Distributed Representations of Topics

by

Dimo Angelov

A thesis
submitted to the University of Ottawa
in partial fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

University of Ottawa

Examining Committee

The following served on the Examining Committee for this thesis.

External Examiner:

Scott Sanner
Professor, Department of Mechanical and Industrial Engineering
University of Toronto

Internal Member(s):

Majid Komeili
Professor, School of Computer Science
Carleton University

Marcel Turcotte
Professor, School of Electrical Engineering and Computer Science
University of Ottawa

Hussein Al Osman
Professor, School of Electrical Engineering and Computer Science
University of Ottawa

Supervisor(s):

Diana Inkpen
Professor, School of Electrical Engineering and Computer Science
University of Ottawa

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University's regulations concerning plagiarism, including those concerning consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

In an era where around 330 million terabytes of data are generated each day, it is crucial to have effective methods for extracting knowledge and structure from this vast amount of information. Topic modeling is a technique for extracting themes, topics, and structure within large data sets which allow for organizing, searching and making sense of the data efficiently. It is a fundamental technique that has a lot of downstream uses in information retrieval, recommender systems, content summarization, content tagging, trend detection, and many others. Some major challenges of topic modeling are finding the right resolution of topics, labeling the topics, segmenting text by topics, evaluating topic model performance, and dealing with topic change over time. The most widely used methods for topic modeling are Latent Dirichlet Allocation and Probabilistic Latent Semantic Analysis. They are probabilistic generative models and, despite their popularity, they have several weaknesses. In order to achieve optimal results, they often require the number of topics to be known. They need custom stop-word lists, stemming, and lemmatization. Lastly, they model topics as distributions over a vocabulary which necessarily make uninformative words the most probable in a topic. Modern neural topic modeling approaches have tackled some of these problems, but none have been able to solve all of them. We introduce distributed representations of topics where topics are vectors in a semantic vector space. We redefine topics to be the most informationally representative of documents rather than representing an underlying distribution over a vocabulary. Our novel topic modeling approach uses document contextual token embeddings. It creates hierarchical topics, finds topic spans within documents, and labels topics with phrases rather than just words. We propose a density-based agglomerative clustering for semantic vector spaces, which is essential for topic hierarchies. Most previous topic modeling evaluation methods focus on topic coherence without evaluating how well topics represent the documents specifically assigned to a topic, leaving a gap in topic model evaluation. To close this gap, we propose the use of BERTScore and topic information gain to evaluate topic coherence and to evaluate how informative topics are of the underlying documents in addition to the existing topic coherence measures.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Diana Inkpen, for her support, insightful guidance, and constant encouragement throughout the course of my PhD. Her mentorship has been vital in shaping both my research and my academic journey.

I am deeply grateful to my parents, Aglika and Dobrin, who have always inspired me to stay curious and keep learning. Their lifelong support, encouragement, and love have been a constant source of strength and inspiration, forming the foundation of every step I've taken toward this achievement.

Special thanks go to Leland McInnes, John Healy, and Colin Weir, who not only inspired me with their ideas and the spirit of research, but were also exceptional mentors in the earliest stages of this work. Their influence was foundational, and their encouragement and insight played a key role in setting me on this path.

To my fiancée, Faye Voight, thank you for being endlessly supportive, incredibly patient, and tolerant of late-night ramblings about obscure topics.

Lastly, I would like to acknowledge Mark Baker, whose persistent question, "When are you going to do your PhD?", served as both a challenge and a motivation. Without that gentle nudge, this journey might never have begun.

Thank you all for being part of this journey.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

There are approximately 330 million terabytes of data created each day[1], and the rate of data generation is only increasing. The volumes of structured and unstructured data being generated globally surpass the capacity of human analysis, making it impossible for individuals or even teams to manually derive insights from such an overwhelming volume of information. This has necessitated the development and use of advanced computational methods and algorithms, including artificial intelligence, to process, analyze, and extract meaningful knowledge from data at scale.

Topic modeling is essential for distilling and comprehending information from large volumes of data. Topic modeling is a technique in machine learning for extracting themes, topics, and structure within large data sets which allow for organizing, searching and making sense of large amounts of information efficiently. It is a fundamental technique that has many downstream uses in information retrieval, recommender systems, content summarization, content tagging, trend detection, and many other use cases.

Advanced topic modeling techniques are essential in the big data era because they allow for unstructured data from all domains to be transformed into structured knowledge improving information retrieval and, therefore, empowering informed decision making processes. As big data proliferates, for all industries, the demand for sophisticated topic modeling methods will only increase, making it a an area of research priority with potential for wide-ranging benefits to the society.

[1] https://explodingtopics.com/blog/data-generated-per-day

The most widely used topic modeling methods are Latent Dirichlet Allocation [17] and Probabilistic Latent Semantic Analysis [50]. They are probabilistic generative models and despite their popularity, they have several weaknesses. In order to achieve optimal results they often require the number of topics to be known. They rely on manual preprocessing like stop-word removal and lemmatization. Lastly, they model topics as distributions over a vocabulary, which necessarily make uninformative words the most probable in a topic. These methods are fundamentally constrained by their simplistic bag-of-words representation, which ignores word order and context. Modern neural topic modeling approaches have tackled some of these problems, but none have been able to solve all of them.

Additionally, most evaluation methods for topic models focus heavily on topic coherence, neglecting how well the topics represent the documents assigned to them. This creates a critical gap in evaluation, which is essential not only for selecting appropriate models, but also for advancing the field by enabling meaningful comparisons of existing approaches.

1.2 What is a topic?

Colloquially, a topic is the theme, matter, or subject of a text or segment thereof. Topics can be broad or narrow in scope; for example, some high-level topic examples are *Politics*, *Science* and *Sports*, and some more granular examples are *Monetary Policy and Inflation*, *Chemotherapy Drugs for Breast Cancer*, and *Basketball Dunking Techniques*. Topics can also be subdivided or aggregated; for example the topics *Physics*, *Chemistry* and *Biology* can be aggregated into the topic *Science*. To complicate matters further, topics can also overlap; for example, the topics *Health* and *Politics* can overlap to create the topic *Healthcare*. These hierarchical and compositional relationships between topics are shown in Figure 1.1.

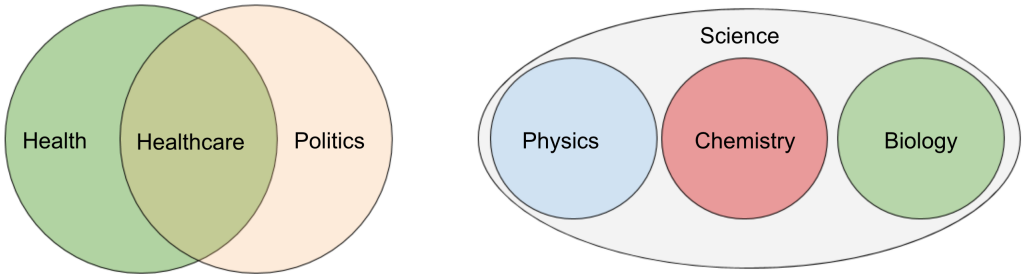


Figure 1.1: This figure shows how topics can be hierarchical, compositional, and overlapping—for example, how *Health* and *Politics* intersect as *Healthcare*, or how subtopics aggregate into broader themes like *Science*.

Figure 1.2: This figure illustrates how topics function as high-level abstractions of information beyond text. Despite visual differences, all images convey the shared conceptual topic of *airplanes* or *flight*, highlighting how topics capture common meaning across varied representations.

A topic is a high-level abstraction of information. Given the images in Figure 1.2, two possible topics of the images are *airplanes* or *flight*. The main observation is that topics represent information provided by the images, although they each contain a different type of airplane, what is common is the abstract concept of an airplane. Topics are not limited to text; they represent the information abstraction encoded in language.

1.3 How to represent a topic?

Topics are often represented as sets over a vocabulary, where each word is associated with a non-negative weight indicating its importance within the topic. In probabilistic models such as Latent Dirichlet Allocation (LDA) and Probabilistic Latent Semantic Analysis (pLSA), these weights define a probability distribution $P(\text{word} \mid \text{topic})$, meaning the weights are real-valued, non-negative, and sum to one across the vocabulary. However, not all topic models enforce this normalization. For example, Non-negative Matrix Factorization (NMF) assigns non-negative weights to words in topics, but these weights are not required to sum to one and are not necessarily interpretable as probabilities.

Although often visualized as lists, topics are more accurately thought of as sets of words that have associated weights. These sets can be ordered by weight to reflect the relative importance of each word. Figure 1.3 illustrates such a visualization, where more informative words appear larger, reflecting their greater contribution to the topic.

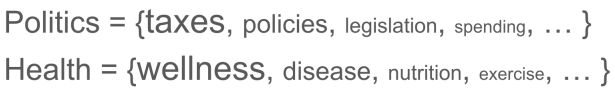


Figure 1.3: This figure depicts a topic as a weighted set of words, where each word contributes to the topic with varying importance. The font size of each word is proportional to its weight, illustrating how more representative or informative words are emphasized visually within the topic distribution.

A good topic representation is one that is informative of the content it represents. Given two example topics from text about global warming with words for Topic 1; *the, and, it, such, they, many, were* and words for Topic 2; *global, climate, warming, change* it is evident that Topic 2 is much more informative of the content it represents, therefore making it a better topic. The ideal topic representation should be one that captures the most informative concepts or ideas that represent the topic.

Informativeness refers to the ability of discriminating between documents given a topic description, by highlighting terms that are both semantically coherent and statistically distinctive within the corpus. In the context of probabilistic topic models such as LDA, this typically translates into topics composed of high-probability terms that are not only frequent in documents associated with the topic, but also rare in documents outside the topic. In other words, informative topic words exhibit high term frequency within the topic documents and low document frequency across the corpus, thereby maximizing their mutual information with the topic label.

For example, while words such as "the" or "and" occur frequently, they offer little discriminatory power because their distribution is fairly uniform across all topics and documents. In contrast, theme-specific terms like "warming" or "climate" are far more useful, as their presence in a document strongly suggests relevance to environmental or a climate-related topic. From an information-theoretic perspective, such words reduce the entropy of document classification under a given topic, thus making the representation more informative.

Ultimately, a good topic representation should not be merely a reflection of term frequency but it should encapsulate the latent semantic themes of the data in a way that is both discriminative and interpretable.

1.4 Traditional Topic Modeling Methods

The most widely used topic modeling method is Latent Dirichlet Allocation (LDA). It is a generative probabilistic model that represents each document as a mixture of topics and represents topics as a distribution of words. Less powerful but related models are Latent Semantic Analysis (LSA) and pLSA.

LDA and pLSA assume a discrete number of topics, t, which is assumed to be known. These methods use bag-of-words (BoW) representations of documents, which largely ignore the semantics and syntax of the underlying text. It is important to note that the BoW model loses sequential context and meaning derived from word order and collocations. For example, the sentences "a dog is hot" and "is a hot dog" yield identical BoW representations, despite having fundamentally different meanings. This limitation arises because BoW treats text as unordered sets of words, disregarding grammar and structure.

These BoW document representations are used to extract the t underlying topics. These topics are then each the distribution of their word probabilities.

The main limitations of these methods are: the number of topics is assumed to be known, the BoW document representations largely ignore semantic and syntactic structure, and they model the topics as word distributions. Since topics are modeled as word distributions, the highest probability words in a topic are usually the most common words in the language, such as *the*, *and*, *it*. These words are called stop-words and often need to be filtered out to extract the informative topic words.

The authors of the LDA paper state: "We refer to the latent multinomial variables in the LDA model as topics, so as to exploit text-oriented intuitions, but we make no

epistemological claims regarding these latent variables beyond their utility in representing probability distributions on sets of words." [17]. The objective of probabilistic generative models like LDA and pLSA is to find topics which can be used to recreate the original document word distributions with minimal error. However, a large proportion of all text contains uninformative words which may not be considered topical. These models do not differentiate between informative and uninformative words as their goal is to simply recreate the document word distributions. Therefore, the high-probability words in topics they find do not necessarily correspond to informative words indicative of a topic in the colloquial sense.

1.5 Topic Modeling Challenges

One of the greatest challenges in topic modeling is choosing the number of topics t that are representative of a dataset. In models like LDA and pLSA, choosing t can alter the quality of the resulting topics and therefore is a very important hyper-parameter of the models. However, this problem is more general; the optimal granularity of topics is subjective even if there was a topic modeling approach that could create high-quality topics for any number of topics t. In addition to the challenges posed by the hierarchical nature of the topics, there is also the challenge of their compositional nature, where topics can overlap. These challenges make it difficult to find a representative set of t topics of a dataset.

Another major challenge in topic modeling is labeling of topics. Topics represented as lists of words are often difficult to interpret and can be ambiguous. Topic models often do not provide context for understanding of the labels.

Identifying spans of topics within documents is also another major challenge. Models like LDA and pLSA use BoW representation which ignores the syntax of the documents, making it difficult to map the topics to spans within the original documents. This makes topic interpretability and explainability difficult.

A fundamental challenge in topic modeling is evaluation. A primary challenge for evaluation is the lack of ground truth, as there are limited labeled datasets to use for the evaluation of topic models. Further, the quality of topics is often subjective to individuals and individual use cases. Since topic modeling is often an unsupervised method, it faces the general challenges of unsupervised model evaluation.

1.6 Distributed Representations

Early approaches to modeling semantic similarity in language, such as LSA and pLSA, aimed to uncover latent semantic structure underlying word usage. These models represent words and documents as dense vectors in a low-dimensional space, where semantic similarity corresponds to spatial proximity. While not implemented as neural networks, such methods introduced the powerful idea of *latent* representations that encode conceptual similarity. Neural network-based methods extend this notion by learning distributed latent representations via backpropagation and large-scale data, enabling more flexible and scalable modeling of language. The key insight in both traditional and neural methods is that distributed latent representations are essential for capturing the conceptual relationships embedded in surface-level word usage.

In neural networks, a *distributed representation* means that each concept learned by the network is represented by many neurons. Each neuron therefore participates in the representation of many concepts. When a neural network's weights are changed to incorporate new knowledge about a concept, the changes affect the knowledge associated with other concepts that are represented by similar patterns [49]. A distributed representation has the advantage of leading to automatic generalization of the concepts learned. Distributed representations are often central to NLP machine learning techniques for learning vector representations of words and documents.

Another key idea behind learning vector representations of words and documents is the *distributional hypothesis*. The essence of the idea is captured by John Rupert Firth who famously said "You shall know a word by the company it keeps" [110]. This statement implies that words with similar meanings are used in similar contexts.

The continuous skip-gram and BoW models [76, 77] known as `word2vec`, introduced *distributed word representations* that capture syntactic and semantic word relationships. The `word2vec` neural network learns word similarity by predicting which adjacent words should be present to a given context word in a sliding window over each document. The learning task of `word2vec` embraces the idea of distributional semantics by learning distributed, latent word vectors for words used in similar contexts. It also learns distributed representation of words, in the form of vectors, which facilitates generalization of word representation. The `word2vec` model generated word vectors produced state-of-the-art results on many linguistics tasks compared to traditional methods [9, 10, 76, 77].

There has been interest in methods of finding distributed word vectors that do not rely on neural networks. It has been shown that the skip-gram version of `word2vec` is implicitly factorizing a word-context pointwise mutual information (PMI) matrix [69], based on this

finding the authors proposed *Shifted Positive* PMI word-context representation of words. This has inspired other methods such as `GloVe` [89], which learn context and word vectors by factorizing a global word-word co-occurrence matrix. Although `word2vec` implicitly factorizes a word-context PMI matrix, what it *explicitly* does is maximize the dot product between word vectors for words which co-occur while minimizing dot product between words which do not co-occur. Additionally, it uses a neural network, which takes advantage of its learned distributed representation of words. This allows the model to learn about all words simultaneously from a single training step on a context word [11]. The ability of `word2vec` word vectors to capture syntactic and semantic regularities of language that other methods try to recreate is a result of the former points, as is its ability to scale to large corpora [76, 79]. As shown in [10], quantitative comparisons between neural and non-neural word vectors show that neural learned vectors consistently perform better. Results from [69, 70] show that, at best, non-neural methods achieve results on certain tasks that are on par with neural methods by replicating hyper-parameters of neural methods like `word2vec`. However, these methods lack the ability to scale to large corpora.

Crucially, the neural training procedure uses back-propagation on each observed target word and context pair to adjust not only the target and context word but all other word vectors, propagating gradient updates across the entire embedding matrix from each single example. In contrast, non-neural factorization approaches derive embeddings in a batch process by decomposing a fixed co-occurrence or PMI matrix, without global parameter updates.

With the goal of overcoming the weaknesses of BoW representations of documents, the *distributed paragraph vector* was proposed with `doc2vec` [65]. This model extends `word2vec` by adding a paragraph vector to the learning task of the neural network. In addition to the context window of words, a paragraph vector is also used to predict which adjacent words should be present. The paragraph vector acts as a memory of the topic of the document; it informs each context window of what information is missing [65]. The `doc2vec` model can learn distributed representations of varying lengths of text, from sentences to documents. The `doc2vec` model outperforms BoW models and produces state-of-the-art results on many linguistics tasks, compared to traditional methods [64, 65].

The self-attention mechanism introduced in the Transformer [106] and later used by BERT [32] led to contextualized token embeddings. Unlike traditional word embeddings that have a single vector representation, contextualized embeddings have different vectors depending on their context. In `doc2vec` which uses `word2vec` for word embeddings, each word has a single static vector representation so the word *apple* will have a single vector whether it is referring the fruit or the company. This was a major limitation of all neural methods with static word embeddings as words can have different meanings depending

on their context. The Transformer also introduced positional embeddings which aid the attention mechanism in figuring out the context of words and allow it to learn embeddings of words that change with the context. Embedding a text sequence with the encoder of a Transformer results in different word embeddings for each word depending on their context allowing it to capture nuanced meanings and semantic relationships within the text, enabling a richer and more accurate representation of language that adapts dynamically to the surrounding words in the sequence.

While the transformer led to BERT which learns contextual token embeddings which better represent language, it does not produce single vector representations of text sequences like `doc2vec` which are optimized for semantic similarity. The single document vectors allowed for semantic comparison of documents using cosine similarity. Although it is possible to compare two sequences of vectors from a BERT model by doing a pairwise comparison of the individual tokens, it is both very expensive, $O(n^2)$, and the model was not explicitly trained for cosine similarity so it is not always appropriate to use in this way. Similarly the '[CLS]' token which capture sentence level information is not directly optimized for semantic similarity and is therefore also not a good choice for single vector representation of a document. This was the motivation behind Sentence-BERT (SBERT) [93] which uses the BERT architecture to train embedding models that ensure that semantically similar documents are close in the embedding space. The sentence embeddings are created by pooling the contextualized token embeddings of the document. The model uses a siamese BERT networks and contrastive learning to align semantically similar documents closer together in the embedding space while pushing dissimilar ones farther apart, optimizing the embeddings for tasks like semantic similarity, clustering, and retrieval with cosine similarity. The SBERT embeddings quickly overtook `doc2vec` for text embedding.

1.7 Distributed Representations of Topics

A *semantic space* is a spatial representation in which distance represents semantic association [41]. A lot of attention has been given to semantic embedding of words; specifically, to distributed word vectors generated by models like `word2vec` which have been shown to capture syntactic and semantic regularities of language [76, 80].

The `doc2vec` model is capable of learning document and word vectors that are jointly embedded in the same space. It has been observed that doing so, or using pre-trained word vectors, improves the quality of the learned document vectors [64]. Similarly, SBERT and other Transformer based models trained for semantic similarity can create embeddings

of words and documents that are jointly embedded. These jointly embedded document and word vectors are learned such that document vectors are close to word vectors which are semantically similar. This property can be used for information retrieval as words or queries can be embedded in the same space as the documents and they can be used to query for similar documents. It can also be used to find which words are most similar to a document, or most representative of a document. As mentioned in [65], the paragraph or document vector acts as a memory of the topic of the document. Thus, the most similar word vectors to a document vector are likely the most representative of the document's topic. This joint document and word embedding is a *semantic embedding*, since distance in the embedded space measures semantic similarity between the documents and words.

In contrast to traditional BoW topic modeling methods, the semantic embedding has the advantage of learning the semantic association between words and documents. We argue that the semantic space itself is a *continuous representation of topics*, in which each point is a different topic best summarized by its nearest words. In the jointly embedded document and word semantic space, a dense area of documents can be interpreted as many documents that have a similar topic. We use this assumption to propose the *topic vector*, which is calculated from dense areas of document vectors. The number of dense areas of documents found in the semantic space is assumed to be the number of prominent topics. The topic vectors are calculated as the centroids of each dense area of document vectors. A dense area is an area of very similar documents, and the centroid, or topic vector, can be thought of as the average document most representative of that area. We leverage the semantic embedding to find the words that are most representative of each topic vector by finding the closest word vectors to each topic vector.

In this thesis, we introduce `Top2Vec`, a model that produces jointly embedded topic, document, and word vectors such that the distance between them represents semantic similarity. Removing stop-words, lemmatization, stemming, and a priori knowledge of the number of topics are not required for `Top2Vec` to learn good topic vectors. This gives `Top2Vec` a major advantage over traditional methods. The topic vector can be used to find similar documents and words can be used to find similar topics. The same vector algebra demonstrated with `word2vec` [76, 77] can be used between the word, document, and topic vectors. The topic vectors allow for topic sizes to be calculated based on each document vector's nearest topic vector. Additionally, topic reduction can be performed on the topic vectors to hierarchically group similar topics and reduce the number of topics discovered.

The greatest difference between `Top2Vec` and probabilistic generative models is how each models a topic. LDA and pLSA model topics as distributions of words, which are used to recreate the original documents' word distributions with minimal error. This often necessitates uninformative words which are not topical to have high probabilities in the

topics since they make up a large proportion of all text. In contrast a `Top2Vec` topic vector in the semantic embedding represents a prominent topic shared among documents, indicated by the dense cluster of documents. The nearest words to a topic vector best describe the topic and its surrounding documents. This is due to the joint document and word embedding learning task, which is to predict which words are most indicative of a document, which necessitates documents, and therefore topic vectors, to be closest to their most informative words.

Unlike LSA's orthogonal bases or LDA's parametric distributions, our method treats the embedding space itself as a continuum of latent topics. Specifically, we identify topics as the centroids of naturally occurring dense clusters of document embeddings in a joint word–document semantic space, automatically discover the number of prominent topics by detecting the number of high-density regions rather than fixing the number of topics a priori. Moreover, our framework is non-parametric and model-agnostic, and can be applied with any pretrained text embedding model enabling the discovery of topics that reflect the specific semantic nuances captured by that model.

Our results show that topics found by `Top2Vec` are significantly more informative and representative of the corpus they were trained on than those found by LDA and pLSA.

1.8 Research Questions

Current topic modeling approaches face several key limitations:

- **Selecting the number of topics:** Determining the optimal number of topics is subjective and often relies on unreliable heuristics.
- **Labeling topics:** Automatically generating meaningful and coherent topic labels remains a challenge.
- **Segmenting topics within documents:** Topic segmentation at the document level is an underdeveloped area.
- **Evaluating topic models:** Existing evaluation methods often emphasize topic coherence while neglecting how well topics represent the underlying documents.

Traditionally, topics have been defined as distributions over a vocabulary, primarily because probabilistic generative models first dominated topic modeling. This representation

emerged from the practical need to generate document word distributions in these models. However, this topic definition is limiting and may not be the best way to conceptualize topics, as it reduces them to statistical artifacts rather than capturing their full complexity and informativeness.

In this thesis, we aim to redefine topics, moving beyond the traditional view of topics as distributions over a vocabulary. Instead, we propose a more general definition that emphasizes how well topics capture the informativeness of their underlying documents. A good topic representation is one that is informative of the content it represents. The ideal topic representation should be one that captures the most informative concepts or ideas that represent the topic. This perspective allows us to leverage distributed representations of text, such as embeddings, for more effective topic modeling.

Our objectives are as follows:

1. **Distributed topic representations:** Develop distributed representation of topics, so that we can leverage distributed representations of text for topic modeling.
2. **Informative and coherent topic labels:** Create topics and labels that are not only meaningful and coherent but also representative of their underlying documents.
3. **Automatic discovery of topic hierarchy:** Explore methods to determine the number of topics present in a corpus automatically and construct topic hierarchies that allow users to explore topics at varying granularities.
4. **Document-level topic segmentation:** Develop novel approaches to segment documents into topic spans.
5. **Comprehensive evaluation of topic models:** Propose novel evaluation metrics that go beyond topic coherence and evaluate how well topics represent the underlying documents.

