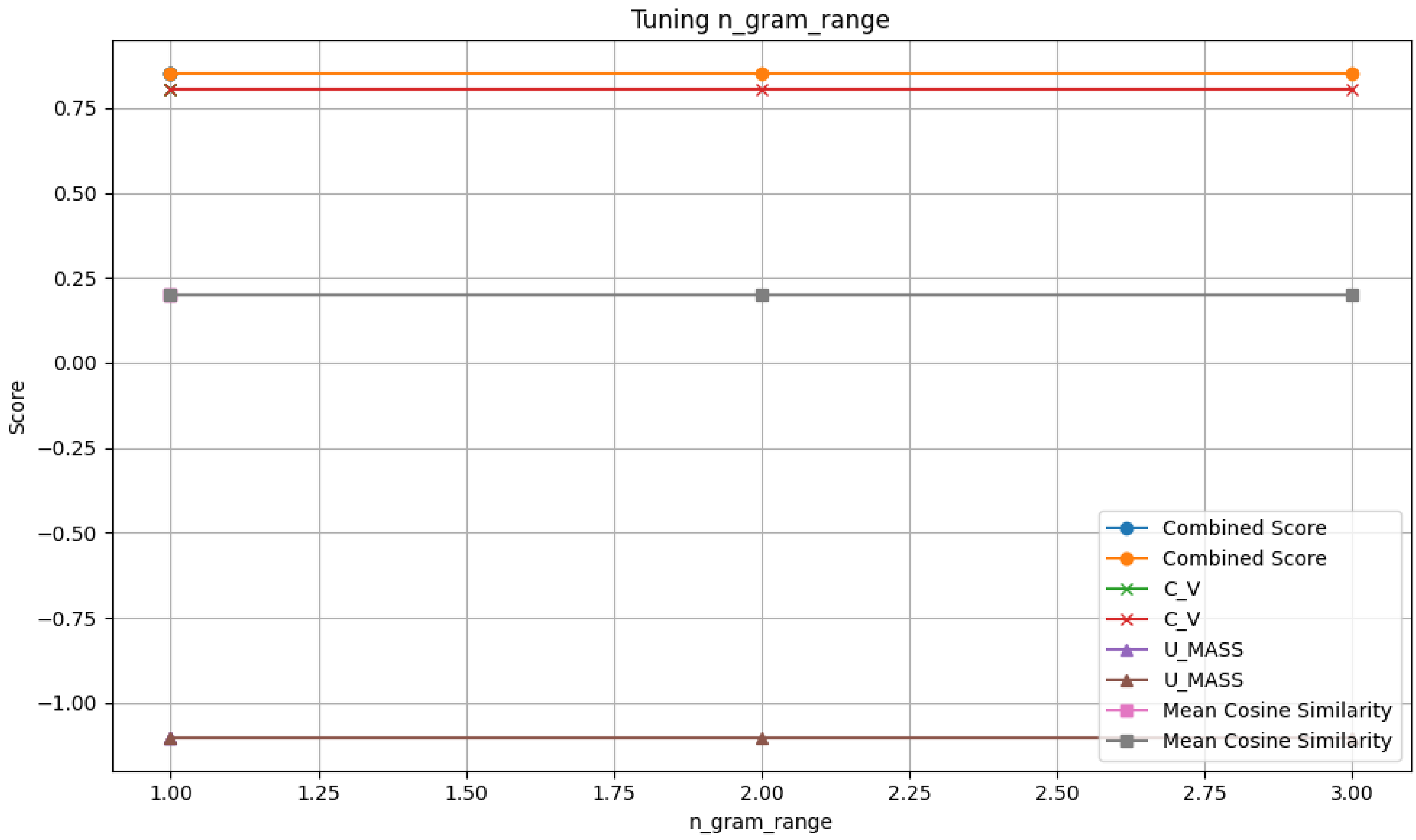


Figure 47: `n_gram_range` tuning for 500 tokens

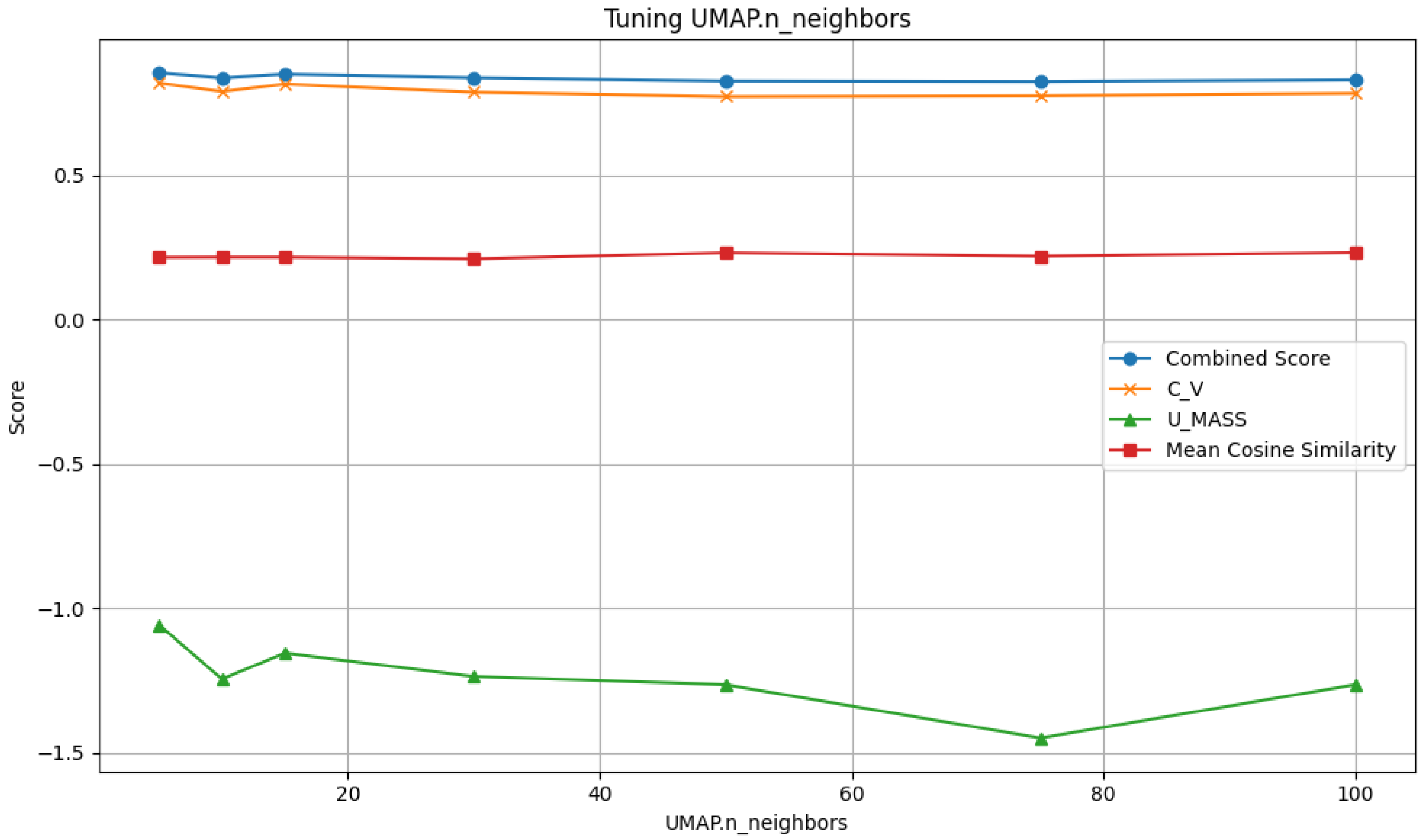


Figure 48: UMAP `n_neighbors` tuning for 500 tokens

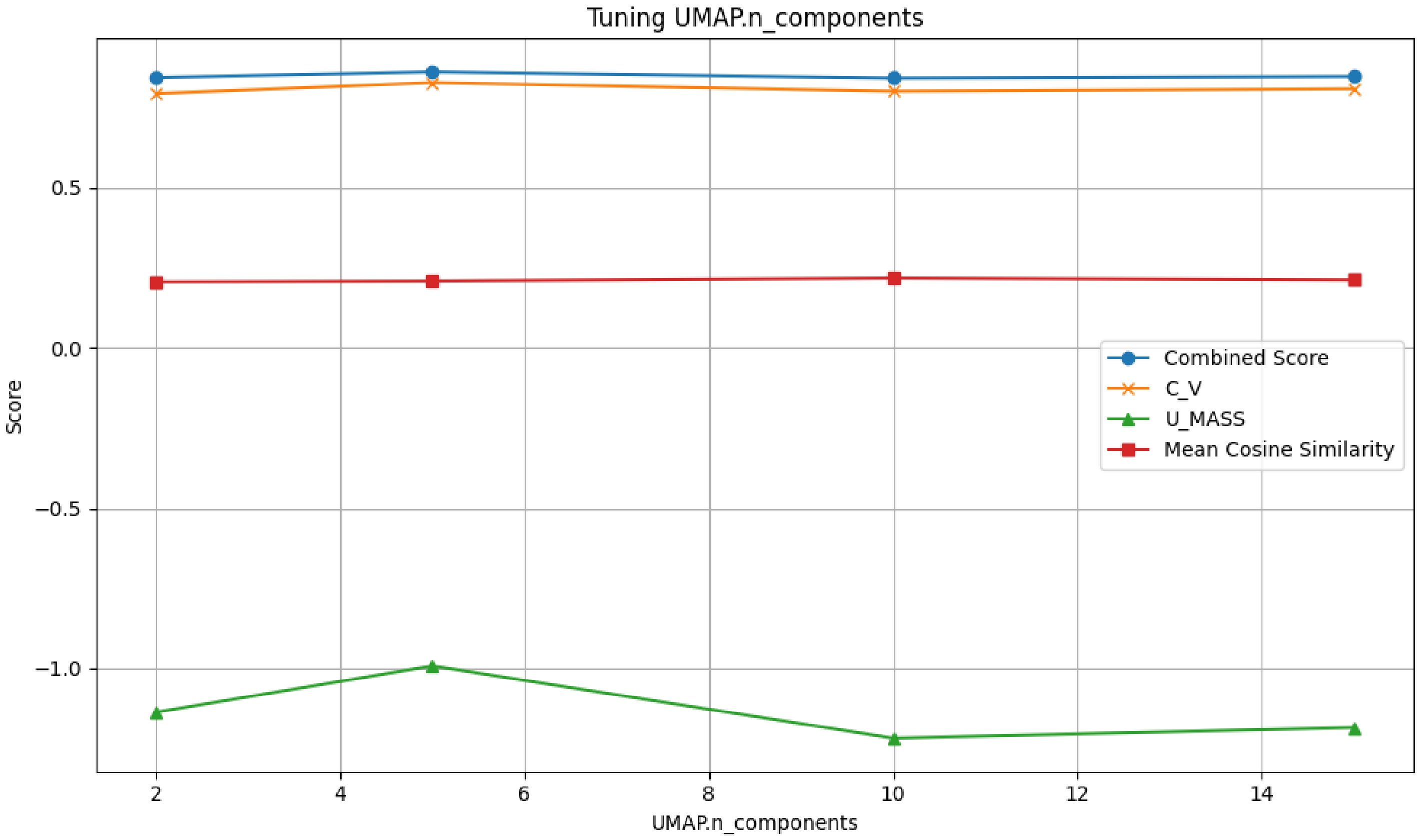


Figure 49: UMAP `n_components` tuning for 500 tokens

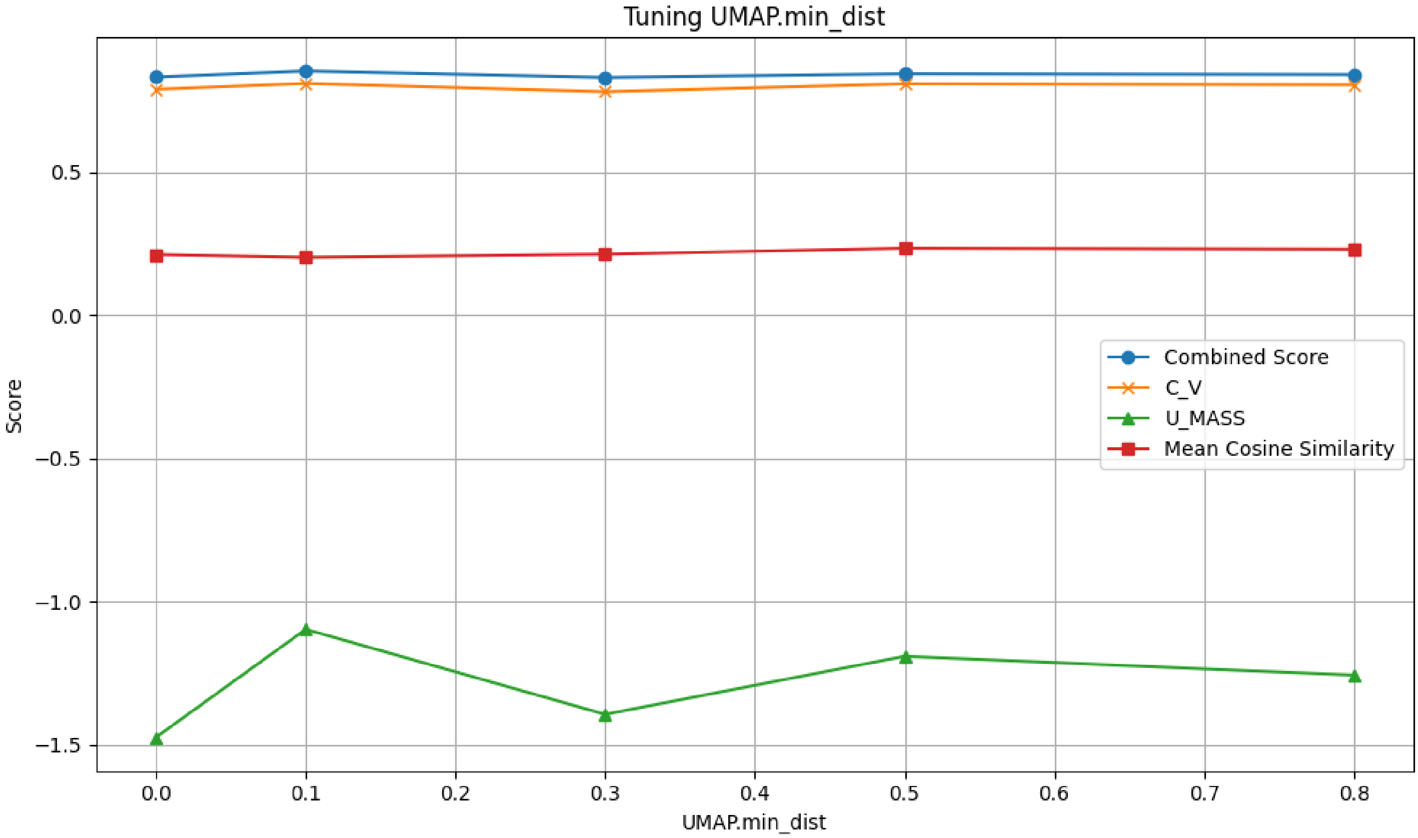