We formalize the objectives of this thesis through the following research questions (RQs):

- **RQ1**:How effectively can neural embeddings represent topics compared to traditional probabilistic methods?
- **RQ2**: Can embedding-based techniques generate topic labels that are more coherent and informative than those produced traditional probabilistic methods?

- **RQ3**: How can embedding-based representations of documents and topics be used to discover topic hierarchies that support varying levels of granularity, where higher-level topics are both diverse and representative of their subtopics?
- **RQ4**: Can neural embeddings be used to find accurate document-level topic spans?
- **RQ5**: Can metrics that evaluate coherence as well as document representativeness of topics better reflect the quality of topic models than traditional coherence scores?

This thesis aims to contribute novel approaches to topic modeling by integrating neural text representation, addressing key challenges, and redefining how topics are conceptualized, created, and evaluated.

1.9 Contributions

We propose a new topic modeling approach, `Top2Vec` which enables automatic discovery of the number of topics in an embedding space. Our approach produces jointly embedded topic, document, and word vectors such that the distance between them represents semantic similarity. Building on this foundation, we propose Contextual-`Top2Vec` (`C-Top2Vec`), which automatically finds the number of topics in the embedding space of document contextual token embeddings and shows the locations of the topics in the document. Both of our approaches supports hierarchical topic reduction which aids in exploring the optimal topic granularity for downstream use-cases.

We conduct a detailed study of clustering-based approaches for topic modeling, introducing several novel contributions and providing insights into the effectiveness of different clustering techniques and their impact on the overall performance of our method. First, we perform a comparative analysis of clustering algorithms specifically for topic modeling, systematically evaluating their ability to find semantically coherent clusters. Second, we propose a method that utilizes UMAP and HDBSCAN to control the granularity of discovered topics by balancing the preservation of local and global structure in the embedding space. Most notably, we introduce a hierarchical clustering framework that first uses UMAP and HDBSCAN to identify fine-grained, initial topic clusters and to form corresponding topic vectors. Subsequently, Agglomerative Clustering is applied to the UMAP-reduced topic vectors to construct a full topic hierarchy. This combined approach enables flexible control over topic granularity and facilitates the discovery of hierarchical topic structures.

Previous topic modeling methods do not specify where a topic occurs in a document or how relevant a document section is to the topic, they only provide the proportion of a document that belongs to a topic. Our approach segments each document into topic spans allowing for much more granular topic discovery. It also produces a per-topic relevance score for each topic span within a document which allows for ranking the relevance of document segments for a topic.

Previous methods use ranked lists of words to label topics, which can be difficult to interpret. We label our topics with phrases that lead to improved interpretability and topic coherence.

Most previous evaluations focus on topic coherence without evaluating how well topics represent the documents specifically assigned to a topic, leaving a gap in topic model evaluation. We propose Topic Information Gain and using BERTScore [118] to evaluate the coherence of topics and simultaneously evaluate how representative topics are of the topic documents.

Our implementation is available as a Python library at the following link: `https://github.com/ddangelov/Top2Vec`

Parts of this work were published in "Topic Modeling: Contextual Token Embeddings Are All You Need" [7] and "Top2Vec: Distributed Representations of Topics" [6].

1.10 Outline of Thesis

This thesis is divided into eight chapters:

- Chapter 1 introduces the motivation, key challenges, and foundational concepts of topic modeling and the research questions addressed in this thesis.
- Chapter 2 reviews the evolution of topic modeling approaches and evaluation methods.
- Chapter 3 introduces the topic vector and the `Top2Vec` algorithm which performs automatic topic discovery, topic vector calculation, and informative topic labeling.
- Chapter 4 explores various clustering methods for topic modeling, comparing their strengths and limitations.

- Chapter 5 outlines the methods for creating multi-vector representations of documents, creating distributed representations of topics, finding topic spans in documents and labeling topics.
- Chapter 6 presents novel methods for topic model evaluation, including Topic Information Gain and BERTScore, which assess both topic coherence and how well topics represent their underlying documents.
- Chapter 7 evaluates our proposed models and compares them to other topic modeling approaches on several datasets.
- Chapter 8 concludes and discusses future work.

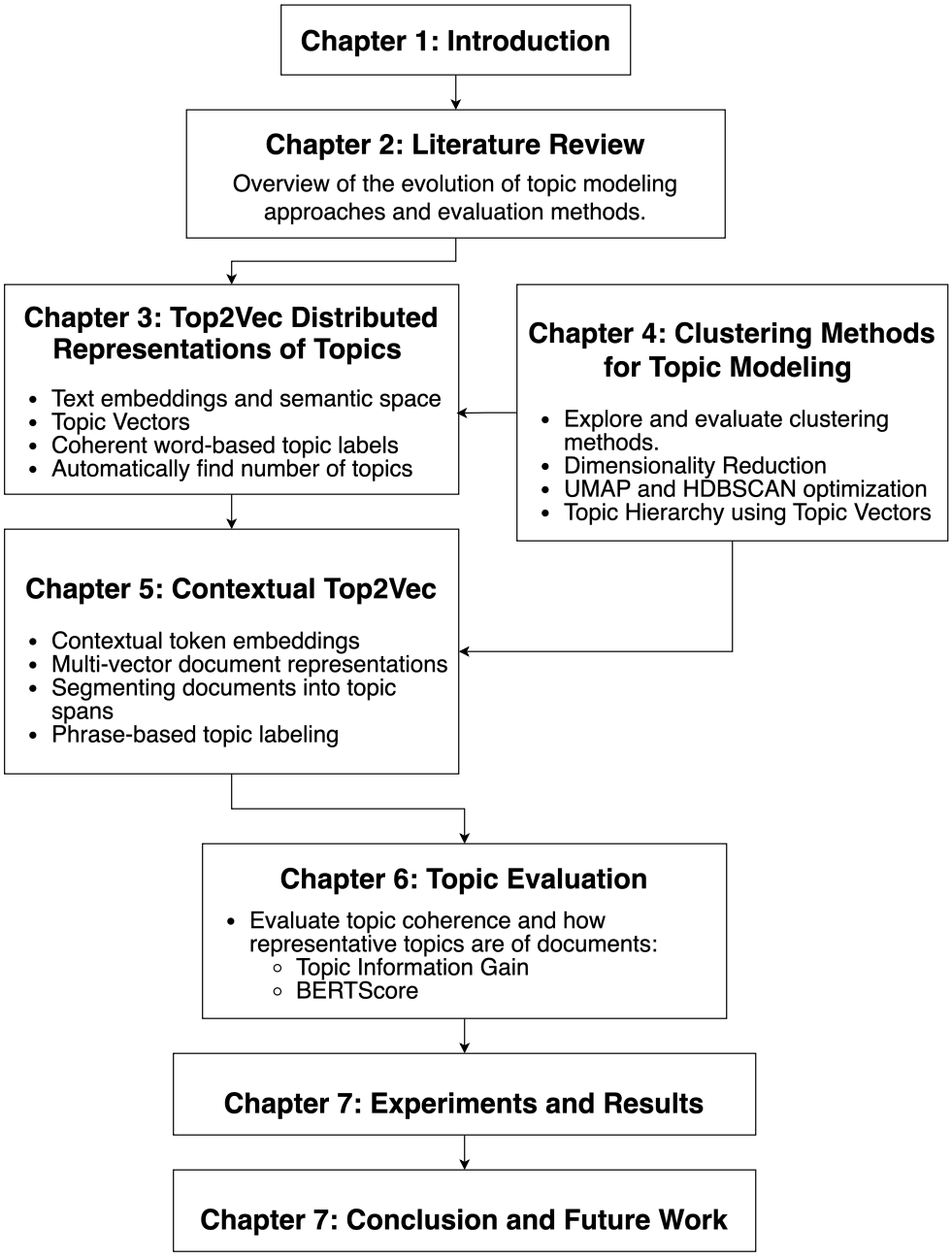


Figure 1.4: This diagram presents the structure of the thesis and the relationships between chapters.

Chapter 2

Literature Review

2.1 Traditional Methods

2.1.1 Early Text Analysis

One of the earliest representations of text was the bag-of-words (BoW) representation, where each document is represented as a vector, for which the dimensionality is the number of unique underlying vocabulary words of the corpus and each word's occurrence in the document is recorded as its count in the respective vector. This can also be referred to as Term-Frequency (TF). Using this representation a text corpus can then be transformed into a term frequency document matrix. These vectors can then be compared with cosine similarity or other metrics for measuring distance between vectors therefore allowing for comparison of documents. The limitations of this representation are that it ignores word syntax, it ignores semantics and it gives equal weight to all words. Since common words outnumber informative words they end up dominating the representation and making unrelated documents similar due to them sharing many common words. These limitations prevent BoW representations from capturing semantic meaning in text effectively.

In order to deal with the unequal distribution of uninformative words in documents, the concept of Inverse-Document-Frequency was introduced by [100], providing a method to emphasize more distinctive terms and make common words have less weight. IDF is calculated as the inverse of the number of documents a word appears across the corpus. Therefore, a word appearing in many documents will have a very small IDF value, while a rare word would have a larger IDF. This led to the TF-IDF representation of documents,

which takes the (BoW) representations of documents in a corpus and then finds the product of TF with the IDF score for each word, which weights each word by its importance, emphasizing terms that are frequent in a specific document but rare across the corpus. This transformation refines the basic BoW model by reducing the influence of common, less-informative words and highlighting distinctive terms. TF-IDF remains a strong baseline for text vector representation due to its simplicity, efficiency, and effectiveness. This representation can be seen as probability weighted information, where TF is the probability and the IDF is the amount of information [3].

2.1.2 Matrix Factorization Techniques

In BoW and TF-IDF models, each term is treated independently, ignoring the semantic relationships between words. This leads to limitations in handling synonymy, where different words convey the same concept, and polysemy, where a single word has multiple meanings depending on the context.

LSA [31] was motivated by the challenges of BoW and TF-IDF document representations, where words with similar meanings are treated as entirely unrelated, and words with multiple meanings have a single representation. LSA is a technique that applies singular value decomposition (SVD) to a term frequency document matrix to uncover the latent semantic structure. This creates a reduced-dimensional representation of the data, where both terms and documents are represented as vectors in a latent semantic space.

The term-document matrix, which captures the frequency or importance of terms across a collection of documents, can be approximated using Singular Value Decomposition (SVD), which factorizes the matrix into three smaller matrices. The first matrix represents the association between terms and a set of latent topics, where each row corresponds to a term and each column to a topic, with entries indicating how strongly a term is associated with each topic. The second matrix is a diagonal matrix containing values that represent the relative importance of each topic in capturing the structure of the original data. The third matrix captures the association between documents and latent topics, where each row corresponds to a document and each column to a topic, with entries reflecting the strength of the document's connection to each topic. The number of latent topics is typically much smaller than the number of terms or documents, allowing for a reduced-dimensional representation that preserves the most significant semantic information. This process, known as LSA, reveals conceptual relationships between terms and documents, enabling similar documents to be grouped based on shared underlying themes, even when they do not use the same vocabulary.

Although LSA deals with synonymy, it only partially addresses polysemy. It conditions word meanings on context but struggles with representing words with multiple distinct meanings, leading to potential distortions as each term is modeled as a single point in the semantic space. Further, SVD can be computationally prohibitive for very large datasets.

Non-Negative Matrix Factorization (NMF) [66,67] is another matrix factorization technique that decomposes a term-document matrix into two lower-dimensional matrices, one representing topic and terms and the other documents and topics. It enforces non-negativity constraints on the elements of these matrices. LSA, which uses SVD and can produce factors with both positive and negative values, which are not easily interpretable. The non-negativity constraint ensures that the resulting components are more intuitive, often aligning with human concepts such as the presence or absence of topics in documents or terms in topics.

Model Design Space

LSA represents documents using a term-document matrix derived from a BoW approach. It applies SVD to reduce this high-dimensional matrix into a lower-dimensional latent semantic space, where topics are represented as dense abstract vectors that do not align directly with the original vocabulary. A linear decoder reconstructs an approximation of the original matrix, allowing topic-word associations to be inferred from the projection. Documents are softly assigned to topics by projecting them into this latent space, with each dimension reflecting the degree of association with a latent topic.

NMF also begins with a BoW-based document-term matrix, but instead of reducing dimensionality, it factorizes the matrix into two non-negative matrices using Non-negative Matrix Factorization. Topics are represented as sparse, non-negative vectors in the original vocabulary space, where each dimension corresponds to a specific word. This alignment makes topic-word relationships explicitly interpretable as weighted word distributions. Documents are softly assigned to topics based on their non-negative mixture of topic vectors, with weights indicating the strength of each topic's presence.

2.2 Probabilistic Techniques

pLSA [50] improves on LSA by adding a probabilistic framework to model term-document co-occurrences as mixtures of latent topics. This makes it more interpretable and flexible

than LSA. It assumes that documents are mixtures of topics and topics are distributions over words. It uses a generative process to model the observed word distribution in a document by first selecting a document, then sampling a topic from the document's topic distribution, and finally sampling a word from the selected topic's word distribution. It captures semantic relationships in text better due to the probabilistic framework and is also much more interpretable.

LDA [17] improves on pLSA by adding a robust generative framework. It can be seen as a Bayesian version of pLSA, as it adds Dirichlet priors to the document topic and word-topic distributions. This adds regularization by smoothing the document-topic and word-topic distributions and preventing them from over-fitting the data. This also allows for generalization to unseen documents, in contrast, pLSA cannot generalize to unseen data because it assigns unique parameters to each document in the training set, making it impossible to infer topic distributions for new documents without retraining.

Another advantage of LDA is the control over the Dirichlet parameters, specifically α (for the document-topic distribution) and β (for the topic-word distribution), which allows for tuning the sparsity or smoothness of these distributions. A smaller α creates documents with fewer topics, while a larger α creates documents that have a balanced mix of topics. Similarly, a smaller β creates topics with a few specific words, while a larger β creates topics with a broader range of words, providing flexibility in modeling the granularity of topics and their representations.

LDA is still one of the most widely used topic modeling approaches. Selecting the optimal number of topics is one of the primary challenges with LDA and most other topic modeling approaches. Additionally, LDA uses BoW representations of documents that ignore word semantics and syntax. LDA models the underlying word distribution of documents, which necessarily makes uninformative words the most probable in topics, leading to poor topic interpretability. The authors of the LDA paper make it clear: "We refer to the latent multinomial variables in the LDA model as topics, so as to exploit text-oriented intuitions, but we make no epistemological claims regarding these latent variables beyond their utility in representing probability distributions on sets of words." [17].

Model Design Space

pLSA models documents using a BoW representation, similar to LSA and NMF, but introduces a probabilistic latent variable model to explain the generation of word occurrences. Each document is treated as a mixture of latent topics, and each topic is defined as a multinomial distribution over vocabulary terms. Word occurrences are modeled by

marginalizing over latent topics, with each word probabilistically associated with a topic given the document context. Unlike LSA and NMF, which rely on matrix factorization techniques, pLSA defines an explicit probabilistic model over word-topic-document triples. However, it does not define a generative process for documents themselves, learning a separate topic distribution for each document in the training set. This limits generalization: unlike LSA and NMF, which compute latent representations via global transformations, pLSA cannot infer topic distributions for unseen documents without retraining.

LDA retains the BoW representation and models topics as multinomial distributions over vocabulary terms. It extends pLSA by introducing a fully generative probabilistic framework, incorporating Dirichlet priors over both document-topic and topic-word distributions. In this admixture model, for each document, a topic distribution is first sampled from a Dirichlet prior; then, for each word in the document, a topic is sampled from this distribution, followed by a word sampled from the topic's word distribution. This contrasts with **pLSA**, where the document-topic distribution is treated as a fixed parameter learned individually for each document in the training set, with no generative mechanism for unseen data. By treating document-topic distributions as latent random variables, LDA enables inference on new documents and improves generalization. The Dirichlet priors also enforce sparsity and smoothness, making the model more robust and statistically well-grounded compared to pLSA.

2.3 Neural Methods

Neural topic models (NTMs) incorporate deep learning techniques to capture the syntax and semantics of text by leveraging word and document embeddings instead of relying solely on statistical inference on BoW representations. Unlike traditional topic models like LDA, which rely on statistical inference over BoW representations, NTMs leverage distributed representations such as word embeddings and document embeddings. These embeddings, derived from neural embeddings models, enable NTMs to capture both syntactic structures and semantic relationships within the text. By incorporating pre-trained embeddings NTMs enable transfer learning by leveraging the linguistic and contextual knowledge of the embedding model which is often trained on a very large datasets, reducing the need to learn everything solely from the target corpus. This allows them to better capture the semantics of the corpus and enhances the interpretability and the quality of extracted topics.

2.3.1 Embedding Based Models

The Embedded Topic Model (ETM) [33] combined the power of generative models with word embeddings. This approach overcomes the limitations of BoW representation of documents and allows for richer semantic representation of words in a document. It embeds both words and topics in the same low-dimensional space, leveraging the semantic information from word embeddings. ETM can learn word embeddings or use pre-trained ones. It models topics as vectors in the embedding space of words. A topic's distribution over words is proportional to the inner product of the topic's embedding and each word's embedding. Documents are modeled as mixtures of topics, with document-topic proportions drawn from a logistic-normal distribution. The model generates words using a log-linear distribution based on the similarity between word and topic embeddings. Although the use of word embeddings is an improvement over LDA, ETM still ignores word syntax and requires the number of topics to be known.

Model Design Space

ETM uses direct word embedding vectors to represent documents. Topics are represented as latent embeddings situated within the same embedding space as words. Topic-word relationships are computed through a neural linear decoder based on the similarity between topic embeddings and word embeddings. Documents receive probabilistic soft assignments to topics through an admixture model, indicating the mixture proportions of topics within documents.

2.3.2 Autoencoder Models

Product-of-Experts Latent Dirichlet Allocation (ProdLDA) [101] replaces the Dirichlet prior over the latent topic distributions with a product of experts framework. This allows for learning a more expressive and flexible prior than the Dirichlet distribution. The experts are parametrized by a neural variational autoencoder (VAE), where the encoder maps input documents to latent topic representations and the decoder reconstructs document-word distributions. The encoder is trained jointly with the rest of the model. The main advantage of ProdLDA is that it creates more distinct and coherent topics. Each topic contributes uniquely to the document representation, reducing redundancy and noise in the topics. In LDA, topics can overlap significantly due to the independence assumptions of the Dirichlet prior, leading to redundant or noisy topics. The product of experts, however, penalizes redundancy by focusing on areas of the latent space where the experts contribute

distinct and complementary information. While ProdLDA improves on LDA it still requires the number of topics to be known, it relies on BoW representation of documents which ignore word syntax and semantics, and it has additional computational overhead.

Contextual Embedding Model (CTM) [14], [15] extends Neural Product-of-Experts LDA (ProdLDA) [101] by adding contextualized document embeddings. CTM uses pre-trained neural vector representation documents using SBERT which allows it to captures word semantics and syntax and benefit from transfer learning. These embeddings represent words in the context of their surrounding text, enabling the model to understand polysemy and synonymy. The VAE encoder takes as input the SBERT document vectors as well as BoW representation of the documents, unlike ProdLDA which only uses BoW. The decoder reconstructs the BoW representation from the latent topic distribution. The addition of the SBERT document embeddings is the key improvement that allow it to outperform ProdLDA as its better able to capture the semantic and syntactic relationships and contextual nuances of the document. The dual-input mechanism allows CTM to better generalize across datasets and adapt to various textual data types, including short texts where traditional topic models struggle due to sparsity. The main limitation of CTM is that its decoder reconstructs the BoW representation, which ignores word syntax, and semantics constraining the expressiveness of the model despite the use of contextualized embeddings in the encoder. An additional limitation is the need to know the number of topics.

Model Design Space

ProdLDA improves upon LDA by replacing the probabilistic graphical model with a neural variational autoencoder framework. It still uses a BoW document representation, but instead of Dirichlet priors and explicit word sampling, it employs a neural encoder to infer a document's topic mixture and a decoder based on the product of experts formulation. Topics are represented as multinomial distributions over words, decoded via a neural softmax layer. Documents are softly assigned to topics through an admixture framework, indicating probabilistic mixtures of topics within documents.

CTM extends ProdLDA by enriching the document representation. Instead of relying solely on BoW, CTM incorporates document embeddings alongside BoW vectors. The encoder uses both to generate a latent document-topic representation. This allows CTM to leverage semantic and syntactic information captured by contextual embedding. The decoder still reconstructs topic-word distributions as multinomial distributions over the vocabulary. As in ProdLDA, topic assignments are soft and probabilistic under an ad-

mixture model, but CTM's use of contextual signals enables it to model polysemy and context-sensitive topics more effectively.

2.3.3 Neural Clustering Models

We present the `Top2Vec` model [6] in Chapter 3, a novel method that leverages joint document and word embeddings to find topic vectors. It uses manifold learning and uses density-based clustering to identify topic regions in the embedded space. It automatically finds the number of topics, it does not require stop-word removal and it also produces hierarchical topics. This approach has been shown to produce more informative and interpretable topics [59], [37], [4]. `Top2Vec` captures word syntax with document embeddings and word semantics with word embeddings. Its main limitation is that each document is assigned only one topic.

The BERTopic model [42] uses BERT [32] document embeddings to find clusters of documents that are labeled with class-based TF-IDF. The main limitations of this model are that it uses BoW representation of document clusters, which ignores word semantics; it does not assign topics to all documents, and it only assigns one topic per document.

Recent interest in neural topic modeling methods has created a rapidly evolving landscape that is difficult to cover exhaustively. Below we highlight additional relevant neural topic modeling approaches that are not detailed overviews. The purpose is to acknowledge the depth of ongoing research and to guide the reader to other relevant methods.

In their work, [121] explore clustering methods for topic modeling and follow a similar approach to BERTopic [42] for topic labeling. They demonstrate competitive topic coherence compared to neural topic modeling approaches. The limitation of their approach is the BoW representation of clusters for topic labeling.

Another notable contribution is from [16] introduce cross-lingual neural topic models that can be trained on one language and applied to another. This paper uses contextual document representations instead of BoW representations of documents, which allows for zero-shot cross-lingual topic modeling.

In their study, [34] propose adding a training objective to maximize topic coherence. They show that their approach increases topic coherence while maintaining a similar level of perplexity as baseline models.

The authors of [35] evaluate several neural topic models and compare them to traditional probabilistic models. Their results show that neural models are better at finding coherent topics and creating representations useful for downstream tasks; however, they

also conclude that traditional models are strong baselines and are sometimes better at modeling the documents.

For further reading, we suggest "A survey on neural topic models: Methods, applications, and challenges" [111] and "Topic modeling algorithms and applications: A survey" [1].

The main advantage common to all the models discussed is that they improve topic coherence compared to LDA. The main disadvantage is that they do not segment topics within a document; instead, they only give distributions of topics present in a document, with the exception of `Top2Vec` and BERTopic which only assign one topic per document. Another common disadvantage is that a topic distribution does not give information about the relevance of documents to a topic, it only gives the proportion of the document that is relevant. This limitation in topic segmentation will be further explored in Chapter 5, which focuses on Contextual-Top2Vec, which finds topic segments within documents.

Model Design Space

Top2Vec introduces a shift from probabilistic topic models by operating directly in a semantic embedding space. Documents are jointly embedded with words using models like Doc2Vec or other text embedding models, forming a unified vector space for documets and words. Topics are modeled as centroids of dense regions in this space, derived from clustering document embeddings. Topic-word relationships are computed by finding words nearest to each topic centroid in the shared embedding space, thereby bypassing explicit probabilistic distributions. Documents are hard-assigned to the nearest topic vector, which simplifies inference but limits multi-topic representations within a single document.

BERTopic modifies `Top2Vec` primarily in its approach to topic representation. Like `Top2Vec`, it represents documents using document embeddings and clusters them to find dense areas, but diverges by discarding the shared embedding space between words and documents. Instead of representing topics as centroids and retrieving nearest neighbor words, BERTopic concatenates the text of all documents within a cluster and constructs a class-based TF-IDF vector to represent the topic. This frequency-based approach does not leverage the semantic structure of the embedding space, resulting in topic-word representations that are often less expressive and less coherent. Topic assignment is hard, with each document assigned to a single cluster, and documents that fall outside dense clusters are excluded entirely, leading to incomplete topic coverage. While BERTopic benefits from contextual embeddings during clustering, its reliance on TF-IDF for topic construction ultimately limits both semantic fidelity and interpretability.