Figure 50: UMAP `min_dist` tuning for 500 tokens

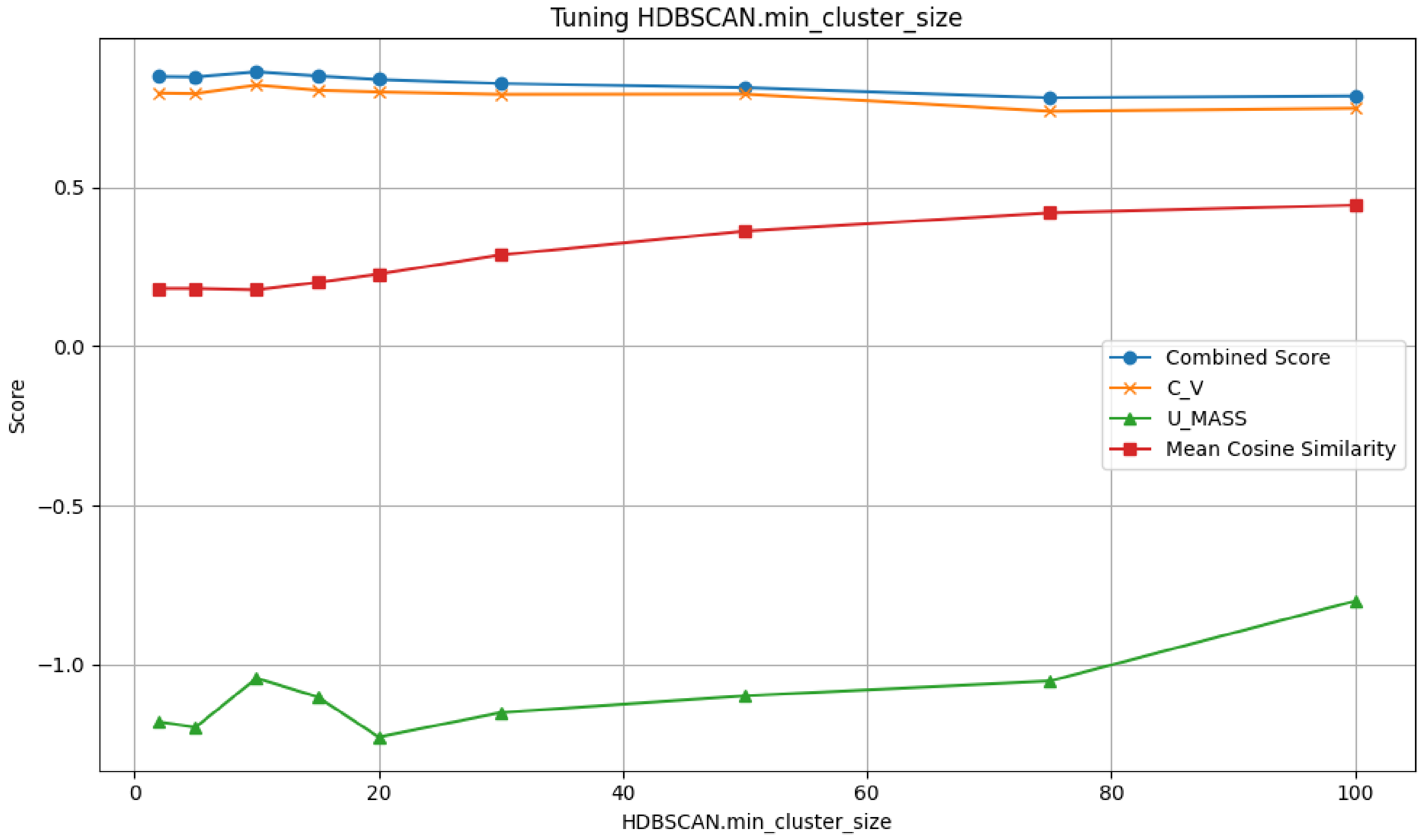


Figure 51: HDBSCAN `min_cluster_size` tuning for 500 tokens

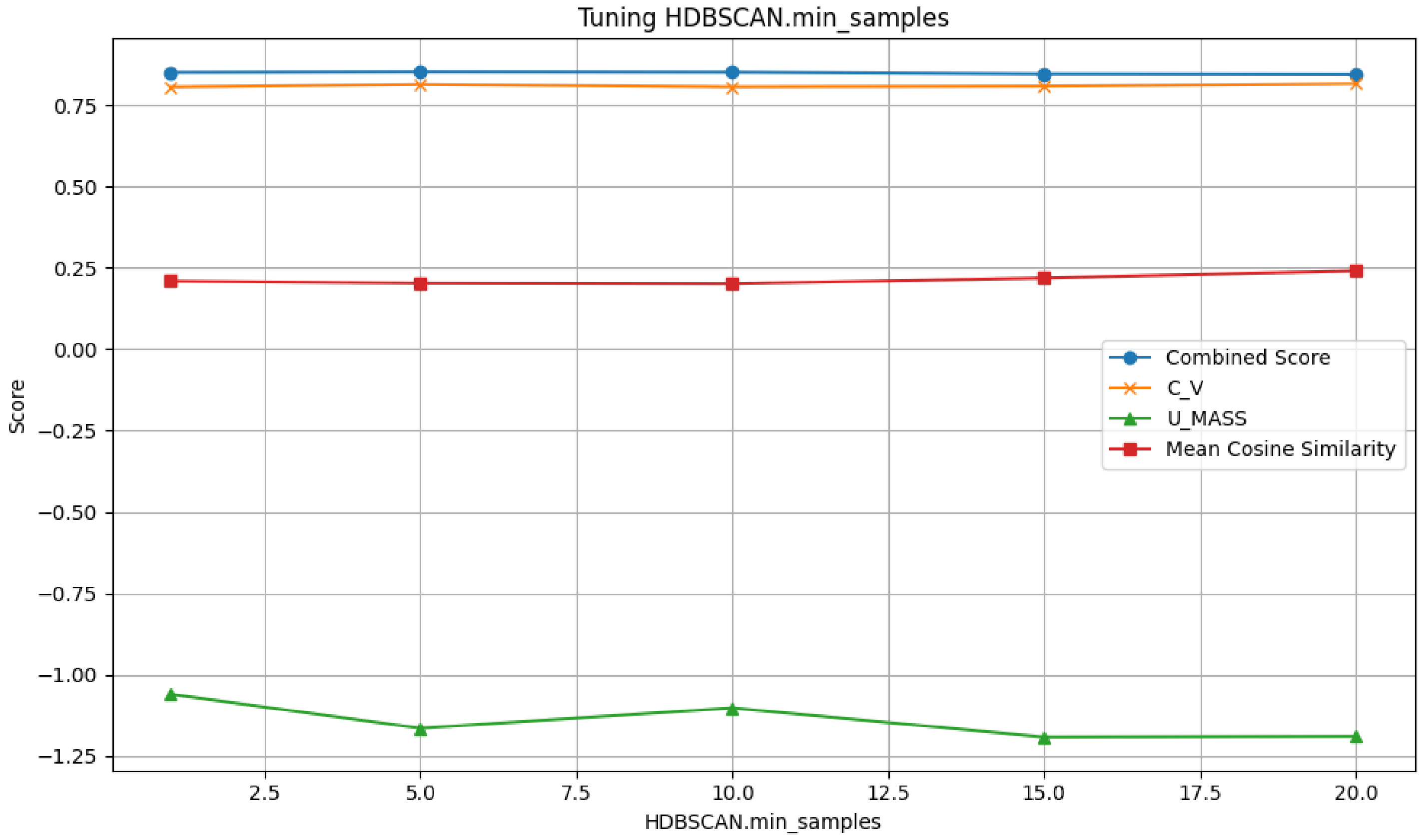


Figure 52: HDBSCAN `min_samples` tuning for 500 tokens

Grid Search - Exhaustive Tuning