C-Top2Vec further advances these methods by addressing a core limitation of both `Top2Vec` and BERTopic: that a document can only be assigned to a single topic.`C-Top2Vec` segments documents into smaller spans and represents each using contextual embeddings capturing more topical diversity from a document. This allows for finer-grained detection of topics. It retains the centroid-based topic representation used in `Top2Vec` but applies it over sub-document units. Topic-word modeling is still handled via nearest-neighbor search in the embedding space, but the topic assignment shifts from hard document-level labels to fine-grained, token-level assignments, enabling intra-document topic segmentation and better modeling of topic transitions within texts.

2.4 Topic Modeling Problem Space

Topic modeling methods have evolved from simple statistical techniques for document representation such as BoW and TF-IDF into more sophisticated approaches like matrix factorization, probabilistic models, and neural methods. Each category introduces improvements, while also exposing limitations that highlight the areas which need further research. This section outlines this progression along the core design dimensions of document representation, topic representation, topic-word modeling, and topic assignment, as summarized in Table 2.2.

Traditional document representation methods such as BoW and TF-IDF are efficient and useful in representing text but fail to capture syntax or semantics. This limits their ability to handle synonymy and polysemy. Matrix factorization techniques like LSA and NMF attempt to uncover latent semantic structure but still rely on BoW inputs, making them sensitive to high-frequency, low-information words. LSA uses dense latent vectors with soft projections in a reduced space, while NMF improves interpretability by constraining topics and document mixtures to be non-negative, aligning more directly with vocabulary terms. These models are computationally expensive and require the number of topics to be predefined.

Probabilistic models, including pLSA and LDA, offer interpretable frameworks and better generalization through statistical inference. pLSA introduces a probabilistic latent variable model over word-document co-occurrences but fails to define a proper generative model over documents, which limits its generalization. LDA addresses this by introducing Dirichlet priors, resulting in a fully generative admixture model. This allows documents to be composed of multiple topics probabilistically and enables topic inference on unseen documents. Additionally, the Dirichlet priors control the sparsity of topic distributions,

offering greater modeling flexibility. However, their reliance on BoW remains a core limitation, and their topic representations often lack coherence. They also require manual tuning of the number of topics and are prone to generating topics dominated by uninformative terms.

Neural topic models (NTMs) leverage pre-trained embeddings and deep learning to capture semantics and syntax more effectively. Models like ETM, ProdLDA, and CTM improve topic coherence and enable transfer learning. ETM embeds topics and words in the same semantic space, improving the alignment of topic-word relations. ProdLDA uses a neural variational autoencoder with a product-of-experts prior to generate more distinct and less redundant topics. CTM further enhances this by introducing contextual embeddings into the encoder, allowing the model to better understand polysemy and word usage in context. CTM is particularly well-suited for short texts and achieves higher coherence than its predecessors. Despite these advances, NTMs often still rely on BoW representations in the decoder, require the number of topics as input, and introduce greater computational complexity.

Neural clustering models such as `Top2Vec`, which we introduce in Chapter 3, and BERTopic address several of these issues by using document embeddings and automatically discovering the number of topics. `Top2Vec` embeds documents and words in the same space and finds topic centroids using density-based clustering. Topics are labeled via nearest neighbors in the semantic space, bypassing explicit probabilistic distributions. BERTopic similarly uses document embeddings but discards the shared space with words, relying instead on BoW-derived TF-IDF for topic labeling. Both models assign one topic per document and do not model soft or multiple topic mixtures. Additionally, BERTopic does not assign topics to all documents, excluding outliers from the clustering.

A key challenge common to most models is the inability to segment documents into distinct topic regions. Instead, they produce global topic distributions, which lack information about where topics occur within a document. This limitation is especially problematic in long-form texts or multi-theme documents, where understanding where the topics are in the documents is crucial.

In Chapter 5, we introduce `C-Top2Vec` which advances these methods by introducing intra-document topic segmentation. It embeds smaller spans within a document using contextual models token embedding, enabling the detection of topic segments within a single document. It retains `Top2Vec`'s centroid-based topic representation but shifts topic assignment from the document level to token-level assignment. This results in more fine-grained and accurate representations of topic transitions and supports multi-topic documents.

In summary, while neural methods improve coherence and enable contextual under-

standing, they often do not resolve the need for topic segmentation and still rely on simplified document representations. Chapter 3 introduces `Top2Vec`, a neural clustering model that partially addresses these issues by using joint document and word embeddings, automatically discovering the number of topics, and removing the need for stopword filtering. However, it still assigns only one topic per document and does not perform intra-document topic segmentation. These remaining gaps motivate the development of `Contextual-Top2Vec` (Chapter 5), which aims to find coherent topic segments within documents by leveraging contextual embeddings and unsupervised segmentation techniques.

A comparative overview of the methods discussed and their respective strengths and limitations is presented in Table 2.1 and Table 2.2.

Method & Key Models	**Advantages**	**Disadvantages / Limitations**
Document Representation: BoW, TF, TF-IDF	- Simple and efficient - Easy to implement - Strong baseline - Scalable	- Ignores syntax and semantics - Equal weight to all words (TF) - Dominated by common words (TF) - Cannot handle synonymy or polysemy
Matrix Factorization: LSA, NMF	- Captures latent semantic structure - Handles synonymy - More interpretable topics - Dimensionality reduction	- Based on BoW - Poor at handling polysemy - Computationally expensive (SVD) - Requires choosing number of topics
Probabilistic Models: pLSA, LDA	- Probabilistic and interpretable - Captures word co-occurrence patterns - Generalizes to unseen documents (LDA) - Tunable granularity via Dirichlet priors	- Based on BoW - Needs pre-defined number of topics - Topics dominated by uninformative words - Lower coherence and interpretability
Neural Topic Models: ETM, ProdLDA, CTM	- Captures semantics and syntax - Leverages pre-trained embeddings - Better topic coherence - Flexible priors (e.g., ProdLDA, CTM)	- Requires known number of topics - Increased computational cost - Still use BoW representations to some extent

Table 2.1: Summary of major topic modeling paradigms, including traditional, matrix factorization, probabilistic, and neural methods.

Model	Document Representation	Topic Representation	Topic-Word Model	Topic Assignment
LSA	BoW	Latent Vector (dense)	Linear Decoder	Soft (Projection)
NMF	BoW	Latent Vector (vocabulary)	Linear Decoder	Soft (Weights)
pLSA	BoW	Multinomial Topics	Multinomial	Mixture
LDA	BoW	Multinomial Topics	Multinomial	Admixture
ETM	Word Embeddings	Latent Topic Embeddings	Neural Linear Decoder	Admixture
ProdLDA	BoW + Neural Encoder	Multinomial Topics	Neural Softmax	Admixture
CTM	Document Embedding	Multinomial Topics	Neural Decoder	Admixture
BERTopic	Document Embedding	C-TF-IDF Vectors	TF-IDF Labeling	Hard per document
Top2Vec	Document Embedding	Centroids in Embedding Space	Nearest Neighbors	Hard per document
C-Top2Vec	Sub-document embeddings	Centroids in Embedding Space	Nearest Neighbors	Hard per token

Table 2.2: Comparison of topic modeling approaches along core compositional dimensions: document and topic representations, modeling strategies, and document-topic assignment.

2.5 Evaluation Techniques

2.5.1 Perplexity

Perplexity is a widely used evaluation metric for probabilistic generative models and measures how well a model predicts a held-out corpus [17]. It is defined as the exponential of the negative normalized log-likelihood:

$$\text{Perplexity}(\mathcal{D}) = \exp\left(-\frac{\sum_{d\in\mathcal{D}} \log P(d)}{\sum_{d\in\mathcal{D}} N_d}\right),$$

where $\mathcal{D}$ is the held-out dataset, $P(d)$ is the likelihood of document d, and N_d is the number of words in d. Lower perplexity indicates better predictive performance and suggests that the model generalizes well to unseen data.

In topic models such as pLSA and LDA, $P(d)$ is calculated using the topic-word distributions and the document-topic distributions learned during training. These models assume that documents are mixtures of latent topics, and each topic is a distribution over words.

In topic models such as pLSA and LDA, $P(d)$, the probability of a document, is computed based on two key components learned during training: the distribution of topics within each document, and the distribution of words within each topic. These models are built on the assumption that documents are generated as probabilistic mixtures of latent topics, and that each topic is itself characterized by a distinct probability distribution over the vocabulary.

During the generative process, a document is assumed to be composed by repeatedly sampling topics from the document's topic distribution, and then sampling words from the corresponding topic's word distribution. In both models, the likelihood $P(d)$ and therefore perplexity depends on how well the model can reproduce the observed word frequencies in the document. However, because these models operate under the BoW assumption and treat words as conditionally independent given the topic, they capture statistical co-occurrence patterns rather than deeper semantic structure.

Despite its popularity, perplexity has limitations when applied to topic modeling. It is computed purely based on the model's ability to reproduce observed word distributions, relying on the BoW assumption that ignores word order and semantic context. As a result, perplexity evaluates how well the model captures surface-level statistical regularities, not whether the learned topics are semantically coherent, distinct, or informative.

Therefore, while a model with lower perplexity may better predict word occurrences, it may still produce topics that are not meaningful or interpretable to humans [24]. This highlights the disconnect between statistical likelihood and the quality of topic representations. In particular, perplexity does not assess how well the topics reflect the thematic structure of documents or how coherent they are to humans, a gap addressed by alternative metrics such as topic coherence or human-centered evaluations.

2.5.2 Human Judgment

The authors of [24] addressed the gap between perplexity performance and semantically interpretable topics. Although perplexity evaluates a model's ability to predict unseen documents, it does not assess whether the inferred topics are meaningful or interpretable by humans. Additionally, topic models optimized to minimize perplexity may infer topics that are less semantically cohesive or useful for exploratory analysis. They propose two human evaluation methods: the word intrusion task and the topic intrusion task.

The word intrusion task involves asking subjects to identify a word that does not belong in the top words from a topic. For this task a topic from the model is chosen at random, the 5 most probable words of the topic are used and an additional intruder word is selected from low probability words of the topic. The words are then shuffled and presented to a subject. If the words in the topic are semantically coherent, the subject should easily be able to identify the intruder word. This task helps assess whether the inferred topics align with human perceptions of natural word groupings.

The topic intrusion task involves asking a subject to evaluate which one of several topics does not represent the document, of which they read a snippet. The subject is given the

top 3 topics of highest probability and 1 topic randomly chosen from lowest probability topics of the document. If the high probability topics of the document are representative of the document, it should be easy for the subject to identify the topic that does not belong.

In the work of [24], they found that low perplexity is negatively correlated with human judgments. This highlights the limitations of perplexity as a metric for evaluating topic models, where human interpretability is desired. This motivated a greater focus on topic coherence in topic evaluation methods.

2.5.3 Topic Coherence

To address the limitations of perplexity, the authors of [83] proposed a co-occurrence measure based on point-wise mutual information (PMI) over Wikipedia data as measure for evaluating topic coherence. PMI measures the co-occurrence probability of word pairs from a topic within a given corpus, in their case Wikipedia, which serves as a proxy for the interpretive quality of topics. They demonstrated that it correlates well with human judgment, making it a good alternative to perplexity.

Another co-occurrence approach by [81] looks at co-occurrence of words within the documents of the training corpus, to compute a coherence score that leverages conditional probabilities among top topic words. The benefit of this method is that it directly uses the modeled corpus to avoid reliance on external reference datasets and correlates well with human judgment.

The work of [5] proposed a novel method for evaluating topic model coherence by leveraging distributional semantic similarity. Instead of computing a similarity score between word pairs using PMI, as was done by [83], this approach uses PMI and Normalized Pointwise Mutual Information (NPMI) to weight the components of word vectors. In this method, a semantic space is built using co-occurrence data from Wikipedia with a sliding window of 5 words, and each topic word is represented as a vector in this space. By normalizing PMI values to a range between –1 and 1, NPMI overcomes the bias of PMI toward infrequent words and the interpretability issues associated with negative PMI values. The coherence of a topic is then evaluated by measuring the similarity between these weighted topic word vectors using cosine similarity, demonstrating an improvement over the direct pairwise similarity computation approach.

Another similar approach was proposed by the authors of [95], which combines normalized pointwise mutual information (NPMI) with cosine similarity over a sliding window of word co-occurrences. They conclude that a larger window, greater than 50, better captures word context. The measure, C_V, is able to capture semantic coherence effectively

by leveraging the strengths of NPMI for statistical co-occurrence and cosine similarity for vector-based similarity. They demonstrate that C_V has the strongest correlations with human judgments compared to other coherence measures.

The authors of [34] propose word embedding based coherence measures. The first measure evaluates topic coherence by calculating the pairwise inner products of word embeddings for the top topic words. It averages the similarity scores of all word pair combinations within a topic. The second measure uses the centroid of the word embeddings for a topic and computes the inner product of each word's embedding with the centroid. The benefit of this approach is that it leverages the semantic information captured in the word embeddings to provide a computationally efficient method for assessing topic coherence. Unlike traditional measures like NPMI, which require external corpora and are computationally intensive, the word embedding-based measures directly utilize pre-trained embeddings, making them practical and scalable. Lastly, they show strong correlations with human judgments of topic coherence.

2.5.4 Topic Assignment

The Adjusted Rand Index (ARI), proposed by [55], evaluates the similarity between two clusterings by comparing how pairs of elements are grouped in both clusterings. ARI is used to evaluate topic models by comparing the topic assignment of documents from a topic model with ground truth labels. Its values range from -1 to 1, where -1 is complete disagreement, 0 means no better than random and 1 means perfect agreement.

Unlike the regular Rand Index, ARI adjusts for chance, ensuring that the score reflects only meaningful agreement. Treating topics as clusters, ARI's pairwise evaluation captures the coherence of document groupings and adjusts for chance, ensuring robust comparisons. It also accommodates variations in the number of clusters, making it suitable for assessing models with different numbers of topics. Topic models often produce a variable number of topics depending model configuration. ARI accounts for these variations, allowing fair comparison even when the number of topics differs.

Adjusted Mutual Information (AMI), introduced by [108], is another method used to evaluate clustering performance by measuring the agreement between two clusterings. AMI extends Mutual Information by adjusting for chance. AMI values range from 0 to 1, where 0 indicates no mutual information beyond random chance, and 1 represents perfect agreement. AMI is used to evaluate topic models by comparing the topic assignments of documents with ground-truth labels, treating topics as clusters. Unlike unadjusted MI, AMI accounts for differences in the number of clusters and their sizes, ensuring fair

comparisons between models with varying topic counts. This adjustment makes AMI particularly effective for evaluating topic models across diverse configurations.

2.5.5 Existing Challenges in Evaluation

Most current topic evaluation metrics measure topic coherence, but little attention is given to how well the topics represent the underlying documents, which can lead to misleading results [13].

The authors of [36] compared topic model performance using a variety of coherence measures and compared them with human evaluation. They specifically looked at tweets as they are short and often messy text. They found that coherence measures do not always align with human judgment. They use five propositions for their evaluation:

1. If coherence scores are robust, they should correlate.
2. An interpretable topic is one that can be easily labeled.
3. An interpretable topic has high agreement on labels.
4. An interpretable topic is one where the document-collection is easily labeled.
5. An interpretable topic word-set is descriptive of its topic document-collection.

Their main conclusion is that current coherence measures are not sufficient and that measures that incorporate their five propositions are more likely to correlate with human judgment.

The authors of [54] look at the usefulness of traditional coherence metrics like NPMI with sliding window and compare the results with correlation to human evaluation. They do these analysis with traditional topic modeling approaches and neural topic modeling approaches, including ETM [33] and Dirichlet-VAE [19]. They conclude that NPMI often fails to align with human evaluations and that the metric may not always be appropriate for NTMs.

They also find that preprocessing of data used for topic models greatly affects results and leads to difficulty in repeatability when different comparisons do not use the same preprocessing. They also find that evaluation metrics do not always work equally well in different domains of data.

Overall the work of [54] highlights the shortcomings in the evaluation of topic models, particularly the reliance on automated coherence metrics like NPMI. They find that these metrics often fail to correlate with human judgments of topic quality. Lastly, variability in preprocessing, evaluation datasets, and metric implementation across studies undermine the reproducibility of results. They conclude that automated coherence metrics are insufficient for robust model evaluation and advocate for standardized preprocessing pipelines and the inclusion of human evaluation and domain specific evaluations to better reflect real world use cases of topic modeling.

In Chapter 6, we explore novel approaches to topic evaluation that address the shortcomings of existing methods. Specifically, we introduce Topic Information Gain, which quantifies how well topics represent their underlying documents using mutual information, and BERTScore, which leverages contextualized embeddings to capture semantic similarity between topics and documents. These methods provide a more comprehensive evaluation framework that goes beyond traditional coherence metrics by assessing both topic coherence and representativeness of topic documents. Through these approaches, our goal is to improve topic model evaluation.

2.6 Dimensionality Reduction for Topic Modeling

High-dimensional representations, such as those produced by neural embedding models like SBERT, capture rich semantic information about text. However, these high-dimensional spaces pose challenges for clustering algorithms due to the *curse of dimensionality*, where distances between points become less meaningful and neighborhoods increasingly sparse [2, 12, 48, 62]. Dimensionality reduction techniques help by projecting embeddings into a lower-dimensional space that retains the most salient structure of the data.

Early methods like Principal Component Analysis (PCA) [53,87] and Multi-Dimensional Scaling (MDS) [63] are effective for capturing global variance or preserving pairwise distances. However, these approaches often fail to retain the non-linear structures present in neural embeddings, which are essential for discovering meaningful clusters in textual data.

To better preserve local neighborhood structures, non-linear techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) [105] and Uniform Manifold Approximation and Projection (UMAP) [47,75] have gained popularity. t-SNE is particularly effective for preserving local structure, while UMAP offers a compelling trade-off between local and global structure preservation, along with improved scalability to large datasets [116].

In the context of topic modeling, dimensionality reduction plays a crucial role in finding low-dimensional representation of vectors which can be effectively used for clustering. UMAP and t-SNE, haves proven well-suited for downstream clustering and are widely used as a pre-processing step in clustering high-dimensional data. In chapter 4 we will explore dimensionality reduction methods in more detail and discuss their benefits and drawbacks when used in topic modeling.

2.7 Clustering Approaches for Topic Modeling

In clustering-based topic modeling, particularly in models that use neural embeddings to represent documents the clustering algorithm is central to the performance of the topic model. The goal is to identify regions in the semantic space that correspond to coherent topics. A wide range of clustering algorithms can be used for this purpose, each with different assumptions and trade-offs.

Centroid-based methods, such as K-Means [72], are commonly used due to their simplicity and scalability. They partition the embedding space into K fixed clusters, where each cluster is defined by a centroid. While efficient, K-Means assumes spherical cluster shapes and equal cluster sizes, which may not align with the true distribution of topics in heterogeneous corpora [8, 56].

Density-based approaches, including Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [38] and its hierarchical extension, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [21, 74], offer more flexibility by discovering clusters of arbitrary shape and density. These methods do not require specifying the number of clusters in advance and can identify outliers, making them well-suited for topic modeling. HDBSCAN, in particular, is more robust to varying cluster densities and produces more coherent topic clusters from high-dimensional embeddings.

Hierarchical clustering methods, such as Agglomerative Clustering [109], build nested cluster trees that support multi-resolution topic modeling. These methods are especially useful for exploring topic hierarchies, where broader thematic categories subsume more granular subtopics. While computationally more intensive, hierarchical clustering provides valuable structure for applications requiring topic summarization or navigation.

Recent research has also explored **deep clustering methods**, which integrate representation learning with clustering objectives. Techniques such as Deep Embedded Clustering (DEC) [112], Deep Clustering Networks (DCN) [114], and Dip-based Deep Embedded

Clustering with k-Estimation (DipDECK) [68] optimize embeddings specifically for clustering tasks, offering potential gains in topic separability and coherence. Additionally, Sparse Autoencoders [73, 91, 103] have been proposed to disentangle dense representations into interpretable latent dimensions, enhancing the transparency of topic models.

Collectively, these clustering techniques form the basis for distributed topic modeling methods, where the semantic organization of the embedding space is leveraged to discover latent topics without relying on traditional probabilistic generative assumptions. We will explore the use of these algorithms for topic modeling and discuss their benefits and drawbacks in Chapter 4.

2.8 Summary

This chapter provided an overview of the evolution of topic modeling and its evaluation methods, exploring the progression from early text analysis techniques to traditional, probabilistic, and modern neural approaches. It also examined the development of evaluation techniques, from human evaluation to automated metrics. They key ideas discussed were:

- **Traditional Methods**: Early techniques like BoW and TF-IDF laid the groundwork for text representation, emphasizing simplicity and efficiency. Despite their limitations in capturing semantic and syntactic relationships, they remain strong baselines. Matrix factorization techniques such as LSA and NMF improved upon BoW by uncovering latent structures in text, however they have computational and interpretative challenges.
- **Probabilistic Techniques**: pLSA introduced a generative framework, further enhanced by LDA, which added Bayesian priors for better generalization and regularization. Despite their strengths, these models are limited by their reliance on BoW representations of text, challenges in determining the optimal number of topics, and topic interpretability.
- **Neural Methods**: Recent advances in neural topic modeling, including ETM, autoencoder-based approaches like ProdLDA and CTM, and clustering models like `Top2Vec` and BERTopic, address many limitations of traditional models. By incorporating pre-trained embeddings and leveraging deep learning architectures, these methods capture richer semantic and syntactic relationships, improving topic coherence and interpretability.

- **Evaluation Techniques**: We outlined evaluation metrics for topic models, including perplexity, human judgment tasks (e.g., word intrusion, topic intrusion), topic coherence measures (e.g., PMI, NPMI, C_V), and clustering-based ARI and AMI. While automated metrics are useful for assessing model performance, studies highlight their limitations in aligning with human judgment and domain-specific requirements. There is an ongoing need for more robust, standardized, and interpretable evaluation methods as well as metrics that go beyond coherence and evaluate how representative topic are of underlying documents.

- **Dimensionality Reduction**: Dimensionality reduction techniques such as UMAP and t-SNE enable effective clustering by projecting high-dimensional neural embeddings into lower-dimensional spaces. These methods preserve local and global semantic structure, making them crucial for discovering coherent topic clusters.

- **Clustering Approaches**: Clustering algorithms like K-Means, HDBSCAN, and hierarchical or deep clustering techniques cab be used to identify regions in semantic space corresponding to latent topics. Each method has distinct assumptions and trade-offs, influencing the interpretability, granularity, and coherence of discovered topics.

In the next chapter, we will outline the methodology for `Top2Vec`. We will detail the steps for automatically finding the number of topics in a corpus, generating topic vectors and assigning coherent labels to topics.

Chapter 3

Top2Vec: Distributed Representations of Topics

3.1 Introduction

In this chapter, we will introduce `Top2Vec` [6] and describe how we automatically discover the number of topics in a corpus, how we create distributed representations of topics, and how we find informative labels for topics.

`Top2Vec` is a topic modeling approach that leverages distributed representations of text. It finds topic vectors based on density in the document vector space without requiring prior knowledge of the number of topics in the corpus. It leverages joint embedding of topic vectors and word vectors to find informative topic labels without the need for stop-word removal, stemming or lematization. This chapter addresses the following thesis objectives:

- **Distributed topic representations**: `Top2Vec` generates topic vectors in a continuous vector space, allowing for meaningful comparisons and manipulations of topic embeddings. This directly addresses RQ1 by demonstrating how neural embeddings can be used to represent topics.
- **Informative and coherent topic labels**: By leveraging the joint embedding of topic and word vectors we can find the most informative terms to represent a topic. This supports RQ2, as it ensures that the generated topic labels are coherent and informative for the underlying documents.

This chapter provides a high-level overview of the `Top2Vec` approach, discussing its core mechanisms and its advantages over traditional topic modeling techniques. It will introduce fundamental aspects of our neural clustering-based topic modeling approach such as document embedding, clustering, finding topic vectors and labeling topics. A high-level overview of the `Top2Vec` algorithm is summarized in Algorithm 3.1.

In Chapter 4 we will go deeper into the clustering techniques, examining the effectiveness and advantages of different clustering approaches. In Chapter 5 we will explore multi-document vector representation, topic segmentation within documents and phrase-based topic labeling.

3.2 Topic Vectors

The central premise of `Top2Vec` is that the embedding space generated by a text embedding model is a semantic space, where proximity between document vectors reflects semantic similarity [6]. In such a space, high-density regions correspond to groups of semantically related documents, suggesting a shared underlying topic. We define a *topic vector* as the centroid of a dense region, representing the most semantically central point among the documents in that cluster. This formulation enables an unsupervised approach to topic discovery, where topics emerge naturally from the distribution of documents in the embedding space. By identifying these topic vectors, we can effectively discretize the semantic space, characterize the content of a corpus, and assign documents to topics based on proximity.

3.2.1 Semantic Space

The vector space learned by models like SBERT that use contrastive learning to make similar documents have nearby vectors to each other and be distant from dissimilar documents give rise to a semantic space [41]. We argue that this semantic space is a *continuous representation of topics*, where every position in the semantic space represents a given topic. Figure 3.1 shows an example of a semantic space.

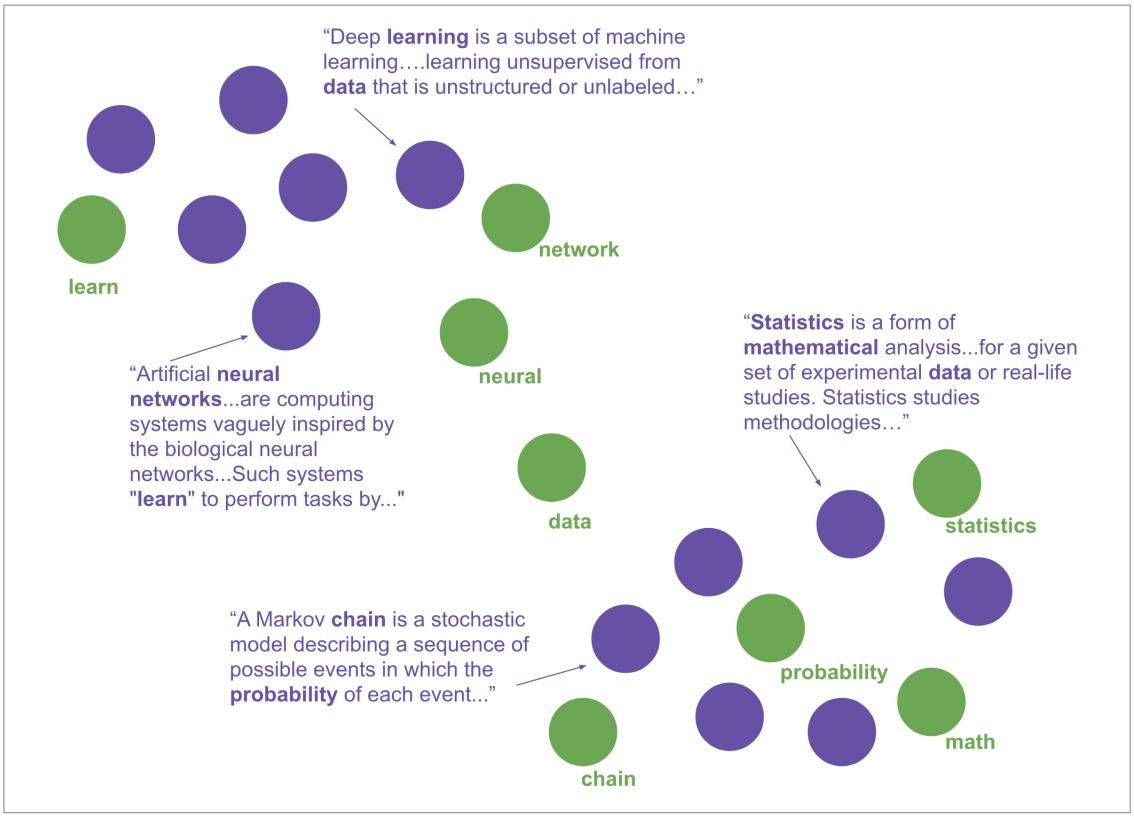


Figure 3.1: An example of a semantic space where the purple points are documents and the green points are words. Words are closest to documents they best represent and similar documents are close together.

Given a corpus of documents that is embedded in a semantic space, like that generated by an SBERT model, the goal of topic modeling is to identify the topics of the documents. Given a single vector per document we could say that the topic vector of each document is just its position in the semantic space and is equivalent to its document vector. We could then leverage a joint embedding of vocabulary and label each document with the nearest word vectors as its topic labels. The problem with this approach is that it does not tell us anything about the relationship between documents and their shared topics, or how topics are distributed across documents. What we really want is to find common topics across documents so that we can describe the corpus.

Given our assumption of a semantic space and our desire to find topics common across the corpus, we leverage the information encoded in the density of document vectors. We

therefore make the assumption that a dense areas of documents represents an underlying common topic across those documents. Based on this assumption, we propose the *topic vector*, which is the centroid of a dense area of documents. The significance of the topic vector is that its the point in the semantic space that is most similar to all the documents, allowing us to characterize their common topic. We can then leverage the embedding space to find the vocabulary that is most representative of that topic vector. Further, once we have topic vectors, we have effectively discretized the topic space for that corpus, and we can calculate topic sizes by assigning documents to their nearest topic vector.

3.2.2 Embedding Models

The semantic space is essential to finding topic vectors, as it enables the identification of dense regions of documents used to calculate topic vectors. The quality of the semantic space is determined by the embedding model used, as it dictates where documents are placed in the vector space and therefore how well similar documents cluster together.

There are several ways to create an embedding space that has the desired properties of a semantic space. One is to train a model specifically on the data so that it learns the nuances of the language of the corpus. One such approach is to leverage the `doc2vec` model [65], which learns joint document and word embeddings where semantically similar documents and words are close together. The benefit of this approach is that, if the corpus has a lot of nuanced language, it will learn how to distinguish between very similar documents within that corpus. The drawback of `doc2vec` is that it can only learn a single vector per word and therefore cannot contextualize the meaning of words, limiting the nuances of language it can learn in the document vectors. We will discuss contextual embeddings in more detail in Chapter 5.

Another major drawback of training an embedding model on just the corpus which is being topic modeled is that there is limited context and information to learn from, and therefore the embedding model may not capture the full meaning of the texts. A small, domain-specific corpus may lack the textual variety needed for an embedding model to learn nuanced language patterns. Therefore, another approach is to use a pretrained embedding model which has been trained on large and diverse corpora. Pretrained models like SBERT [93] capture rich contextual information and generate embeddings that generalize well across different domains.

SBERT models are trained on large-scale natural language inference datasets which allow the model to learn diverse sentence structures and semantic relationships. This

comprehensive training enables SBERT to effectively capture semantic similarity making it well-suited for general-purpose sentence embeddings.

Although SBERT is one of the most popular choices for document embeddings, a range of other models exist. Other alternatives include the *Universal Sentence Encoder* (USE) [23, 115], *Sentence-T5* [84], including multilingual embedding models and countless others. In general, these encoders are pretrained on large natural language inference (NLI), semantic textual similarity (STS), or web corpora. Empirical results show that using a contrastive, triplet, or ranking loss to directly optimize cosine similarity during training is far more important for embedding quality on cosine-based tasks than which encoder architecture the model uses [39, 84, 93]. Any model, whether an RNN or a Transformer, fine-tuned with a loss that pulls semantically similar texts together in cosine space will yield embeddings useful as a semantic space.

However, while pretrained embeddings offer strong generalization, they may not perform as well on a highly specialized corpus that is not reflected in the embedding model's training data. In such cases, domain adaptation or fine-tuning on the target corpus can help bridge the gap between general language understanding and domain-specific semantics.

The ideal approach would be to use a pretrained embedding model like SBERT and then fine-tune it on the corpus being topic-modeled. This would leverage the knowledge from diverse training corpora while also learning the specifics of the target corpus, balancing generalization and domain-specific knowledge. However, SBERT models are typically fine-tuned using contrastive learning objectives, such as triplet loss, which require labeled pairs of sentences. It is not trivial to select these pairs in an unsupervised manner. Even if such an approach is found, there is no guarantee that the learned representations will align with meaningful semantic differences in the corpus. Further, without human-annotated similarity scores or some ground truth labels, there is no reference metric to compare the embeddings against to know if the fine-tuning has improved the labels. Therefore, we opt to use pretrained SBERT models, which generally perform well across a wide range of corpora, and leave the challenge of fine-tuning as a direction for future work.

3.2.3 Dimensionality Reduction

Finding dense areas of vectors in high dimensions, such as those from an SBERT model, is a challenge due to the curse of dimensionality, which makes finding density computationally expensive and less effective. We therefore need to use dimensionality reduction to find a low-dimensional representation of the vectors where we can apply density-based clustering. There are many options for dimensionality reduction, with the prominent choices be-

ing Principal Component Analysis (PCA) [53,87], Multi-Dimensional Scaling (MDS) [63], t-Distributed Stochastic Neighbor Embedding [105] and UMAP [75]. We explore dimensionality reduction in more detail in Chapter 4.

UMAP is a good choice for dimensionality reduction because it preserves both local and global structure of the data. Unlike t-SNE, which focuses primarily on preserving local structure, UMAP constructs a topological representation of the data. This enables it to maintain detailed relationships within clusters while preserving the overall layout of the dataset. Additionally, UMAP scales efficiently to large datasets, outperforming t-SNE in terms of computational speed and memory requirements, which is critical when working with large volumes of high-dimensional embeddings like those from SBERT models [75,116]. Lastly, PCA is not an optimal choice, as it is a simpler method that only preserves linear relationships in the data by maximizing the variance captured in the principal components. We go into more detail of dimensionality reduction in the next chapter.

We use UMAP [75] to find a low-dimensional representation of the vectors. We use UMAP since it preserves the local and global structure of the vectors, allowing us to find dense areas in their low-dimensional representation. Figure 3.2 shows UMAP-reduced document vectors; it can be seen that a lot of the global and local structure is preserved in the embedding.

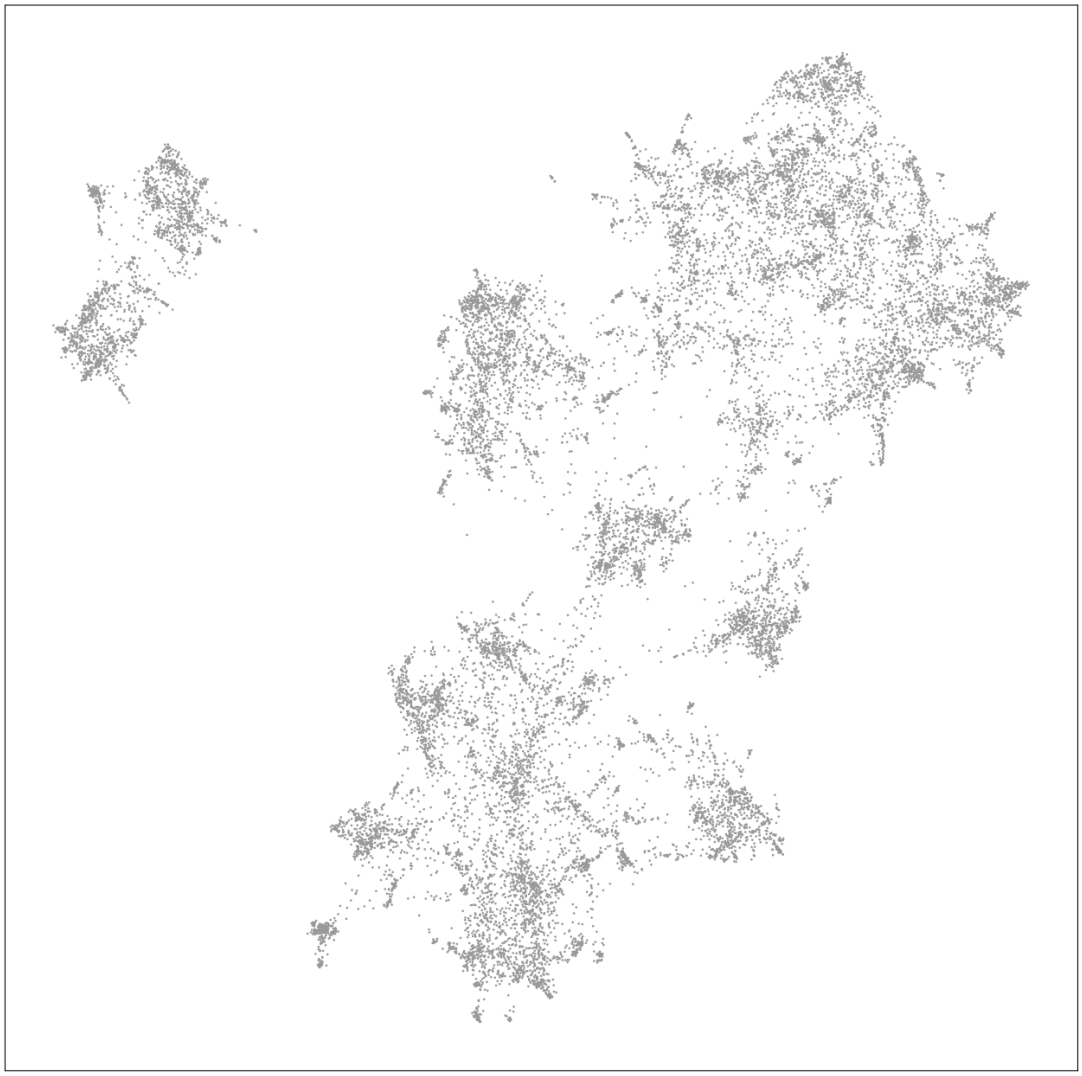


Figure 3.2: A 2D projection of 768-dimensional SBERT document vectors from the *20newsgroups* dataset using UMAP. The visualization preserves both local and global structure, revealing dense clusters and semantic relationships among documents in a lower-dimensional space.

3.2.4 Density-Based Clustering

Once we have UMAP-reduced vectors we can apply density-based clustering to identify dense areas of vectors. We use HDBSCAN [21] on the low-dimensional vectors to find dense areas. We use HDBSCAN since it can find clusters of varying densities, unlike DBSCAN. Additionally, it does not require the number of clusters to be specified. This allows us to find the natural number of clusters present in the data which allows us to automatically find the upper bound of the number of topics present in the dataset. We explore clustering in more detail in Chapter 4.

Figure 3.3 shows an example of dense areas of documents identified by HDBSCAN. It can be seen that HDBSCAN identifies clusters of varying, sizes, shapes and densities.

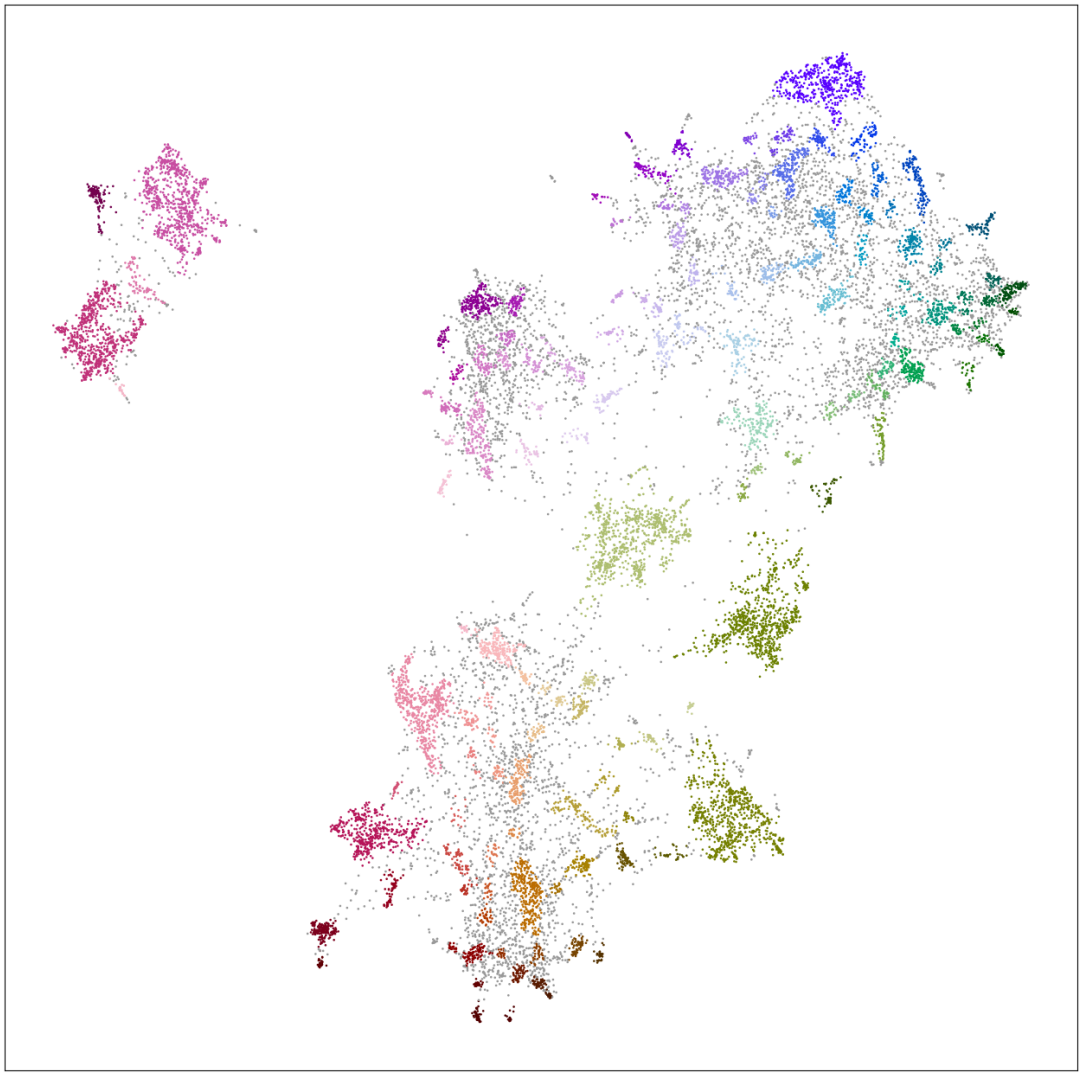


Figure 3.3: UMAP-reduced SBERT document vectors from the *20newsgroups* dataset, clustered using HDBSCAN. Each colored region represents a dense cluster of semantically similar documents, while grey points denote noise—documents not assigned to any cluster.

3.2.5 Calculating Topic Vectors

The dense clusters of documents and noise documents identified by HDBSCAN in the UMAP-reduced dimension, correspond to locations in the original semantic embedding

space. The use of UMAP and HDBSCAN can be seen as a process which labels each document in the semantic embedding space with either a noise label or a label for the dense cluster to which it belongs.

Given labels for each cluster of dense documents in the semantic embedding space, topic vectors can be calculated. There are a number of ways that the topic vector can be calculated from the document vectors. The simplest method is to calculate the centroid, i.e. the arithmetic mean of all the document vectors in the same dense cluster. There are other reasonable options such as the geometric mean or using probabilities from the confidence of clusters created by HDBSCAN. We experimented with these techniques and found that they resulted in very similar topic vectors, with almost identical nearest-neighbour word vectors. We speculate that this is mainly due to the sparsity of the high dimensional space. Therefore, we decided to use the simple method of calculating the centroid. Figure 3.4 shows a visual example of a topic vector being calculated from a dense area of documents.

The arithmetic mean of the document vectors captures the principal direction and location of the topic manifold, averaging out deviations along the finer cluster boundaries. Further, downstream nearest-neighbour retrieval and cosine similarity computations depend primarily on angular relationships, which are predominantly affected by the centroid's direction rather than the exact geometry of the cluster. Thus, even for elongated or non-spherical clusters, the centroid provides a robust and computationally efficient proxy for the underlying thematic direction in the semantic space.

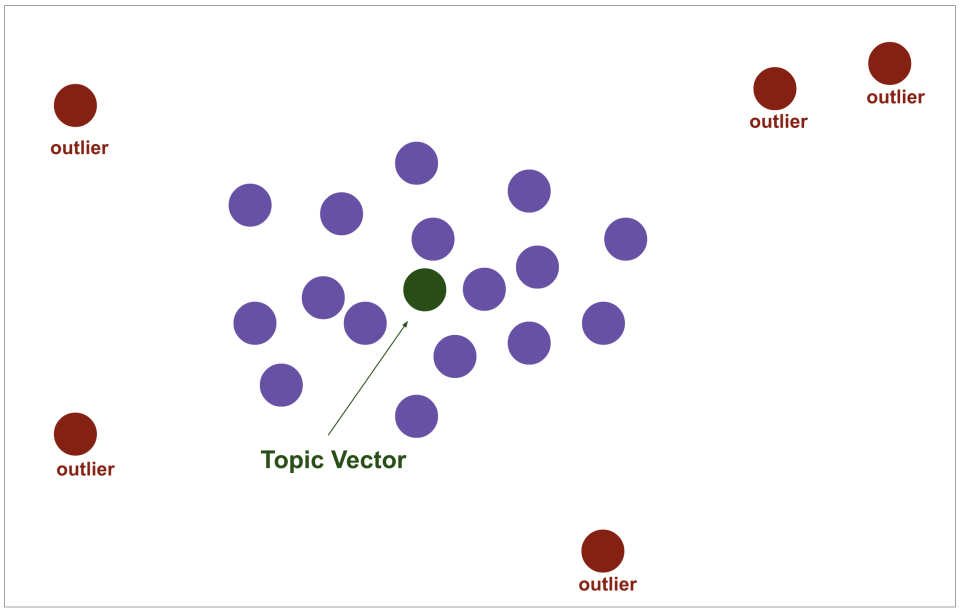


Figure 3.4: Illustration of a topic vector computed as the centroid of a dense cluster of documents identified by HDBSCAN (purple points). Outliers, which are points not in a dense area detected by HDBSCAN, are excluded from the centroid calculation. This method provides a simple yet effective way to represent the topic as a single point in the semantic space.

The centroid is calculated for each set of document vectors that belong do a dense cluster, generating a topic vector for each dense cluster. The number of dense areas found is the number of prominent topics identified in the corpus.

3.3 Find Topic Words

In the semantic space, every point represents a topic that is best described semantically by its nearest word vectors. Therefore the word vectors that are closest to a topic vector are those that are most representative of it semantically. The distance of each word vector to the topic vector will indicate how semantically similar the word is to the topic. The words closest to the topic vector can be seen as the words that are most similar to all documents into the dense area, as the topic vector is the centroid of that area. These words can be used to summarize the common topic of the documents in the dense area. Figure 3.5 shows an example of a topic vector and the nearest words.

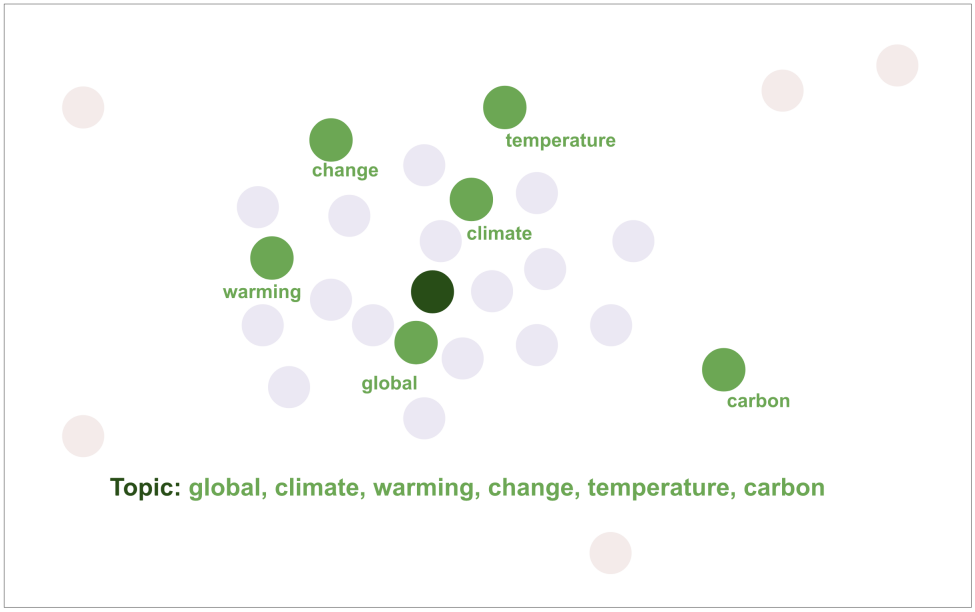


Figure 3.5: The topic words are identified as the word vectors nearest to a topic vector in the semantic space. Since the topic vector represents the centroid of a dense cluster of documents, its closest word vectors reflect the terms most semantically representative of the shared content in that cluster. These words effectively summarize the underlying topic.