	Embed.	Min Topic Size	Nr topics	Min Dist.	Nr Neighbors	Min Cluster Size	Min Samples	CV	UMass	Mean Cosine Sim.	Comb. Score
300	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	15	5	0.8673	-1.2638	0.1497	0.8808
300	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	5	0.8528	-1.5047	0.1286	0.8724
300	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	15	10	0.8475	-1.5985	0.1424	0.8648
500	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	5	0.8238	-1.2072	0.1560	0.8638
400	intfloat/e5-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	10	0.8230	-1.1674	0.1620	0.8634
300	intfloat/e5-base-v2	5-10-15	75	0.1-0.3	5-10-15	15	10	0.8440	-1.5598	0.1480	0.8634
500	all-mpnet-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	10	0.8213	-1.0437	0.1779	0.8631
400	intfloat/e5-base-v2	5-10-15	75	0.1-0.3	5-10-15	10	5	0.8148	-1.3026	0.1372	0.8612
300	intfloat/e5-base-v2	5-10-15	75	0.1-0.3	5-10-15	10	5	0.8265	-1.5108	0.1319	0.8610
400	all-mpnet-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	10	0.8079	-1.2258	0.1602	0.8560
300	intfloat/e5-base-v2	5-10-15	75	0.1-0.3	5-10-15	15	5	0.8235	-1.5265	0.1496	0.8558
500	all-mpnet-base-v2	5-10-15	auto	0.1-0.3	5-10-15	10	5	0.8131	-1.1964	0.1849	0.8540
400	all-miniLM-L6-v2	5-10-15	75	0.1-0.3	5-10-15	10	10	0.8076	-1.2360	0.1701	0.8537