Common words appear in most documents and, as such, they are often in a region of the semantic space that is equally distant from all documents. As a result the words closest to a topic vector will rarely be stop-words, which has been confirmed in our experiments. Therefore, there is no need for stop-word removal.

3.4 Discussion

The `Top2Vec` algorithm advances topic modeling by leveraging distributed representations of text and using manifold learning combined with density-based clustering to automatically discover topics in an unsupervised manner. Unlike traditional topic modeling approaches such as LDA [17] or NMF [67], `Top2Vec` does not require a predefined number of topics and eliminates the need for pre-processing, such as stop-word removal, stemming, or lemmatization. The ability to generate topic vectors in a continuous semantic space

enables for a novel way of representing topics which allows for better discovery of topic labels as well as allowing for the vectors to be used for other downstream retrieval tasks.

One of the key advantages of `Top2Vec` is its effectiveness in handling short texts or documents that predominantly discuss a single topic. Traditional topic models often struggle with short texts due to the sparsity of word co-occurrence patterns, which results in incoherent topics. More specifically, LDA and pLSA capture document-level word co-occurrence patterns in order to find topics. When used on short text these models suffer from data sparsity, as each document has few word co-occurrence patterns [25, 51, 58, 102]. Top2Vec operates in a high-dimensional semantic space and clusters documents based on their embeddings rather than word co-occurrences, so it is able to capture topic structure even when individual documents contain limited textual information. Further, since it can use pre-trained embedding models, it allows for transfer learning by leveraging knowledge from embedding models trained on diverse corpora.

Additionally, the joint embedding of words and topics allows `Top2Vec` to generate more informative topic labels. This approach overcomes a common limitation of traditional topic modeling methods, where topics are typically represented by high-probability terms, often including stop words that require manual filtering. By identifying the closest word vectors to the topic vector, `Top2Vec` generates topic labels which are both semantically meaningful and representative of the underlying documents, improving interpretability without requiring human intervention.

Despite its advantages, `Top2Vec` does have limitations, particularly when applied to long documents that cover multiple topics. Since it relies on document-level embeddings, a document that discusses multiple distinct topics will be embedded as a single point in the semantic space, potentially loosing the nuance of its internal topic structure. This can lead to situations where a document is assigned to a single topic even though it should be associated with multiple topics. However, since both documents and topics are vectors, a document can be compared to multiple topics and therefore assigned to multiple topics. Therefore single document vectors do not necessarily prevent documents from being assigned to multiple topics. Its the topic vector creation that limits the distinct document topics from being discovered.

The process of creating the topic vectors relies on the density of single document vectors, and that is what actually limits the learning of multiple topics within documents. Since each document is represented as a single point in the semantic space, any internal topic structure within the document is represented by a single vector. This means that if a document discusses multiple topics, its embedding will be positioned somewhere in between those topics rather than clearly contributing to each one separately. As a result, when

`Top2Vec` clusters document vectors to form topic representations, the granularity of the topics is limited by the fact that multi-topic documents blur the distinctions between different topics.

3.5 Summary

In this chapter, we presented `Top2Vec`, a neural embedding-based approach to topic modeling that leverages distributed document representations to find topic vectors in an unsupervised manner. The key ideas discussed include:

- **Semantic Space and Topic Vectors:** We defined the semantic embedding space as a continuous representation of topics, where dense regions correspond to underlying topics shared by multiple documents. We introduced *topic vectors*, which are computed as the centroids of these dense areas. These topic vectors serve as representative points for each topic, allowing us to characterize and label them effectively.
- **Dimensionality Reduction and Clustering:** High-dimensional embeddings present challenges in identifying dense regions, so we applied **UMAP** for dimensionality reduction, ensuring that local and global structures were preserved. To detect dense clusters, we used **HDBSCAN**, a hierarchical density-based clustering algorithm that automatically determines the number of clusters based on density rather than requiring a predefined number.
- **Informative and Coherent Topic Labels:** Unlike traditional topic models that require stop-word removal, stemming, or manual topic interpretation, `Top2Vec` identifies topic labels directly by leveraging joint embeddings of topic and word vectors. The words closest to a topic vector in the embedding space provide informative and coherent topic labels without additional pre-processing.
- **Advantages over Traditional Topic Models:** `Top2Vec` improves upon traditional topic modeling approaches like **LDA** and **NMF** by eliminating the need for prespecifying the number of topics and improving interpretability through semantic embeddings. It also performs well on short texts, where traditional models struggle due to data sparsity.
- **Limitations:** While `Top2Vec` excels at discovering dominant topics in a corpus, its reliance on single document embeddings can limit its ability to capture *multi-topic* documents.

This chapter introduced the core ideas behind `Top2Vec`, setting the stage for a deeper discussion on clustering techniques in Chapter 4 and multi-document representations and phrase based labeling in Chapter 5.

Algorithm 3.1 `Top2Vec` Algorithm

Input: A set of documents `D`
Output: Topic vectors, topic labels, and document-topic assignments

1. **Embed documents and words**
   Generate vector representations for all documents and vocabulary words.
   - **Model:** `all-mpnet-base-v2` (SBERT)
2. **Dimensionality reduction**
   Reduce document vectors to lower-dimensional space for clustering.
   - **Model:** UMAP
   - **Parameters:** `n_neighbors` = 15, `n_components` = 5
3. **Cluster documents**
   Apply HDBSCAN to find dense clusters in the reduced space.
   - **Algorithm:** HDBSCAN
   - **Parameter:** `min_cluster_size` = 15
4. **Compute topic vectors**
   Compute the centroid of each cluster in the original SBERT space.
   - **Method:** Arithmetic mean
5. **Find topic labels**
   Find nearest word vectors to each topic vector.
   - **Similarity:** Cosine
   - **Labels:** Top-k nearest words
6. **Assign documents to topics**
   Assign each document to the nearest topic vector.
   - **Method:** Cosine similarity

Chapter 4

Clustering Methods for Topic Modeling

4.1 Introduction

One of the main challenges of topic modeling is the identification of semantically similar documents within a large corpus. Finding the similarity between documents is essential for discovering their latent topics.

In traditional topic modeling approaches, LDA, topics are inferred from word co-occurrence patterns. However, these methods have a limited capacity to capture deeper semantic relationships between documents.

As discussed in Chapter 3, neural clustering based topic modeling approaches including `Top2Vec` leverage dense vector representations to embed documents into a continuous semantic space, where proximity in this space reflects semantic similarity. Finding regions of documents that are similar is what allows for segmenting the space into topics. The main idea behind `Top2Vec` is to use density in the embedding space as a signal of an underlying topic shared by the documents. However, there are many other approaches to clustering that can be used to find regions of topically similar documents.

In this chapter, we will explore and evaluate different clustering methods specifically for topic modeling, examining their advantages and limitations. By comparing different clustering approaches, we aim to identify effective strategies for segmenting the document embedding space into coherent topic clusters. Additionally, introduce a hierarchical clustering framework that uses UMAP and HDBSCAN to identify fine-grained, initial topic

clusters and to form corresponding topic vectors. Subsequently, Agglomerative Clustering is applied to the UMAP-reduced topic vectors to construct a full topic hierarchy.

We also This directly addresses the following research question:

- **RQ1**: How to represent topics with neural embeddings? By investigating clustering techniques, we explore how different approaches impact the representation of topics in a continuous semantic space.
- **RQ3**: How to find hierarchies of topics? By combining fine-grained clustering using UMAP and HDBSCAN with Agglomerative Clustering on topic vectors, we create a topic hierarchy, enabling a multi-level semantic organization of the document space.

We aim to develop a deeper understanding of how clustering methods capture topic formation and contribute to the broader goal of improving topic modeling through distributed representations.

4.2 K-Means

K-Means [72] is one of the most popular clustering algorithms and it is usually a strong baseline. It partitions a set of vectors into K clusters. The algorithm seeks to minimize the variance within each cluster by grouping similar data points together. It divides a dataset into K clusters, where each cluster is represented by a centroid. The algorithm iteratively refines the cluster assignments and centroids until a stable configuration is reached. This partitions the vector space into Voronoi cells, where each cell contains points that are closer to a specific cluster centroid than to any other.

The process of K-Means begins with the initialization of K cluster centroids, which can be selected randomly or using more sophisticated techniques. Each data point is then assigned to the nearest centroid based on a distance metric. Once all points are assigned to a cluster, the centroids are recalculated as the mean of all points within their respective clusters. This process of reassigning points and updating centroids continues until the centroids no longer change significantly between iterations, indicating convergence.

The easiest way to use K-Means for topic modeling would be to apply the algorithm directly on the document vectors from a text embedding model. It would require selecting K, which would be the number of desired topics topics. From there we could follow the rest of the `Top2Vec` algorithm to find topic vectors, topic labels, and assign documents to topics.

The advantages of K-Means is that its computationally efficient and can scale to large datasets, additionally it allows for specifying the number of topics which can be desirable when a know number or granularity is known.

However, it also has limitations, such as the need to predefine the number of clusters, K, which can be challenging when the number of topics or the desire granularity is unknown. The algorithm is also sensitive to the initial placement of centroids [8], which can impact the final clustering outcome. Also, K-Means assumes that clusters have a spherical shape [56], which may not always hold true for topic boundaries of document vectors. Further K-Means does not work well with clusters of different size and density [56]. These all make it a suboptimal choice for topic modeling as there can be topics of varying sizes, densities and shapes. An example of a small and dense topic might be something very specific which has been discussed a lot in a corpus. Conversely, a large and sparse topic might cover a broader and more diverse subject but is discussed less.

4.2.1 Agglomerative Clustering

Agglomerative Clustering [109] is a hierarchical clustering algorithm that builds clusters from the bottom up. Each document starts as its own cluster, and pairs of clusters are iteratively merged based on a linkage criterion until a stopping condition is met, usually until a predefined number of clusters is reached or all documents are merged into a single cluster. One of the most commonly used linkage criteria is Ward's method, which merges clusters so as to minimize the increase in total within cluster variance.

A significant advantage of Agglomerative Clustering is that it creates a full clustering hierarchy, which allows exploration of multiple levels of topic granularity. This is particularly useful for topic modeling, where a hierarchy of topics, from fine-grained to coarse-grained allow for exploring the optimal granularity of topics for downstream use-cases. Unlike density-based methods which may label points as noise, agglomerative clustering assign every document to a cluster, ensuring full coverage of the corpus.

However, Agglomerative Clustering also has limitations. It requires a method for selecting the final number of clusters, this would be the most coarse grained topic granularity which may not be known. Additionally, the algorithm has a computational complexity of $O(n^3)$ for naive implementations and $O(n^2 \log n)$ for optimized ones, making it unsuitable for large corpora.

However, its deterministic nature, flexibility in linkage choices, and ability to produce clustering hierarchies make Agglomerative Clustering a valuable method for neural topic modeling, especially for building a hierarchy of topics.

4.3 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [38] is a clustering algorithm that clusters vectors which are in dense areas and marks vectors in low-density regions as outliers. Unlike K-Means, DBSCAN does not require the number of clusters to be predefined, making it particularly useful when the number of topics is unknown. Additionally DBSCAN can handle clusters of arbitrary shape.

One of the main advantages of DBSCAN is that it can handle noise, therefore finding the areas in the semantic space where documents share a coherent topic. Additionally it automatically discovers the number of clusters. This property is particularly useful as it allows to find the natural number of topics in a corpus.

DBSCAN operates using two key parameters: *epsilon*, which defines the radius of a neighborhood around a point, and *minimum samples*, which specifies the minimum number of points required to form a dense region. A point is classified as a core point if it has at least *minimum samples* points within its *epsilon*-neighborhood. Points within the neighborhood of a core point are reachable and become part of the same cluster. Points that are not reachable from any core point are considered outliers.

However, DBSCAN also has some limitations. It may struggle when clusters have varying densities, as a single value of *epsilon* may not be suitable for all regions of the vector space. Additionally, the algorithm's performance can be sensitive to the choice of parameters, requiring careful tuning to achieve optimal clustering results. DBSCAN's inability to handle clustes of varying degrees of density make it suboptimal for topic modeling as there can be topics of varying densities as previously discussed.

4.4 HDBSCAN

HDBSCAN [21, 74] is an improvement of DBSCAN that addresses some of its limitations by introducing a hierarchical clustering approach. Unlike DBSCAN, which requires a fixed distance threshold, *epsilon*, to define dense regions, HDBSCAN works by constructing a hierarchy of clusters and extracting a stable clustering from this hierarchy. This makes it more robust at identifying clusters of varying density, which is a crucial advantage for topic modeling where topics may naturally exhibit different levels of density in the embedding space.

However, HDBSCAN has some challenges. The algorithm requires setting the *minimum_cluster_size* parameter, which controls the smallest allowable cluster size. Higher

values of *minimum_cluster_size* leads to larger clusters by filtering out smaller less stable clusters and a lower value allows more fine grained clusters [74]. Choosing an inappropriate value can lead to either overly granular topics or excessively merged topics.

The selection of the *minimum_cluster_size* parameter plays a major role in controlling topic granularity. A smaller value tends to yield a large number of small clusters, which often capture fine-grained subtopics rather than coherent, high-level themes. While this may enhance sensitivity to local structure, it frequently results in topic fragmentation and can hinder interpretability. Conversely, a larger value can lead to the merging of semantically distinct topics, particularly in the presence of clusters with varying densities, thereby obscuring meaningful distinctions within the embedding space. To ensure a reliable clustering outcome, it is essential to choose a *minimum_cluster_size* that avoids over-fragmentation while maintaining sufficient sensitivity to topic diversity.

We chose HDBSCAN due to the fact that it can automatically discover the number of clusters in the data, it can handle clusters of varying sizes and it can handle clusters of varying densities. This allows the clustering algorithm to match our expectation about topics and prevent us from not being able to discover topics due to the constraints of the other clustering methods discussed.

4.5 Dimensionality Reduction

4.5.1 Curse of Dimensionality

The *curse of dimensionality* refers the the various challenges of working with high dimensional data specifically with regard to distance between vectors, which affects the performance of clustering methods [62]. As dimensionality of a vector space increases, the relative difference between the nearest and farthest points diminishes, making distance metrics less meaningful [2, 12, 48].

In order to overcome the curse of dimensionality, it is often necessary to apply dimensionality reduction before clustering. We will explore several approaches and outline their advantages and disadvantages.

4.5.2 PCA

Principal Component Analysis (PCA) [53, 87] is one of the most popular and used dimensionality reduction techniques. It transforms high-dimensional vectors into a lower-

dimensional representation while preserving as much variance of the data as possible. It computes principal components, a new set of orthogonal basis vectors which are ranked in order of the variance they capture from the original data. By selecting only the top principal components, PCA effectively reduces the dimensionality of the dataset while retaining its most informative features.

PCA is a simpler method that only preserves linear relationships in the data by maximizing the variance captured in the principal components. It assumes that the most important features in the data are those associated with the largest variances, which can loose non-linear relationships often present in high-dimensional embeddings like those from SBERT embeddings. Additionally, PCA does not focus on preserving local neighborhood relationships, which are essential for finding dense areas of the high-dimensional space. This limitation makes it less effective for topic modeling where nuanced representations of dense high-dimensional spaces are necessary.

4.5.3 MDS

Multi-Dimensional Scaling (MDS) [63] is a dimensionality reduction technique that aims to preserve the pairwise distances between vectors when mapping them to a lower-dimensional space. Unlike PCA, which relies on variance maximization, MDS seeks to position vectors in a way that maintains their original distances as accurately as possible. This makes it useful in cases where preserving the global structure of the data is important.

MDS is also not an ideal choice for dimensionality reduction for our use case because it primarily focuses on preserving pairwise distances. While it can handle non-linear relationships, MDS often struggles with scalability as it involves iterative optimization processes that become computationally expensive for large datasets. Additionally, it does not explicitly account for local density variations or neighborhood structures, which are critical for finding dense regions of high-dimensional vectors.

4.5.4 TSNE

One of the most popular and used non-linear dimensionality reduction techniques for high-dimensional data is t-Distributed Stochastic Neighbor Embedding (TSNE) [105]. TSNE works by converting the distances between high-dimensional vectors into conditional probabilities that represent similarities. In the low-dimensional space, it then minimizes the Kullback-Leibler divergence between the original and projected distributions, preserving local relationships between data points.

One of the main strengths of TSNE is its ability to preserve local structure of the high-dimensional data in the low-dimensional embedding, making it effective for identifying clusters present in the high dimensional vectors. However, TSNE comes with several limitations. Its focus on local structure means that global relationships may not be captured in the low-dimensional embeddings. Additionally, TSNE is computationally expensive making it difficult to scale to large volumes of data. It also requires tuning of hyperparameters such as perplexity and learning rate. Lastly, its stochastic nature can also lead to variations in the output between runs, making reproducibility a challenge.

4.5.5 UMAP

UMAP [47,75] is another very popular and widely used non-linear dimensionality reduction technique. UMAP is built upon a theoretical framework based on manifold learning and topological data analysis. It constructs a high-dimensional graph representing the high dimensional vector connectivity and then optimizes a low-dimensional graph to preserve these relationships.

One of UMAP's key advantages is its ability to capture both local and global structures unlike t-SNE. Additionally, UMAP is more computationally efficient, scaling well to larger datasets, outperforming t-SNE in terms of computational speed and memory requirements, which is essential for large volumes of high-dimensional embeddings like those from SBERT models [75, 116].

UMAP has several import hyperparameters that need to be selected. A key hyperparameter, *n_neighbors*, controls the balance between local and global structure. Adjusting this parameters allows for controlling the low-dimensional representation for preserving local versus global structure of the original embeddings. Another important parameter is *n_components*, which determines the dimensionality of the projected embedding space. Lower-dimensional projections may lead to the loss of fine-grained semantic information, particularly when reduced to two dimensions, where structural distortions are more likely. Higher-dimensional projections preserve more information from the original space, but beyond a certain point, they offer diminishing returns. Finally, the choice of distance metric significantly affects how semantic similarity is encoded in the low-dimensional space. For embeddings generated by models such as SBERT, cosine distance tends to outperform Euclidean distance by better capturing angular relationships between vectors, which correspond more closely to semantic proximity. These hyperparameter choices shape the topology of the reduced embedding space, influencing the semantic relationships encoded in their positions.

The configuration of UMAP has a significant impact on the structure and quality of the resulting low dimensional embeddings. Emphasizing local structure leads to the formation of compact, semantically coherent clusters, which are critical for identifying meaningful topics. In contrast, preserving global structure tends to maintain broader topological relationships at the expense of local density, which can obscure fine-grained topic boundaries. The resulting embeddings reflect this trade-off: stronger local preservation enhances the detection of dense topic regions representing granular topics, while global preservation would preserve information about broader themes.

Dimensionality reduction further influences the expressiveness of the embedding space. While extremely low-dimensional projections can compress the data too aggressively, leading to information loss, increasing dimensionality beyond a certain point offers marginal benefit. An appropriate embedding dimension captures the essential semantic structure while avoiding the curse of dimensionality.

We select UMAP as our dimensionality reduction technique due to its ability to preserve local and global structure in the data as well as its ability to scale to large volumes of data.

4.6 Deep Clustering

Data representation is essential for clustering algorithms because it directly influences how well clustring algorithms can identify meaningful patterns and group similar data points. Deep Clustering is an unsupervised learning method that combines representation learning and clustering, leveraging neural networks to learn latent feature spaces that improve clustering [94, 112]. Deep clustering is particularly useful for handling complex and high-dimensional data such as images and text where traditional clustering methods struggle due to the curse of dimensionality. By learning representations specifically for clustering, deep clustering improves performance of clustering without requiring explicit dimensionality reduction. This makes deep clustering attractive as it can remove the reliance on dimensionality reduction techniques like PCA, t-SNE and UMAP.

We will explore several deep clustering algorithms relevant to our research and discuss their strengths and weaknesses.

4.6.1 Autoencoders

An autoencoder (AE) [61, 97] is a neural network-based unsupervised learning model which learns compressed latent representations of data by encoding the input into a lower-

dimensional space and then reconstructing it with minimal loss of information. The bottleneck in the latent layer forces the model to generalize and learn the most essential features of the data.

An AE has two main parts: an encoder, which transforms the input into a latent representation, and a decoder, which reconstructs the original input from this representation. The model is trained to minimize the difference between the input and its reconstruction, typically with mean squared error loss, ensuring that the learned representation captures the most important features of the data.

Variants of AEs such as denoising autoencoders [107], variational autoencoders [60], and sparse autoencoders introduce additional methods to improve robustness, disentanglement, or generative capabilities. Autoencoders are widely used for dimensionality reduction and feature learning.

4.6.2 Deep Embedded Clustering

Deep Embedded Clustering (DEC) [112] is an unsupervised clustering algorithm that simultaneously learns a latent feature representations and cluster assignments. It uses a deep autoencoder to learn a mapping into a lower-dimensional latent space by iteratively optimizing a KL divergence based clustering objective with a self-training target distribution. Its main objective is to learn a representation specialized for clustering without ground-truth cluster labels.

DEC first pre-trains a deep autoencoder using reconstruction loss to learn meaningful latent representations. After pretraining the decoder is discarded, and the encoder is fine-tuned using a clustering objective. After initial pre-training reconstruction loss is no longer used, and the model focuses solely on improving cluster assignments.

The approach clusters data points into k clusters by first transforming the data into a lower-dimensional latent space, Z, using deep autoencoder. The dimension of Z is usually lower than the original dimensionality of the data in order to avoid the curse of dimensionality. The DEC algorithm clusters data by jointly learning k cluster centers in the latent feature space and optimizing the parameters of a deep autoencoder that maps data points into the latent space.

One of the main advantages of DEC in clustering-based topic modeling is its ability to learn a latent representation that is optimized for clustering, which can enhance topic separability and coherence.

Despite its advantages, a major drawback is that it requires the number of clusters k apriori, which is often unknown in topic modeling. Additionally, since DEC discards the decoder after pretraining, it loses the ability to reconstruct text data, which can limit the semantic coherence of the latent space.

4.6.3 Deep Clustering Network

Deep Clustering Network (DCN) [114] is an unsupervised clustering method that jointly optimizes a learned latent space and K-means clustering. DCN proposes an optimization criterion which consists of three components: dimensionality reduction, data reconstruction, and cluster structure-promoting regularization. A decoding network is used for reconstruction to prevent trivial solutions, like forming arbitrarily tight clusters, thereby achieving a low clustering loss without capturing meaningful structure in the data. DCN uses a deep autoencoder to learn the lower dimensional latent space and also clusters data points into k clusters.

Unlike DEC, which refines clusters using KL divergence without explicitly optimizing for K-means, DCN integrates the K-means objective directly into its learning process, so that the learned feature space is cluster-friendly for K-means.

DCN has several advantages for clustering-based topic modeling. First, its direct integration of K-means clustering into the learning process, which ensures that the latent space is optimized for clustering. This can lead to improved topic coherence and more distinct topic representations compared to traditional autoencoder-based methods. Additionally, by incorporating a reconstruction objective, DCN helps preserve essential semantic information in the learned latent representations, allowing it to preserve the semantic relationships in the latent space. However, DCN has a key limitation, its requirement to specify the number of clusters k beforehand, as the optimal number of topics is often unknown in topic modeling.

4.6.4 Dip-based Deep Embedded Clustering with k-Estimation

One of the major limitations of both DEC and DCN is that they require the number of clusters, k, to be known apriori. Dip-based Deep Embedded Clustering with k-Estimation (DipDECK) [68], is an unsupervised clustering algorithm which estimates the number of clusters while learning the clustering-friendly representation and number of clusters simultaneously. Additionally, it can cluster complex data without assuming only spherically shaped clusters.

DipDECK learns three key components simultaneously: estimating the number of clusters (k), determining cluster assignments, and enhancing latent vector representation. It initially overestimates clusters using a deep autoencoder and then applies Hartigan's Dip-test [44], a statistical test that produces a Dip-value to detect structurally similar sub-clusters.

By incorporating a clustering loss that encourages merging of clusters with high Dip-p-values, the method compacts these groups into common clusters. This eliminates the need for a fixed k value and can identify non-convex cluster shapes. Despite this DipDECK still requires an initial number of clusters to be selected, which acts as an upper bound. It also requires a threshold to be selected for the Dip-value which determines whether two clusters should be merged.

One of the main advantages of DipDECK is its ability to estimate the number of clusters dynamically, removing the need for a predefined k. This makes it particularly useful for topic modeling, where the number of underlying topics is typically unknown. Additionally, its ability to handle non-spherical clusters allows for more flexible topic representations. However, DipDECK has a few limitations. The requirement for an initial overestimation of k, can be difficult to estimate. Further, selecting a threshold for the Dip-value means that hyperparameter tuning is still required. In topic modeling, where interpretability is crucial, the merging process of clusters may lead to difficulties in understanding the final topic structure, particularly if the merging criteria do not align with meaningful semantic groupings.

4.6.5 Sparse Autoencoders

The Sparse Autoencoder (SAE) [91,103] modifies the standard autoencoder by introducing a constraint that limits the number of active neurons in the compressed latent layer. In addition to the objective of minimizing the difference between the input and its reconstruction, SAEs incorporate an extra penalty, usually L1 loss, designed to encourage most hidden units to remain inactive. The constraint of only a few active latent neurons forces the network to represent the most important features of the input. The resulting representation tends to be more interpretable, as each active neuron is more likely to correspond to a distinct, meaningful attribute of the data.

A major challenges in training SAEs is tuning the L1 regularization parameter, which controls the degree of sparsity in the latent representation. An overly strong regularization can lead to excessive sparsity, reducing the model's capacity to capture meaningful patterns, while too weak a regularization may result in dense activations, undermining

the benefits of sparsity. Top-k SAE [73] introduce an alternative approach by enforcing sparsity through an activation thresholding mechanism, where only the top-k most activated neurons are retained for each input sample. This method provides more controlled and interpretable sparsity, reducing the need for extensive hyperparameter tuning while maintaining robust feature learning.

In distributed representations generated by embedding models or in intermediate layers of LLMs, concepts are represented by a pattern of activations in multiple neurons. Therefore, a single neuron can contribute to representations of many different concepts, including semantically unrelated concepts. This is because the meaning is encoded by the combination of active neurons. Polysemanticity is a term used to describe the behaviour where individual neurons can activate in multiple semantically different contexts.

The challenge of polysemanticity has motivated research to explore methods for disentangling activations and uncovering interpretable representations. The authors of [29] use a top-k SAE on internal activations of a language model to find monosemantic sparse activations in the SAE. This method allows for discovering interpretable features in language model activations. The authors of [85] use a top-k SAE to extract interpretable features from dense representations generated by an embedding model. These approaches demonstrate that using a top-k sparse SAEs can be used to disentangle polysemantic activations. This allows for revealing interpretable features within language model and embedding model representations. By sparsifying activations, these methods enable a clearer understanding of how different concepts are encoded, potentially improving model interpretability. This shows that SAEs could potentially be used for topic modeling.

SAEs can be useful in clustering-based topic modeling by providing dimensionality reduction and enhancing the interpretability of learned features. By enforcing sparsity in the latent space, SAEs can learn interpretable features, allowing for discovering distinct topic structures in document embeddings. Additionally, SAEs can serve as a an alternative to dimensionality reduction techniques like PCA or UMAP by preserving the most meaningful features of high-dimensional text embeddings.

However, SAEs also have some disadvantage. One of the largest issues is dead latents, where some neurons remain inactive across all inputs, reducing the model's capacity to learn a diverse set of features. There are many strategies for dealing with this issue but none completely solve it. Another challenge is balancing the sparsity constraint, excessive sparsity can lead to an overly constrained latent space that fails to capture meaningful representations of the data, while insufficient sparsity results in less interpretable and overly dense representations. Further, training SAEs requires careful tuning of regularization parameters, such as the L1 penalty or top-k activation threshold. Despite these challenges,

SAEs can be used to improve the interpretability and effectiveness of distributed representations.

4.7 Clustering for Hierarchical Topic Modeling

One of the main challenges in topic modeling is choosing the right resolution of topics. Most topic modeling methods require the number of topics to be selected. We want to automatically find the natural number of topics present in a dataset at multiple resolutions, therefore we want to leverage unsupervised methods or build a hierarchy that captures many granularities. For hierarchical topic modeling, it is important for the higher-level clusters to *cover* the semantic space and be *representative* of the fine-grained clusters they summarize. An effective hierarchy ensures that coarse-grained topics encapsulate the diversity of their subtopics. This allows for effective exploration of the hierarchy to find relevant semantic information at the right granularity.

The assumption of `Top2Vec` is that the dense areas in the embedding space are indicative of underlying topics. UMAP can be configured to capture more local or global structure through tuning the `n_neighbors` parameter and HDSBCAN can be configured to find more or less granular structures within that density by tuning the `min_cluster_size` parameter. In order to find the natural underlying topic structure in a dataset, we propose using multiple configurations of UMAP followed by HDBSCAN in order to find densities in both local and global structures. When tuning UMAP for global structure, the HDBSCAN clustering will capture high-level topics. When tuned for local structure, it can capture more fine-grained subtopics. In Chapter 7, we evaluate this idea empirically using labeled datasets to show how UMAP's structure preservation and HDBSCAN's density sensitivity enable discovery of topic hierarchies.

The combination of UMAP tuned for local structure and HDBSCAN for granular clusters is very good at finding natural and diverse clusters of semantic regions which identify coherent topics while also identifying sparse areas that are not topically coherent. We show in Chapter 7 that UMAP can be tuned to preserve more global structure and HDBSCAN can be tuned for finding less granular clusters which will result in finding high-level topics. Although that is a very effective technique it is important to clarify what global structure this combination actually captures and what the underlying hierarchy represents. HDBSCAN with a higher `min_cluster_size` will not aggregate coarse-grained clusters found by smaller `min_cluster_size` it instead finds the most persistent and dense regions; therefore it will not find a coarse to fine-grain topic hierarchy.

HDBSCAN builds a single-linkage dendrogram from mutual-reachability distances, then condenses it by pruning any cluster that ever falls below the specified `min_cluster_size`. Raising this threshold does not merge finer clusters but instead reconstructs the hierarchy to retain only the most persistent and densely populated regions. Thus, higher values of `min_cluster_size` uncover stable dense regions rather than aggregates of previously discovered granular clusters, capturing persistence rather than semantic coverage.

Therefore, the resulting clusters should be interpreted as the most salient regions of the semantic space, but not necessarily as representative summaries of the full diversity of granular topics. Smaller yet semantically coherent clusters may be excluded. Thus, while this approach offers a robust high-level view, it is not optimal if the goal is to construct comprehensive and representative abstractions over all fine-grained topics.

In order to overcome this limitation and create a true topic hierarchy that represents all granular clusters correctly propose a two-stage approach. First, using UMAP and HDBSCAN we find the fine-grained topics that naturally exist in the corpus and simultaneously filter the noise documents that are not part of salient topics. We then calculate topic vectors from each dense area. The assumption here is we have effectively captured the granular topic distribution. To build a coherent hierarchy over these topics, we then apply agglomerative clustering [109] to the topic vectors. We also use UMAP on the topic vectors in order to deal with the curse of dimensionality and capture their topology. Unlike HDBSCAN, this method produces a complete hierarchical tree without pruning, ensuring that all fine-grained topics are retained in coarser representations, therefore allowing for multi-resolution analysis of the topic space.

4.8 Summary

In this chapter, we explore various clustering methods for topic modeling, including their strengths, limitations, and suitability for clustering document embeddings. We compare traditional clustering methods, density-based approaches, deep clustering techniques, and evaluated the role of dimensionality reduction in improving clustering performance. Our results in 7.3 demonstrate that UMAP followed by HDBSCAN outperforms other clustering techniques for finding semantically coherent clusters of text.

Key ideas discussed in this chapter include:

- **Clustering in High-Dimensional Space is Challenging:** Traditional clustering methods like K-Means and DBSCAN struggle with high-dimensional text embed-

dings due to the curse of dimensionality. Applying clustering directly to raw document embeddings leads to suboptimal results, confirming the need for dimensionality reduction.

- **UMAP is the Best Dimensionality Reduction Technique for Clustering:** We compared dimensionality reduction techniques and conclude that UMAP is the best option for topic modeling due to its ability to capture local and global structure as well as scale to large datasets.
- **HDBSCAN is a better choice than K-Means and DBSCAN for Topic Discovery:** HDBSCAN does not require specifying the number of clusters in advance and can discover topics of varying densities, shapes, and sizes unlike K-Means and DBSCAN.
- **Deep Clustering Learns Representations Specialized for Topic Modeling:** Deep clustering methods like DEC, DCN, and DipDECK simultaneously learn latent feature spaces and cluster structures.
- **Sparse Autoencoders Enhance Topic Interpretability by Disentangling Features:** By enforcing sparsity in the latent space, Sparse Autoencoders reveal interpretable features in document embeddings, which can uncover distinct and semantically meaningful topics.
- **Multi-resolution Clustering Enables Hierarchical Topic Discovery:** By adjusting UMAP's `n_neighbors`, we can control the granularity of the topic space, from fine-grained subtopics (local structure) to broad thematic categories (global structure). This enables hierarchical topic modeling using `Top2Vec` by running UMAP and HDBSCAN at different resolutions to uncover both coarse and detailed topic levels.
- **Two-Stage Clustering Builds Complete and Coherent Topic Hierarchies:** First clustering documents into fine-grained topics with UMAP and HDBSCAN, then applying agglomerative clustering on the resulting topic vectors, produces a full hierarchical tree that retains all topics, enabling multi-resolution exploration of the topic space without losing important granular clusters.

In the next chapter, we introduce `C-Top2Vec`, a novel topic modeling approach that leverages multi-vector document representations, contextualized token embeddings, and phrase-based labeling to improve topic discovery, segmentation, and labeling.

Chapter 5

Contextual Top2Vec

5.1 Introduction

In this chapter, we propose `C-Top2Vec` and show how we create multi-document vector representations, automatically discover topics, and label topic segments within documents.

`C-Top2Vec` is a novel topic modeling approach which builds upon `Top2Vec`. This approach leverages contextualized token embeddings and creates multi-vector document representations to improve topic discovery and representation. It also uses phrases to label topics rather than just words for improved interpretability. This chapter addresses several key research objectives and research questions outlined in this thesis:

- **Distributed topic representations:** We develop another method that represents topics using neural embeddings, ensuring that topics are situated in a meaningful semantic space. This approach addresses RQ1 by finding topic vectors as centroids of dense regions of sub-document vectors in the embedding space, enabling a distributed representation of topics.
- **Informative and coherent topic labels:** We propose a phrase-based topic labeling approach that uses phrase embeddings derived from the corpus. This addresses RQ2 by ensuring that topic labels are meaningful, coherent, and representative of their associated documents. We also explore the use of LLMs to generate topic labels.
- **Automatic discovery of topic hierarchy:** Our method applies hierarchical topic reduction, which merges semantically similar topics in a data-driven manner. This

addresses RQ3 by allowing for the discovery of a hierarchical topic structure, enabling exploration of topics at different levels of granularity. We also describe several methods for finding hierarchical topics.

- **Document-level topic segmentation:** We develop a method for assigning topics at the token level, addressing RQ4, by enabling fine-grained segmentation of documents into topic spans. This provides a detailed view of how topics are distributed within documents.

This chapter begins by discussing the foundation of contextualized token embeddings and how they enable improved document representations. We then introduce our multi-vector document representation technique, which enhances clustering-based neural topic modeling by capturing diverse topical information within a single document overcoming the limitations of single vector document representation. Next, we describe how we discover topics using density-based clustering of sub-document vectors and construct topic hierarchies through hierarchical reduction. We further present our method for topic labeling using corpus-extracted phrases and illustrate how we segment documents into topic spans. Lastly, we explore the use of LLMs for generating topic labels. A summary of the full `C-Top2Vec` algorithm is presented in Algorithm 5.1.

5.2 Contextual Token Embeddings

The self-attention mechanism introduced in the Transformer [106] and later used by BERT [32] led to contextualized token embeddings. Unlike traditional word embeddings that have a single vector representation, contextualized embeddings have different vectors depending on their context. SBERT [93] uses the BERT architecture to train embedding models that ensure that semantically similar documents are close in the embedding space. The sentence embeddings are created by pooling the contextualized token embeddings of the document.

The self-attention mechanism introduced in the Transformer [106] revolutionized neural language models by allowing models to focus on different parts of a sequence depending on their relevance to the task being learned. Unlike recurrent neural networks (RNNs), which process sequences linearly, self-attention enables the model to capture long-range dependencies and relationships between tokens of a sequence regardless of their distance from each other within the limits of a fixed context window. The self-attention mechanism calculates the relationships between all pairs of tokens in the sequence simultaneously, assigning attention scores that dictate how much importance each token should have relative to others.

One of the major benefits of the self-attention mechanism is its ability to create contextualized token embeddings, which create different embeddings for words depending on their context. In traditional embedding models like word2vec, each word is assigned a single static vector that does not change regardless of its usage. For example, the word "apple" would always have the same vector representation, whether it refers to the fruit or the company. In contrast, the attention mechanism generates token embeddings that depend on the context in which a word appears.

The motivation behind BERT [32] was the success of the Transformer architecture [106] in capturing long-range dependencies and generating contextualized token embeddings, combined with the limitation that pre-Transformer models processed text in a unidirectional or shallow bidirectional manner. BERT addressed this limitation by introducing a masked language model objective, enabling bidirectional training of the Transformer to produce context-rich embeddings by considering the tokens preceding and following a given word. Further, BERT included a next-sentence prediction task to enhance its understanding of sentence-level relationships. Although BERT performs very well in token-level and sentence-level tasks, its architecture was not optimized for producing meaningful document embeddings. The approach of averaging or pooling token embeddings from BERT often results in suboptimal performance for tasks such as semantic similarity, clustering, and information retrieval [93].

SBERT [93] extended BERT's architecture to create document embeddings, enabling the representation of entire sentences or documents as fixed-size vectors. SBERT achieves this by fine-tuning BERT using a siamese network structure with contrastive loss, ensuring that semantically similar documents are close in the embedding space and semantically dissimilar documents are far apart. To create these embeddings, SBERT uses a pooling strategy over the contextualized token embeddings of a text sequence using mean pooling, max pooling, or by just taking the '[CLS]' token representation produced by the model for the entire sequence. This pooling aggregates the information across the tokens to produce a single vector representation for the sentence. SBERT is trained to be highly effective for tasks such as semantic similarity, clustering, and information retrieval.

The first step in our algorithm is to create contextualized token embeddings for each document. This is done by embedding the documents with an SBERT model and taking the contextual token embeddings before the pooling layer, as shown in 5.1. Any embedding model can be used in this step as long as it has contextual tokens, it uses average pooling, and it was trained for semantic similarity. We justify these requirements in the following section.

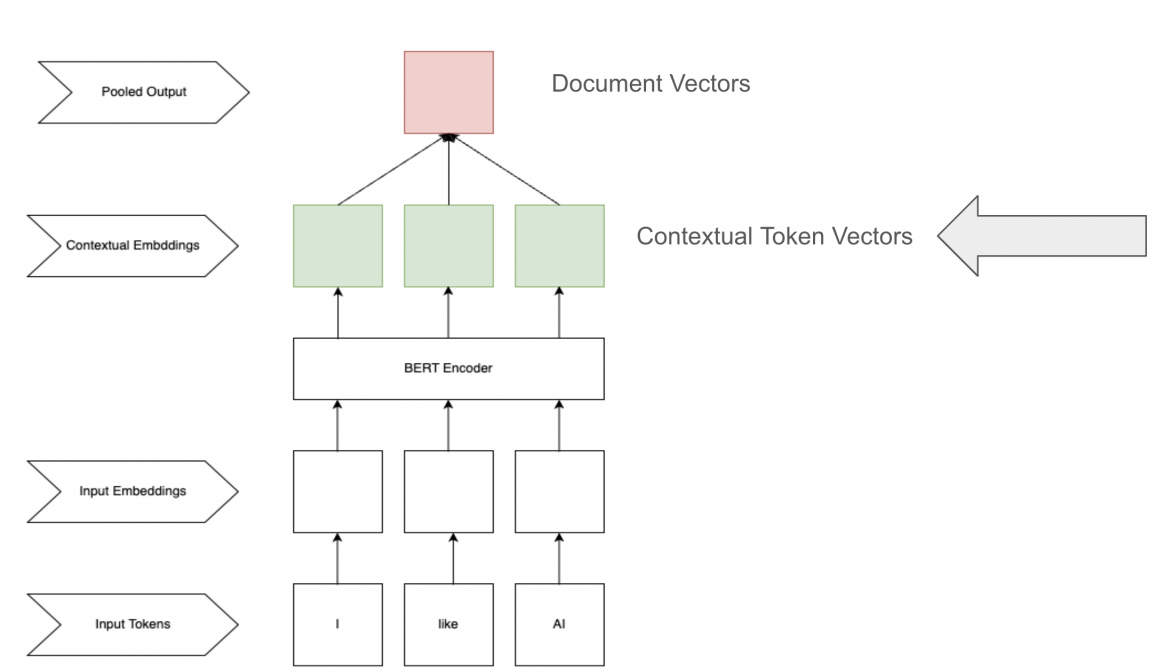


Figure 5.1: This figure shows the contextual token layer of a transformer-based embedding model like SBERT. Before the pooling layer, each token is represented by a contextualized vector that reflects its meaning within the sentence. These token-level embeddings form the basis for downstream semantic representations in our method.

5.3 Multi-vector Document Representation

Most neural models represent documents as single vectors. However, single-vector document representations cannot capture all contextual information, especially in long documents or ones with diverse topics [71], [117].

The authors of [71] study dense retrieval by looking at single-vector representation of documents and they conclude that single-vector representations of documents cannot capture the multiple semantic views of a document because a single vector aggregates the overall meaning of a document. From the retrieval perspective, documents may have different sections relevant to different queries which cannot be distinguished with a single vector. Single-vector representations collapse various semantic aspects into one unified representation, limiting the model's ability to align with queries for specific or diverse

sections of the document. They perform experiments with the SQuAD dataset [90] to demonstrate this. The authors of [71] also find that there is a limit to single vector representation performance in retrieval as document size increases.

In the work of [117], they propose the use of a multi-view representation of a document that gives it multiple vectors for the use in retrieval. In BERT models the '[CLS]' token is used for the next sentence prediction task and it has been shown that it encodes the overall meaning of a sentence [27]. The authors propose adding several '[VIE]' tokens to the beginning of a sequence as randomly initialized trainable parameters, they propose a special loss to ensure the '[VIE]' tokens do not collapse into the same representation and to encourage alignment with varying queries. This method allows the modified BERT model to generate multiple embeddings that better reflect the multi-faceted nature of documents, making them more adaptable to diverse queries. They demonstrate superior performance on the SQuAD dataset using this approach compared to single document vector representation for retrieval.

We conclude from these findings that a document with multiple topics cannot be well represented by a single vector as the different topics within the documents must all be captured in the single vector. Aggregating all the information from a document into a single vector can obscure the nuanced relationships and distinctions between different topics of the document. This limits the power of topic modeling using single document vectors, as the diversity of topics within the document is reduced to a single representation which makes it difficult to extract by topic models.

We therefore propose the use of multi-vector document representation for topic modeling. A naive approach to creating a document with multiple vectors is to chunk the document into fixed-length chunks and then embed those to create multiple vectors. The limitation of this approach is that the different chunk vectors will be missing context from preceding and following chunks. Another approach is using segmentation techniques to find natural topic boundaries for chunking; however, this approach also has limitations, as it relies heavily on the accuracy of the segmentation technique. Poor segmentation can lead to unnatural breaks, causing loss of coherence and semantic information. Further, segmentation techniques often require task-specific fine-tuning and may not generalize well across diverse datasets or document type.

In order to create multi-vector document representations, we propose a simple approach that leverages contextual token embeddings. In order to generate contextual token embeddings we use SBERT models since they are specifically trained for semantic similarity, and ensure semantically similar documents get embeddings which are close together and those that are not have embeddings that are far apart.

There are three predominant types of pooling used in SBERT models: mean pooling, max pooling and CLS pooling. Mean pooling takes the average of all token embeddings to create a fixed-size sentence embedding, max pooling selects the maximum value for each dimension across all token embeddings. CLS pooling uses the embedding of the special '[CLS]' token as the sentence representation. Since we want token embeddings that represent each token's semantic representation we use SBERT models with mean pooling. Mean pooling is most desirable as it ensures that the learned token embeddings are less influenced by outliers, providing a balanced representation of the document's semantics. CLS pooling on the '[CLS]' token, might not fully capture the nuances of all tokens in the context, as it is optimized for next-sentence prediction. Max pooling will capture the most prominent semantic feature, ignoring semantic information from less dominant tokens. By using mean pooling, we ensure that every learned token contributes equally to the document's representation, preserving the contextual relationships among tokens.

In the original SBERT paper, the experiments show that mean pooling performs better than max and CLS pooling, when evaluating the resulting sentence vectors [93]. The authors of [113] perform a comparative analysis of pooling mechanism. The study demonstrates that mean pooling generates stable and balanced representations by aggregating token embeddings without overemphasizing individual tokens. This is because mean pooling reduces the influence of outliers in the pooled representation. Their findings also show that max pooling, while effective at capturing the most salient feature, can lead to biased embeddings that neglect less dominant contextual information. By leveraging mean pooling, SBERT ensures that all tokens contribute equally, preserving contextual nuances and leading to superior performance in tasks requiring comprehensive semantic understanding.

Additionally, during training of an SBERT model, mean pooling allows for information to backpropagate to each token embedding equally. Since the sentence embedding is the average of all token embeddings, the gradient signal from downstream tasks is distributed across every token. This ensures that each token can adjust its parameters in response to the overall sentence-level objective. In contrast, max pooling directs the gradient updates primarily to tokens with the maximum activation in any dimension, limiting the backpropagation signal to a subset of tokens. Similarly, CLS pooling focuses the gradient on the [CLS] token, reducing the direct updates that other tokens receive. By evenly disseminating the training signal, mean pooling not only captures a more nuanced representation of the entire sentence but also fosters richer, more balanced token-level learning.

In order to create multiple vectors, we want to ensure topical information is captured from throughout the document. Our proposed approach uses a sliding window across the contextualized tokens of a document and applies mean pooling to the tokens in each window. The benefit of this approach is that the contextualized tokens already capture the

context of surrounding tokens, so by arbitrarily taking chunks of them surrounding context is not lost. The larger the sliding window, the less granular the topical representation is and the smaller the context window the more granular the topical representation. The resulting pooled windows become sub-document vectors, or contextual chunks, which capture the topical information from different parts of the document.

The second step of our algorithm is to create multiple vectors for each document in order to capture topical information from each part of the document. This is done by using a sliding fixed-sized window with mean pooling over the contextualized token embeddings of each document. Figure 5.2 shows this process. This operation aggregates information from multiple contextualized token embeddings and creates a single vector for each position. The pooled vectors from each position represent the topical information from that segment of the document. The size of the window determines the granularity of the representation of the document topic. We use a window size of 50 and stride of 40. With contextual tokens overlap is not necessary since the information of surrounding tokens is already captured which is why we chose a very small overlap, we chose the window size based on experimental results, see section 7.5.11 for analysis of window size and table 7.23 for window size comparison.

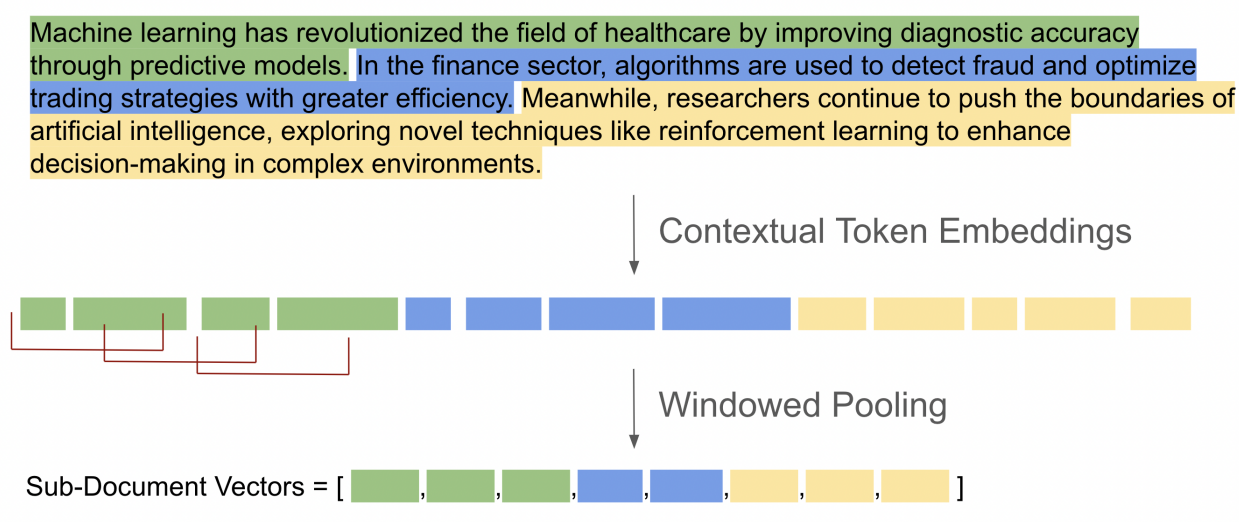


Figure 5.2: This figure illustrates the generation of sub-document vectors using a sliding window over contextualized token embeddings. Each vector is created by applying mean pooling within a fixed-size window, capturing topical information from local segments of the document. These sub-document vectors allow for finer-grained topic representation, with window size and stride controlling the granularity.