Table 3: Grid Search Results for Topic Modeling Parameters

Zero-shot BERTopic Parameters

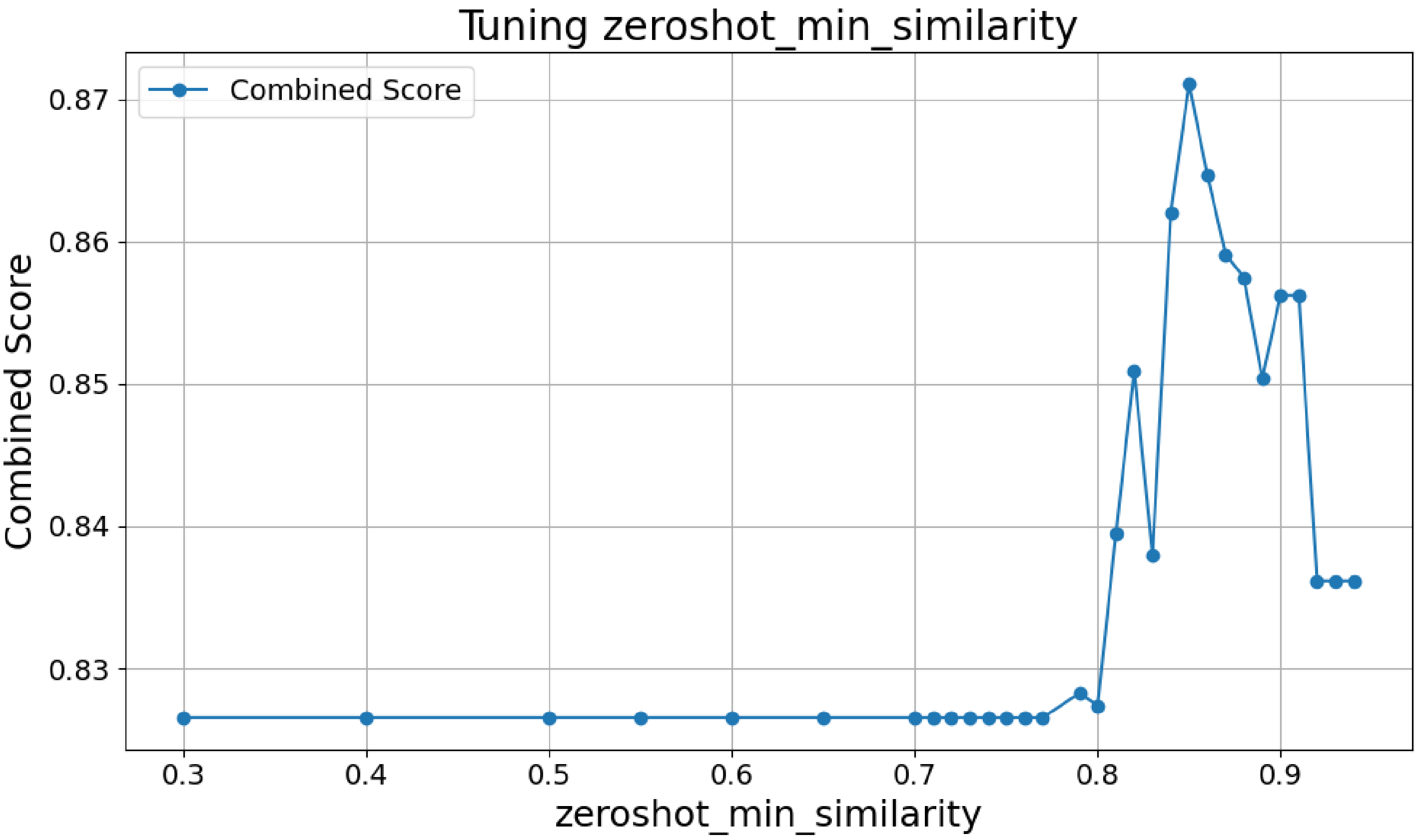


Figure 53: `zeroshot_min_similarity` tuning, using the optimized parameters in Appendix C