5.4 Topic Vectors

The main premise of our topic modeling approach is that the embedding space represents semantic similarity and density of document vectors in that space represents a common underlying topic to those documents [6]. Using our multi-vector representation of documents, we find dense areas of those sub-document vectors. The dense areas are representative of an underlying topic that is common to them. The centroid of each dense area, being the central point, is the most representative of the common topic of the sub-document vectors in that region. Thus, to create topic vectors, we find the centroids of each dense area.

In the case of `C-Top2Vec` [7], we use multi-vector representations of documents in order to capture multiple topical units from each document. So instead of finding dense areas of documents like in `Top2Vec`, we find dense areas of sub-document vectors. This allows us to find much more granular topics.

We first use UMAP to reduce the dimensionality of the vectors so we can apply density-based clustering to identify dense areas of vectors. We use HDBSCAN [21] on the low-

dimensional vectors to find dense areas. We use HDBSCAN since it can find clusters of varying densities, unlike Density-Based Spatial Clustering Of Applications With Noise (DBSCAN). Additionally, it does not require the number of clusters to be specified. This allows us to find the natural number of clusters present in the data which allows us to automatically find the upper bound of the number of topics present in the dataset.

For each dense area identified by HDBSCAN we calculate the centroid of the vectors in their original embedding dimension. This has the benefit of correcting for distortion introduced by dimensionality reduction. The resulting centroids become the topic vectors. Figure 5.3 shows this process.

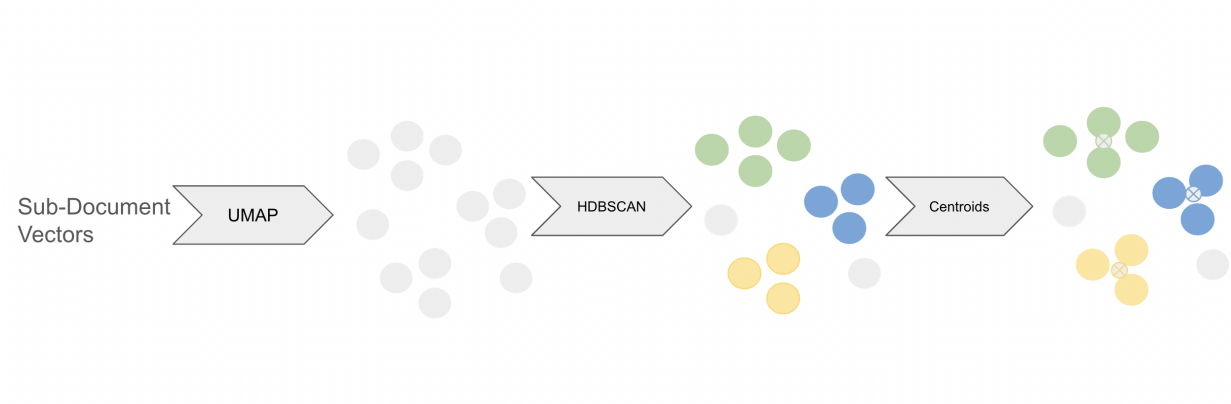


Figure 5.3: This figure illustrates how topic vectors are computed by taking the centroid in the original high-dimensional embedding space of each dense area identified by HDBSCAN in the UMAP-reduced space. Calculating centroids in the original space corrects for distortions introduced by dimensionality reduction, ensuring the resulting topic vectors faithfully represent the underlying document clusters.

5.5 Hierarchical Topics

The number of initial topics is determined by the number of dense areas found by HDBSCAN. An advantage of the topic vectors and the continuous representation of topics in the semantic space is that the number of topics found can be hierarchically reduced to any number of topics less than the number initially found. This can done by iteratively merging the smallest topic into its most semantically similar topic until the desired number of topics are reached. This is done by taking a weighted arithmetic mean of the topic vector of the

smallest topic and its nearest topic vector, each weighted by their topic size. After each merge, the topic sizes are recalculated for each topic. This hierarchical topic reduction has the advantage of finding the topics that are most representative of the corpus, as it biases towards topics with greater size.

Hierarchical topic reduction is performed with the following algorithm until the selected number of topics is reached:

1. For each sub-document vector d_i in the set of sub-document vectors $D = \{d_1, d_2, ..., d_n\}$:
   - Assign d_i to the nearest topic vector in the set of topic vectors $T = \{t_1, t_2, ..., t_m\}$.
   - The size of each topic vector in T is determined by the number of sub-document vectors assigned to it.
2. Find the topic vector t_{min} in T with the smallest size.
3. Find the nearest topic vector t_{nearest} to t_{min}, based on cosine similarity.
4. Merge t_{min} and t_{nearest} by taking their size-weighted mean.

There are several other approaches that can be used for finding hierarchical topics. One other method is to apply K-Means clustering at multiple resolutions to the UMAP reduced embeddings. This allows for precise control over the number of topics at each level of granularity, which can be advantageous when the desired number of topics is known or needs to be fixed for some downstream task. However, K-Means assumes that clusters are spherical and equally sized, which makes it less effective when the true topic distribution exhibits varied densities, irregular shapes, or overlaps, which are common characteristics of textual data, as discussed in Chapter 4. As a result, while K-Means provides precise control over the number of topics, it may fail to accurately capture the true topic distributions, particularly at coarser or more nuanced levels of hierarchy.

Although our hierarchical topic reduction involves iterative assignment and centroid updates similar to K-Means, it differs in its objective and behavior. Rather than minimizing within-cluster variance, it performs greedy, size-aware merges of semantically similar topics, resulting in a bottom-up compression of the topic space. This approach makes no assumptions about cluster shape or size, allows dynamic control over topic granularity, and better preserves the structure of dense, semantically coherent regions identified in the initial clustering. As such, it is more comparable to agglomerative merging in embedding space than to partition-based methods like K-Means.

While the hierarchical reduction method described above is the one implemented in `C-Top2Vec`, the hierchical clustering methods introduced in Chapter 4.7 are reasonable alternatives. The use of UMAP and HDBSCAN with varying configurations to vary the global versus local structure offers an unsupervised approach to discovering topic hierarchies at different levels of granularity. Further, the agglomerative clustering of topic vectors enables the construction of a full hierarchical tree without discarding any fine-grained topics, thereby ensuring complete semantic coverage. Although these methods are not used in the core implementation of `C-Top2Vec`, they serve as meaningful baselines and will be explored in Chapter 7.4, where we evaluate their effectiveness in preserving semantic structure and constructing topic hierarchies across multiple datasets.

5.6 Topic Labeling

To create labels for each topic, we extract phrases from the corpus of documents. We use NPMI to identify n-gram collocations [18] [92], which discover word combinations that co-occur more frequently than would be expected by chance. The use of NPMI reduces bias toward low-frequency collocations and ensures that the phrases are both statistically significant and interpretable. These collocations are extracted directly from the corpus so that they are representative of the documents. The phrases are meaningful labels that can be used to label the topics present in the corpus. Alternatively, the `Top2Vec` method can be utilized, which generates topic labels by leveraging the dataset vocabulary, rather than relying on extracted phrases. This flexibility ensures that the labeling approach can be adapted based on the requirements and nature of the dataset.

Once we select the labels, either phrases or corpus vocabulary, we jointly embed them with the document or sub-document vectors. The main assumption is that a topic vector is the most central point of a topic area and therefore near phrases are most representative of the topic. To create labels for each topic, we select the most similar phrases based on cosine similarity to the topic vector as shown in Figure 5.4.

Using the phrases, we create topic labels for each topic with the following algorithm:

1. Embed all phrases to create a set of phrase vectors $P = \{p_1, p_2, ..., p_n\}$.
2. For each topic vector t_i in the set of topic vectors $T = \{t_1, t_2, ..., t_m\}$:
    - Find the N nearest phrase vectors in P to the topic vector t_i based on cosine similarity.

3. Assign the N nearest phrases found for each topic vector t_i as the labels for that topic.

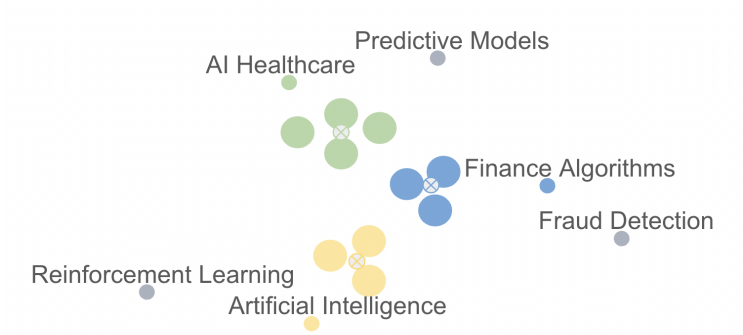


Figure 5.4: This figure illustrates the topic labeling process, where candidate phrases are jointly embedded with topic vectors. Labels are assigned by selecting the phrases with highest cosine similarity to each topic vector, based on the assumption that the topic vector lies at the semantic center of its associated topic area. These nearest phrases serve as representative labels for each topic.

We also propose an alternative topic labeling approach using large language models (LLMs), by leveraging our topic vectors, document topic relevance and phrase-based labels. This method leverages the semantic richness of LLMs to generate more coherent and context aware topic labels. The core idea is that each topic vector represents the semantic center of a topic cluster; thus, documents and phrases located near the topic vector in the embedding space are the most representative examples of that topic.

To generate a label for each topic, we select the top-k most relevant documents and the top-k most similar phrases to a given topic vector, using cosine similarity as the distance metric. These elements are then used as input to an LLM, specifically a Mixtral [57] model, prompted to generate a concise, informative topic label that reflects the shared theme among the selected documents and phrases. This approach is particularly valuable for labeling large or complex topic clusters, where nearest neighbout phrase-based labels may produce overly generic or specific labels. By summarizing multiple documents and common phrases representing the topic, the LLM can find a balance between specificity and generality, potentially improving topic interpretability and coherence.

Furthermore, techniques such as Maximal Marginal Relevance (MMR) [22] can be used to select a diverse and relevant subset of documents, reducing redundancy and enhanc-

ing the representativeness of the input to the LLM. This increases the chances that the resulting label captures the full breadth of the topic. While this method is more computationally intensive and introduces additional considerations such as prompt design and model selection, it provides a powerful framework for generating coherent and interpretable topic labels. In this work, we offer a proof-of-concept implementation to demonstrate its potential, and we leave further optimization and large scale evaluation for future research.

5.7 Topic Assignment and Segmentation

We assign topics to document segments at the token level as shown in 5.5. First, token-level embeddings are smoothed by averaging each token vector with its adjacent tokens using a centered window, which smooths local variations and enhances semantic coherence. Token smoothing affects the granularity of topic segmentation. By averaging each token with its neighbors, local fluctuations in similarity are reduced, which helps produce more coherent, contiguous topic spans and reduces spurious topic assignments. However, this also introduces a trade-off between over-smoothing which can blur sharp transitions between distinct topics, making it harder to detect very fine-grained shifts in topic. The size of the smoothing window controls the balance, narrower window preserves sharper boundaries, while a wider window yields smoother, more stable segments but may merge closely adjacent yet semantically different topics.

After smoothing, each smoothed token vector is then compared to all topic vectors using cosine similarity, and assigned to the most similar topic. By aggregating these token-level assignments, a topic distribution for the entire document is computed. Additionally, the mean cosine similarity between token vectors and their assigned topic vectors measures the relevance of each topic within a document. This creates a topic segmentation at the token level, along with document topic distribution and topic relevance.

The token level topic assignment is done using the contextualized token embeddings of each document and the topic vectors as follows:

1. For each document contextualized token vector c_i in the set $C = \{c_1, c_2, ..., c_n\}$:
    - Using a centered window of size 3, average token vector c_i with adjacent tokens
    - Calculate cosine similarity of pooled c_i with each topic vector t.
    - Assign c_i to the topic vector t with the highest similarity.

2. For each document d:

    - Count the occurrences of each topic based on tokens' assigned topics.
    - Create a topic distribution for document d by dividing the count of each topic by the total number of tokens in d.

3. For each topic t in a document d:

    - Compute the mean cosine similarity between each contextualized token vector c_i assigned to t. This average represents the relevance of topic t in document d.

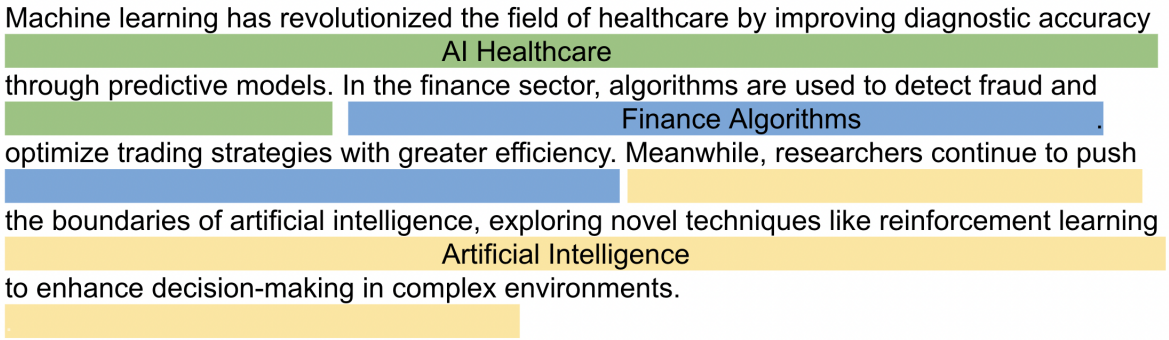


Figure 5.5: This figure illustrates token-level topic assignment and segmentation. Contextualized token embeddings are smoothed using a sliding average, then assigned to the most similar topic vector via cosine similarity. These assignments generate a topic segmentation across the document.

5.8 Discussion

In this chapter, we introduced `C-Top2Vec`, a novel topic modeling approach that enhances `Top2Vec` by leveraging contextual token embeddings to create multi-vector document representations, find topic vectors which capture the multiple topics contained within documents, segment documents into topics at the token level and label topics with phrases rather than just words.

`C-Top2Vec` overcomes the limitations of traditional topic models such as LDA and NMF, which rely on term frequency-based representations and fail to capture contextual

meaning. Unlike these traditional models, `C-Top2Vec` uses dense regions of sub-document embeddings to discover topics, enabling a more fine-grained and semantically coherent topic modeling process. Although traditional methods can find multiple topics per document they do not indicate where in the documents those topics are, in contrast `C-Top2Vec` segments documents into topics at the token level. Additionally with traditional methods the only way to rank the relevance of a document is given their topic distribution, however two documents with the same distribution would be equivalent. The benefit of `C-Top2Vec` is that in addition to topic distribution the topic relevance can be computed using the topic and token vectors which can be very useful in information retrieval tasks.

Compared to `Top2Vec`, which represents each document as a single vector, `C-Top2Vec` introduces multi-vector document representations, allowing for better topic granularity and document segmentation. This enables the identification of multiple topics within a single document. This is very useful for corpora with long documents and documents containing multiple topics.

Despite its advantages, `C-Top2Vec` may not be the optimal choice for all scenarios. For short texts or documents that predominantly contain a single topic, the single-vector representation of `Top2Vec` remains more computationally efficient and sufficient for coarse topic discovery.

Overall, `C-Top2Vec` presents a powerful solution for topic modeling in multi-topic and long-document settings, offering superior topic discovery, segmentation, and labeling capabilities compared to traditional models and `Top2Vec`. However, the choice between `C-Top2Vec` and `Top2Vec` depends on the nature of the dataset and the required level of granularity.

5.9 Summary

In this chapter, we presented `C-Top2Vec`, a topic modeling approach that creates multi-vector document representations to find topic vectors, creates topic hierarchy, labels topics with phrases and segments documents into topic spans. They key ideas were:

- **Contextualized Token Embeddings:** We discussed the self-attention mechanism in Transformer-based models like BERT and SBERT, emphasizing their ability to generate contextualized embeddings. Unlike static word embeddings, these representations depend on the word's context, capturing semantic nuances.

- **Multi-vector Document Representation:** To address the limitations of single-vector document representations, we proposed generating multiple sub-document vectors using a sliding window over contextualized token embeddings. Mean pooling was selected to balance semantic representation and minimize the influence of outliers, ensuring that each token contributes equally to the final embedding.
- **Hierarchical Topic Reduction:** We introduced a method to reduce the number of topics by merging smaller topics into their most semantically similar neighbors. This process uses a size-weighted mean of the topic vectors, ensuring that the resulting topics remain representative of the corpus by biasing larger topics.
- **Topic Labelling:** Topic labels were created by embedding phrases extracted from the corpus using NPMI. Labels are assigned to topics based on cosine similarity between topic vectors and phrase embeddings. This method allows for flexible labeling using either extracted phrases or the dataset vocabulary.
- **Topic Assignment and Segmentation:** We assign topics at the token level by smoothing contextual token embeddings and comparing them to topic vectors using cosine similarity. This approach enables fine-grained segmentation of documents, computation of topic distributions, and assessment of topic relevance within each document.

In the next chapter, we will explore methods for evaluating topic models that go beyond simple coherence and look at how representative topics are of their underlying documents.

Algorithm 5.1 `C-Top2Vec` Algorithm

Input: A set of documents `D`
Output: Topic vectors, topic labels, and document-topic segments.

1. **Contextual token embedding**
   Embed each document using a contextual embedding model to obtain token-level representations.
   - **Model:** `all-mpnet-base-v2` (SBERT)
2. **Multi-vector document representation**
   Create sub-document vectors using a sliding window over contextual tokens.
   - **Method:** Mean pooling with window size = 50, stride = 40
3. **Dimensionality reduction**
   Reduce sub-document vectors to a lower-dimensional space.
   - **Model:** UMAP
   - **Parameters:** `n_neighbors` = 50, `n_components` = 5
4. **Cluster sub-document vectors**
   Identify dense regions using density-based clustering.
   - **Algorithm:** HDBSCAN
   - **Parameter:** `min_cluster_size` = 15
5. **Compute topic vectors**
   Calculate the centroid of each dense cluster in the original embedding space.
   - **Method:** Arithmetic mean of sub-document vectors
6. **Label topics with phrases**
   Assign top-k closest embedded phrases to each topic vector.
   - **Method:** Cosine similarity to embedded NPMI-extracted phrases
7. **Token-level topic assignment**
   Assign each token a topic using smoothed contextual embeddings and topic vector similarity.
   - **Smoothing:** Centered window average with size = 3
   - **Similarity:** Cosine

Chapter 6

Topic Evaluation

6.1 Introduction

Evaluating topic models presents a unique challenge, as no single metric comprehensively captures all aspects of topic quality and interpretability results are subjective depending on the user or use-case of a topic model. Early evaluations primarily relied on perplexity, a measure of statistical likelihood derived from language modeling. While perplexity is useful for assessing the predictive performance of probabilistic generative models, it has been shown to correlate poorly with human interpretability. Models that optimize for lower perplexity may not necessarily produce semantically meaningful topics, therefore creating a need for alternative evaluation techniques.

To improve topic model evaluation, human judgment based evaluations were introduced, such as word intrusion and topic intrusion tasks. These methods assess the semantic coherence of topics by testing whether humans can easily distinguish relevant words and topics from intruding ones. Despite their effectiveness, human evaluations are costly and time-consuming, motivating the development of automated coherence measures that approximate human judgments.

Coherence metrics, such as Pointwise Mutual Information (PMI), Normalized PMI (NPMI), and distributional semantic approaches, have emerged as standard methods for evaluating the semantic consistency of topic models. These measures leverage word co-occurrence statistics from large external corpora or the training dataset itself to assess the degree to which topic words form meaningful groupings. Among them, the C_V coherence measure, which combines NPMI with vector-based similarity, has demonstrated strong alignment with human evaluations.

Most topic evaluation metrics measure topic coherence, but little attention is given to how well the topics represent the underlying documents, which can lead to misleading results [13]. Accessing how well a topic represents the underlying documents is essential to evaluate the performance of a topic model [36]. Lastly, coherence measures designed for older models may not accurately represent the performance of novel neural methods [36], [54].

In this chapter, we propose two new methods that evaluate both topic coherence and how representative topic are of their underlying documents. One method is **Topic Information Gain**, which quantifies how well topics describe documents by measuring the mutual information between documents and their topic words. The other is **BERTScore** which uses contextual token embedding of topics and documents to measure how representative topics are of their underlying documents. This addresses the following thesis objective:

- **Comprehensive evaluation of topic models:** We address RQ5 by proposing novel metrics that assess both topic coherence and how well topics represent their underlying documents. **Topic Information Gain** measures mutual information between topic words and documents, while **BERTScore** leverages contextual embeddings for alignment. These metrics provide a more holistic evaluation of topic models.

6.2 Topic Information Gain

In order to combine topic coherence with evaluation of how representative topics are of their underlying documents, we propose using *mutual information* [28] to measure the information gained about the documents when described by their topic words.

Traditional topic modeling methods describe documents as a mixture of topics. In order to evaluate a set of these topics T generated from documents D, the total information gained is calculated for each document when described with the proportions of topics given by the topic model.

In contrast, `Top2Vec` learns a continuous representation of topics and places documents in that space corresponding to their topic. A topic vector found by `Top2Vec` represents the topic common to a group of documents, or the average of their individual topics. In order to evaluate a set of `Top2Vec` topics T generated from documents D, the documents are partitioned into sub-sets, with each sub-set corresponding to document vectors with

the same nearest topic vector. Thus each document is assigned to exactly one topic. To evaluate these topics, the total information gained is measured for each of the sub-set of documents when described by the words nearest to their topic vector.

The total information gained, or mutual information, about all documents D when described by all words W, is given by:

$$I(D,W) = \sum_{d \in D} \sum_{w \in W} P(d,w) log \left(\frac{P(d,w)}{P(d)P(w)} \right) \tag{6.1}$$

The contribution of each co-occurrence between a document d and word w to the information gain calculation can be seen as the *probability-weighted amount of information* (PWI) d and w contribute to the total amount of information [3], given by:

$$PWI(d,w) = P(d,w) log \left(\frac{P(d,w)}{P(d)P(w)} \right) \tag{6.2}$$

Topics are distributions over the entire vocabulary W. However, in order to evaluate their usefulness to a user, we evaluate them using the top n words of the topic. For evaluation where each document is assigned to only one topic, each topic $t \in T$, will have a set of n words $W_t \subset W$ and documents $D_t \subset D$. The information gained about all documents when described by their corresponding topic is given by:

$$\begin{aligned} PWI(T) &= \sum_{t \in T} \sum_{d \in D_t} \sum_{w \in W_t} P(d,w) log \left(\frac{P(d,w)}{P(d)P(w)} \right) \\ &= \sum_{t \in T} \sum_{d \in D_t} \sum_{w \in W_t} P(d|w) P'(w) log \left(\frac{P(d,w)}{P(d)P(w)} \right) \end{aligned} \tag{6.3}$$

In equation 6.3, $P(w)$ is the marginal probability of the word w across all documents D. It is used to calculate the *log* term, which is the *pointwise mutual information* [26] between w and d. $P'(w)$ is the probability of topic word w, which is used to calculate the *expected mutual information* [28], or the information gained about document d given topic word w. The quantity we are measuring is the information gained about each document given its corresponding topic words as a prior. Therefore $P'(w)$ is 1 and can be omitted [3], which gives rise to:

$$PWI(T) = \sum_{t \in T} \sum_{d \in D_t} \sum_{w \in W_t} P(d|w) log \left(\frac{P(d,w)}{P(d)P(w)} \right) \tag{6.4}$$

Alternatively the equation can be generalized for the case that each document is represented by multiple topics. In this case we replace $P'(w)$ with $P(t)$, which is the proportion of words to be used to represent document d by topic t:

$$PWI(T) = \sum_{d \in D} \sum_{t \in T} \sum_{w \in W_t} P(d|w)P(t) log \left(\frac{P(d,w)}{P(d)P(w)} \right) \tag{6.5}$$

Using Equations 6.4 and 6.5, different sets of topics can be compared. A greater quantity of information gain indicates that the topics $t \in T$ are more informative of their corresponding documents. If topics contain words such as *the*, *and*, and *it* or other intuitively uninformative words, they will receive lower information gain values. This is in large part due to the $P(d|w)$ term in the calculation, since the probability of any specific document given a very common word is very low. Therefore, the information gained is also low. Words that are mostly present in the subset of documents corresponding to the topic lead to higher information gain as they are informative of those documents. Additionally, low values of information gain will be obtained if the topic model assigns topics to the wrong documents. Topic information gain measures the quality of the words in the topic and their assignment to documents. Therefore, Equations 6.4 and 6.5 give values that correspond with what is intuitively more informative.