C Parameters

Default Parameters

BERTopic Parameters:

```
embedding_model:  all-mpnet-base-v2
   min_topic_size:  5
   nr_topics:  auto
   n_gram_range:  (1, 2)
```

UMAP Parameters:

```
n_neighbors:  15
   n_components:  5
   min_dist:  0.1
   metric:  cosine
   random_state:  42
```

HDBSCAN Parameters:

```
min_cluster_size:  15
   min_samples:  10
```

Tuned Parameters

BERTopic Parameters:

```
embedding_model:  intfloat/e5-base-v2
   min_topic_size:  10
   nr_topics:  auto
   n_gram_range:  (1, 2)
```

UMAP Parameters:

```
n_neighbors:  5
   n_components:  5
   min_dist:  0.3
   metric:  cosine
   random_state:  42
```

HDBSCAN Parameters:

```
min_cluster_size:  15
   min_samples:  5
```

D Software Environment and Dependencies

All tests were conducted on a local development machine equipped with an 8-core AMD Ryzen 7 5800U CPU and 16GB RAM. GPU acceleration (via `torch` $\geq$ `2.2.0`) was available, but was not required for model training and evaluation in this setup. For the sake of reproducibility, a complete accounting of all dependencies and version constraints can be found in Table 4.

The code is implemented in `Python (v3.11)` using the `bertopic (v0.16.0)` framework. The dataset consists of U.S. Congressional Hearing transcripts, which were cleaned and tokenized using `spaCy (v3.7.2)` with the `en_core_web_sm` model, after which they were split into chunks. All related functionality was handled by the `utils/preprocessing.py` file in the codebase. The chunk-level embeddings were generated using `sentence-transformers (v2.6.0)` and cached to disk using the `numpy (v1.26.0)` efficient binary format. Furthermore, the embedding space was reduced via `umap-learn (v0.5.0)`, after which `hdbscan (v0.8.0)` is used for density-based clustering. BERTopic is instantiated with these components and runs in both basic and zero-shot modes, within `models/bertopic.py` and `models/zero-shot.py`, respectively. Model evaluation is conducted through the `gensim (v4.3.0)` coherence score metric and the `scikitlearn (v1.4.0)` cosine similarity measure within `analysis/evaluation.py` and `analysis/visualization.py`. Hyperparameter tuning is implemented in `utils/parameter_tuning.py` and `utils/chunk_eval.py` using first individual tuning, and then a grid search over UMAP, HDBSCAN, and BERTopic-level parameters. Results were logged and plotted using `seaborn (v0.12.2)` and `matplotlib (v3.8.2)` to help in the interpretation. All random components, i.e., UMAP, are seeded with a fixed random state (42) to ensure determinism across repeated runs. Evaluation logs are automatically saved in a structured output directory for inspection.

Package	**Version**	**Purpose**
`bertopic`	$\geq$ 0.16.0	Topic modeling framework
`sentence-transformers`	$\geq$ 2.6.0	Sentence embedding via Transformers
`transformers`	$\geq$ 4.38.0	HuggingFace Transformers backend
`torch`	$\geq$ 2.2.0	Model acceleration and tensor operations
`umap-learn`	$\geq$ 0.5.0	Dimensionality reduction
`hdbscan`	$\geq$ 0.8.0	Density-based clustering
`scikit-learn`	$\geq$ 1.4.0	Machine learning utilities
`gensim`	$\geq$ 4.3.0	Coherence evaluation
`spaCy`	=3.7.2	Tokenization and lemmatization
`en-core-web-sm`	=3.7.0	English language model (via URL install)
`nltk`	$\geq$ 3.8.0	Text preprocessing utilities
`tqdm`	$\geq$ 4.66.0	Progress bar visualization
`pandas`	$\geq$ 2.0.0	Data handling
`numpy`	$\geq$ 1.26.0	Numerical operations
`matplotlib`	$\geq$ 3.8.2	Static plotting
`seaborn`	$\geq$ 0.12.2	Statistical data visualization
`pyyaml`	$\geq$ 6.0.0	Config file handling
`srsly`	=2.4.3	Serialization backend for spaCy
`ujson`	=5.4.0	Fast JSON parsing

Table 4: Software Dependencies for BERTopic Pipeline