Topic Information Gain improves upon traditional topic coherence measures by incorporating how well topics represent their underlying documents. Unlike coherence metrics that focus solely on word co-occurrence within topics, this approach measures the mutual information between topic words and their corresponding documents, ensuring that the topics are both meaningful and representative. By penalizing frequent, uninformative words and rewarding topic words that are specific to their assigned documents, Topic Information Gain provides a more comprehensive evaluation of topic models. This metric captures both topic coherence and topic representativeness leading to a more holistic assessment of topic quality.

Despite its advantages, Topic Information Gain does have major limitations: computational expense and lack of semantic understanding. Calculating mutual information for each topic-word and document pair requires extensive probability computations, making it

resource-intensive and less scalable for large datasets. Additionally, because it relies purely on word co-occurrence statistics, it cannot directly evaluate semantic similarity, which is what determines whether a topic is representative of its documents. If topic words are conceptually related to document words but do not frequently co-occur, they will not contribute to the score, limiting its ability to fully capture topic quality.

6.3 BERTScore

Most topic model evaluations focus on topic coherence and do not directly evaluate how representative topics are of documents of that topic. A topic could be coherent but the wrong documents could be assigned to that topic or it may not represent topical content accurately. While Topic Information Gain improves upon traditional coherence measures by incorporating document representativeness, it remains limited in its ability to capture semantic relationships. Since it relies on word co-occurrence statistics, it cannot evaluate whether topic words are conceptually related to their assigned documents beyond simple frequency-based associations. In order to address this gap, we propose using BERTScore [118] to evaluate both topic coherence and how representative the topics are of their underlying documents. Unlike traditional coherence measures and Topic Information Gain, BERTScore captures the *semantic similarity* between topic words and their corresponding documents using contextualized embeddings.

BERTScore was initially proposed to evaluate text generation and has been shown to correlate well with human judgements [118]. It has also been used to evaluate abstract summaries of topics represented as phrases or multiple words [82]. Given a reference and a candidate sequence, BERTscore uses contextualized token embeddings of each to measure the similarity between candidate and reference tokens.

BERTScore computes similarity between a candidate sequence and a reference sequence. It does so by calculating pairwise cosine similarities between the contextualized tokens of each sequence. Unlike frequency-based methods, BERTScore can handle synonymous and contextually related words, allowing it to better reflect human understanding of semantic similarity.

BERTScore computes a recall, BERTScore_R, precision, BERTScore_P, and F1 score, BERTScore_{F1}. We use the recall as a measure of topic coherence, as it evaluates how well each of the topic words is represented in the documents. We use precision to measure how representative the topic is of the document, as it evaluates how well the contents of the documents are captured by the topic words.

To evaluate topics, we create contextualized embeddings for the top N topic words of each topic. It is necessary to select a top D documents per topic, due to the computational expense of computing BERTScore. We also produce contextualized token embeddings for documents of each topic. For each topic, we calculate the cosine similarity between its contextualized token embeddings and the embeddings of the tokens from each topic document. We compute two core BERTScore metrics:

- BERTScore$_R$: Measures how well the topic words are represented in the document. This serves as an alternative measure of topic coherence by evaluating whether the words used to define a topic are semantically aligned with the document content.
- BERTScore$_P$: Measures how well the document content is captured by the topic words. This evaluates how representative the topic is of the document, ensuring that topic words accurately summarize the content they are assigned to.

BERTScore uses maximum similarity as shown in Figure 6.1, so for each topic word, we take the most similar token from the topic document to calculate BERTScore. The greedy matching allows BERTScore to be used even though topic descriptions have fewer tokens than documents and allows it to handle documents of varying lengths. To compute BERTScore, a weighted average of the max scores is taken using the Inverse Document Frequency (IDF), which puts less weight on stop-words and more weight on informative words.

BERTScore$_R$ is computed by taking the maximum of cosine similarities for each of the topic's contextualized tokens. This approach ensures that for each token in the topic, its best representation in the document is recognized. BERTScore$_P$ is computed by taking the maximum of cosine similarities for each of the document's contextualized tokens. This ensures that each token in the document is matched with its closest counterpart in the topic, reflecting how accurately the document tokens are represented by the topic tokens. We average these scores over all topics and their documents to evaluate a model.

Unlike Topic Information Gain, which penalizes frequent words based on their probability distributions, BERTScore uses Inverse Document Frequency (IDF) weighting to down-weight common words while emphasizing informative words. Additionally, BERTScore uses a *greedy matching* strategy, selecting the highest similarity score for each token. Another essential feature of BERTScore is that the tokens are contextualized, so they capture information from their surrounding words. As a result, selecting the highest similarity token still preserves the broader contextual meaning of both the topic labels and documents, allowing BERTScore to account for semantic nuances and contextual dependencies. This

enables it to account for word substitutions and phrase variations, making it particularly effective for evaluating topic models in diverse linguistic contexts.

Despite its advantages, BERTScore has limitations, primarily its computational expense. Because it relies on contextualized embeddings from transformer-based models like BERT, computing pairwise token similarities between topic words and document tokens requires significant memory and processing power. This makes it less scalable for large datasets or models with many topics and long documents. Additionally, while BERTScore captures semantic similarity, its performance depends on the choice of the underlying language model, which may not always align perfectly with domain-specific terminology. These computational demands and dependencies can make BERTScore impractical for large-scale topic model evaluations.

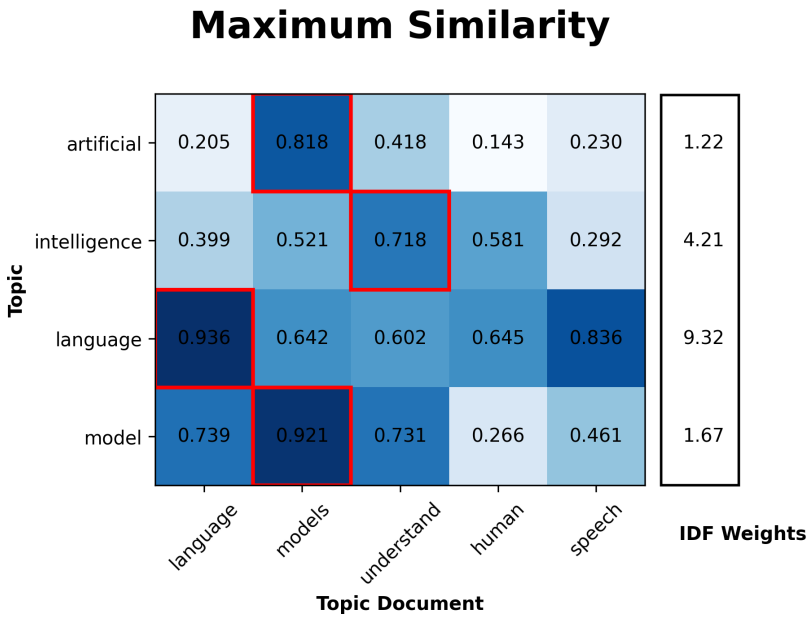
6.4 Summary

In this chapter, we looked at methods for evaluating topic models beyond traditional coherence measures by incorporating document representativeness. The key ideas discussed were:

- **Challenges in Topic Evaluation:** Traditional evaluation methods such as perplexity and coherence metrics like PMI, NPMI, and C_V fail to fully capture both topic coherence and representativeness. While human evaluation methods such as word intrusion provide reliable assessments, they are costly and time-intensive.
- **Topic Information Gain:** We proposed a novel evaluation metric that uses mutual information to quantify how well topics describe documents. Topic Information Gain rewards informative words while penalizing common, uninformative ones, ensuring that topics are both meaningful and representative of their corresponding documents.
- **BERTScore for Topic Evaluation:** To address the limitations of frequency-based methods, we proposed to use BERTScore, which leverages contextual embeddings to measure semantic similarity between topic words and documents. By computing precision and recall metrics, BERTScore captures both topic coherence and document representativeness more effectively than traditional methods.
- **Limitations of the Proposed Methods:** While Topic Information Gain improves upon coherence-based evaluations, it lacks semantic understanding and remains computationally expensive. Similarly, BERTScore is more powerful in capturing semantic

relationships but is also computationally intensive and dependent on pre-trained language models, which may not always align with domain-specific topics.

In the next chapter, we will apply Topic Information Gain, BERTScore, and other evaluation metrics to assess several topic modeling approaches, comparing their performance on topic coherence, topic assignment accuracy, and document representativeness.



$$\text{Score} = \frac{(0.818 \times 1.22) + (0.718 \times 4.21) + \ldots}{1.22 + 4.210 + 9.32 + 1.67}$$

Figure 6.1: This figure shows a heatmap of BERTScore-based similarity between topic words and documents. BERTScore leverages contextual embeddings to assess semantic alignment. Each score reflects the weighted similarity between topic words and the documents they are meant to represent, offering a richer evaluation of topic quality by considering semantic overlap rather than just word co-occurrence.

Chapter 7

Experiments and Results

7.1 Datasets

Several benchmark datasets are widely used for evaluating topic modeling algorithms. These datasets vary in terms of size, domain, and the number of topic labels, offering diverse challenges for assessing model performance. Below are some of the most commonly used datasets in topic modeling research:

- *20 Newsgroups* dataset [88]: Contains around 18,000 posts from 20 newsgroups, each on a different topic. It is popular due to the diverse range of topics and associated labels.
- *Reuters-21578* dataset [45]: Contains around 20,000 news articles from the Reuters news-wire, with multiple labels per document.
- *AG News* dataset [120]: Contains roughly 130,000 news articles with 4 different topic labels.
- *M10* dataset [86]: Contains roughly 10,000 documents that are scientific publications, each assigned to one of 10 distinct topic labels.
- *BBC News* dataset [40]: Contains roughly 2,000 articles categorized into 5 distinct topic labels.
- *Yahoo! Answers* dataset [119]: Contains roughly 1,400,000 questions and their answers. The original dataset can be acquired from the Yahoo! Webscope program.

Our goal is to evaluate not just topic coherence, but also how representative topics are of their underlying documents. An essential criterion of our dataset selection process was to have ground truth on topic labels so that we can perform cluster evaluation. There are many other available datasets, but based on our review of the literature, the datasets *20newsgroups* and *Yahoo! Answers* best met our criteria. We chose these two datasets due to their size and number of topic labels. All other available datasets were either too small, contained too few labels, or did not have topic labels.

We evaluated the topic models on two datasets: *20Newsgroups* [88] and *Yahoo! Answers* [119], selected for their large number of labelled topics. Both datasets contain posts on various topics with labels, 20 topics for *20newsgroups* and 10 topics for *Yahoo! Answers*. Due to the size of *Yahoo! Answers* we take a stratified sample of 50,000 documents, and we use all 18,846 from *20Newsgroups*. We filter the datasets by removing any empty documents as well as the documents shorter than 35 characters, as they did not contain any meaningful text.

For our clustering experiments, we constructed a dataset derived from the *CCNEWS* corpus [43], which contains over 100 million news articles from a wide range of sources and categories. We filtered the dataset to include only articles published in 2024, written in English, and belonging to 50 distinct categories. From these, we sampled a total of 50,000 articles to form our final dataset.

7.2 Evaluation Metrics

Normalized Pointwise Mutual Information C_{NPMI} is used to evaluate topic coherence by measuring co-occurrence of the top topic words within documents from the actual corpus. It is a widely used metric for topic coherence evaluation that has been shown to correlate with human judgements [95].

Topic Coherence C_V also looks at the top topic words of a topic but it calculates NPMI of words using a sliding window rather than at the document level. [95].

Word Embedding Coherence C_{WE} We use pre-trained `word2vec` [78] embeddings trained on a one-hundred-billion word corpus. Using the embeddings we find the average pairwise cosine similarity of top topic words as proposed by [34] and used by [14]. This is intended to evaluate the coherence of the words relative to an external corpus.

SBERT Word Embedding Coherence C_{SBERT} We also propose a modification the the word embedding coherence by using an SBERT [93] model to embed words instead of `word2vec`. This leverages the advancements in neural architectures and also allows for embedding phrases rather than just words. We use this approach to also calculate the average pairwise cosine similarity of top topic words and phrases.

Adjusted Rand Index ARI proposed by [55] measures the similarity between two clusterings by comparing the cluster labels only. We use it to evaluate how well topic models assign documents to topics compared to the ground-truth labels from the reference dataset. The reference clusters are the true topic labels of each document from the labelled dataset which are compared against the clusters formed by the topic model topic assignment of each document.

Adjusted Mutual Information AMI proposed by [108] measures the similarity of two clusterings by comparing cluster labels only. It can be more effective when dealing with unbalanced cluster sizes and small clusters as compared to ARI. We use it in the exact same way as ARI to evaluate document topic assignments with the true labels as a reference.

Cluster Homogeneity: Purity measures how well each cluster contains data points from a single ground truth class. It assigns each cluster to the most frequent class within it and calculates the overall proportion of correctly assigned data points.

BERTScore is used to evaluate topic coherence and how well topics represent their associated documents by leveraging contextualized token embeddings. It computes recall (BERTScore_R) to measure topic coherence by assessing how well topic words appear in documents and precision (BERTScore_P) to evaluate how representative topics are of their documents. By calculating the cosine similarity between contextualized embeddings of topic words and document tokens, BERTScore uses a greedy matching approach with Inverse Document Frequency (IDF) weighting to prioritize informative words. This allows for robust topic evaluation even with varying document lengths and concise topic descriptions.

7.3 Clustering Experiments

Neural clustering-based topic modeling relies on the ability to cluster documents into meaningful topics by leveraging dense vector representations which encode semantic similarities.

Therefore, selecting an appropriate clustering method is very important, as clustering algorithms have different implicit assumptions about cluster shapes, densities, and the number of clusters. We evaluate clustering methods on document embeddings with known labels to provide a direct measure of how well each clustering method can capture semantic coherence. This allows us to evaluate how well the methods can identify areas of similar documents in the embedding space.

We use `all-mpnet-base-v2` [93, 98] to generate document embeddings. Our experiments are conducted on the `YahooAnswers` and `20Newsgroups` datasets. We evaluate each clustering using ARI, AMI, and Purity using the labels from the datasets as ground truth labels. Each clustering method is run for 10 iterations, and we report the mean and standard deviation of the results in Table 7.1.

All clustering methods use default parameters, except for HDBSCAN, where we set the minimum cluster size to 15 instead of 5. For dimensionality reduction, we configure UMAP to reduce embeddings to 5 dimensions instead of 2, as we do not intend to visualize the vectors and wish to avoid excessive compression. For T-SNE, we use 3 dimensions, as higher dimensions significantly increase computational complexity to $O(N^2)$.

For PCA, we reduce the embeddings to 50 dimensions. We experimented with lower-dimensional representations, but the clustering performance deteriorated significantly. This decline occurs because PCA is a linear transformation that retains the components with the highest variance, and reducing the dimensions too aggressively discards too much information. Unlike UMAP, which preserves local and global structures in a nonlinear manner, PCA strictly captures global variance, thus excessive dimensionality reduction leads to a loss of fine-grained semantic information.

7.3.1 Traditional Clustering Methods

To establish a baseline for clustering, we first apply K-means to the document embeddings in their original high-dimensional space. While K-means is widely used due to its efficiency and simplicity, its reliance on spherical cluster assumptions can be problematic in high-dimensional settings where the curse of dimensionality reduces meaningful distance relationships. We additionally use Agglomerative Clustering on the original dimensional data as an additional baseline. The results in table 7.1 show the results from the clusterings on the ARI, AMI and Purity cluster evaluation methods.

We attempted to apply HDBSCAN directly to the high-dimensional embeddings; however, even on the relatively smaller `YahooAnswers` dataset, it takes roughly 30 minutes to run, which is significantly longer than the seconds it typically requires for lower-dimensional

data. This shows that using HDBSCAN on much larger datasets would be computationally too expensive, making it unsuitable for large-scale topic modeling. Additionally, the algorithm consistently returned only a few clusters, indicating that it struggles to identify meaningful density variations in high-dimensional space. Similarly we also tried to apply DBSCAN directly on the high-dimensional embeddings, although it was much faster it still only finds a handful of clusters with the default *epsilon* setting. We tried running DBSCAN with a range from 0.1 to 1 of *epsilon* values and none produced meaningful results. These experiments demonstrate that density based clustering in high-dimensional space is challenging and confirms the need for dimensionality reduction.

We then evaluate whether reducing the dimensionality of the embeddings improves clustering performance. We experiment with PCA, T-SNE and UMAP. We first evaluate PCA followed by K-means clustering. The results indicate that dimensionality reduction does not always lead to improvements in clustering performance. In both datasets, PCA followed by K-Means performs slightly worse than applying K-means directly in the original high-dimensional space. These results suggest that while PCA effectively reduces dimensionality, it may not preserve the most meaningful semantic structures of document embeddings. Since PCA focuses on finding dimensions with maximum variance in a linear fashion, it might struggle to capture the complex nonlinear relationships present in high-dimensional embeddings. We find that applying T-SNE before K-means clustering does not significantly improve performance, likely due to its focus on local neighborhood preservation. However, using UMAP for dimensionality reduction, we observe a notable improvement in clustering performance with k-means as shown in 7.1. This suggests that UMAP is better suited for preserving the semantic structure of document embeddings in a lower-dimensional space. We also show that applying UMAP before agglomerative clustering also improves the clustering performance on both datasets.

We finally evaluate the performance of HDBSCAN following UMAP-based dimensionality reduction. Unlike k-means, which enforces a fixed number of clusters, HDBSCAN is a density-based clustering algorithm that can identify clusters of varying shapes and sizes while also filtering out noise points. Our results show that UMAP followed by HDBSCAN has the highest clustering performance across the evaluation metrics. The improvement is most evident in the `20newsgroups`, where this approach significantly outperforms k-means alone. However, on the Yahoo Answers dataset, while the Adjusted Mutual Information (AMI) score remains competitive, the Adjusted Rand Index (ARI) decreases slightly. Since AMI evaluates pairwise alignment, the lower ARI but high AMI and purity suggest that HDBSCAN is identifying dense clusters that share the same label. This indicates that its is effectively capturing subtopics within broader categories rather than incorrectly clustering documents. As a result, the findings remain strong, as shown by the high AMI and cluster

purity scores.

Model	ARI	AMI	Purity
	20newsgroups		
Original Dimension + K-Means	0.442 ± 0.014	0.582 ± 0.008	0.581 ± 0.008
Original Dimension + Agglom	0.361 ± 0.000	0.548 ± 0.000	0.541 ± 0.000
PCA + K-Means	0.424 ± 0.006	0.580 ± 0.005	0.577 ± 0.004
T-SNE + K-Means	0.413 ± 0.004	0.561 ± 0.002	0.561 ± 0.002
UMAP + K-Means	0.479 ± 0.006	0.620 ± 0.003	0.618 ± 0.002
UMAP + Agglom	0.472 ± 0.003	0.616 ± 0.002	0.610 ± 0.001
UMAP + HDBSCAN	$\mathbf{0.617 \pm 0.027}$	$\mathbf{0.697 \pm 0.001}$	$\mathbf{0.772 \pm 0.022}$
	YahooAnswers		
Original Dimension + K-Means	0.293 ± 0.009	0.355 ± 0.006	0.353 ± 0.006
Original Dimension + Agglom	0.242 ± 0.000	0.355 ± 0.000	0.344 ± 0.000
PCA + K-Means	0.284 ± 0.001	0.348 ± 0.001	0.346 ± 0.001
T-SNE + K-Means	0.313 ± 0.002	0.367 ± 0.001	0.367 ± 0.001
UMAP + K-Means	$\mathbf{0.377 \pm 0.001}$	$\mathbf{0.439 \pm 0.001}$	0.437 ± 0.001
UMAP + Agglom	0.345 ± 0.009	0.418 ± 0.005	0.413 ± 0.005
UMAP + HDBSCAN	0.249 ± 0.054	$\mathbf{0.451 \pm 0.008}$	$\mathbf{0.680 \pm 0.004}$

Table 7.1: Performance comparison of clustering methods across two datasets (`20newsgroups` and `YahooAnswers`) using various combinations of clustering algorithms. The models are evaluated using Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and Cluster Purity. Results demonstrate that applying UMAP before clustering significantly improves clustering quality, particularly when followed by HDBSCAN, which achieves the best scores on the `20newsgroups` dataset. UMAP also enhances the performance of K-Means and Agglomerative Clustering across both datasets.

7.3.2 Deep Clustering Methods

Lastly, we explore the use of deep clustering techniques and evaluate their potential for clustering-based topic modeling. Table 7.2 summarizes the results of various deep clustering methods.

For DEC, DCN, and DipDECK, we apply K-means clustering to the learned latent space as a post-processing step in order to get clusters. This is necessary because these

deep clustering models primarily learn a feature representation optimized for clustering but do not explicitly assign cluster labels. By applying K-means afterward, we can obtain discrete cluster assignments. Since these models require a predefined number of clusters, the choice of k still influences the final results. For the `20newsgroups` dataset we use a k of 20 and for `YahooAnswers` we use a k of 10. In both cases the k is the number of ground truth labels allowing us to evaluate the clustering performance against known topic labels.

For Top-k SAE, we do not apply K-means for clustering. Instead, we set the latent dimesion to the number of underlying labels, with the assumption that each dimension represents the underlying topic. This makes the latent vectors inherently interpretable. To assign a cluster to each document, we select the dimension with the highest activation as the assigned cluster. This approach directly maps each document to the most strongly activated latent without requiring an additional clustering step, simplifying the process and reducing reliance on external clustering algorithms.

Compared to traditional clustering methods, deep clustering approaches show mixed results. DEC performs the worst among the deep clustering techniques, likely due to to the fact that after pretraining the decoder is discarded, and only encoder is fine-tuned using a clustering objective. This limits its ability to learn semantically coherent representations in the latent space, as the network no longer learns to reconstruct the original input. DCN, which integrates K-means optimization directly into its learning process, and uses reconstruction loss, outperforms DEC and achieves competitive results, particularly on the Yahoo Answers dataset. However, its reliance on specifying k in advance remains a significant limitation.

DipDECK demonstrates an improvement over DEC by dynamically estimating the number of clusters, but it does not perform as well as DCN. Its reliance on an initial overestimation of k and threshold selection for merging clusters could contribute to its lower performance.

Top-k SAE outperform DEC and DipDECK on both datasets, demonstrating the benefits of enforcing sparsity in the latent space for uncovering topic structures. However, Top-k SAE does not surpass DCN, which remains the strongest deep clustering method overall. On the `20newsgroups` dataset, Top-k SAE achieves a higher ARI than DEC and DipDECK while maintaining competitive AMI and Purity scores, though it falls short of DCN. Similarly, on the `YahooAnswers` dataset, it underperforms DCN in all metrics, suggesting that while sparsity helps improve clustering performance, it does not perform as well as deep clustering methods explicitly optimized for K-means clustering. These results highlight the potential of sparse representations for topic modeling while also revealing their limitations, particularly in comparison to more structured clustering approaches like

DCN.

Overall, deep clustering techniques do not outperform the combination of UMAP and HDBSCAN when applied to text embedding data. DEC and DCN perform worse than K-means clustering applied directly to the original high-dimensional document embeddings, suggesting that their learned representations do not effectively enhance clustering performance for textual data. Additionally, applying UMAP for dimensionality reduction followed by K-means already provides a reasonable improvement over clustering in the original space. However, the best performance is consistently achieved with UMAP and HDBSCAN, which outperforms all other methods. These results indicate that for text embeddings, non-deep clustering approaches, when combined with effective dimensionality reduction, offer the best balance of scalability, adaptability, and clustering quality, making them the most suitable choice for topic modeling. However, deep clustering methods have shown greater success in other domains, such as image data, where learned feature representations can improve clustering performance. Thus, deep clustering methods remain a promising area for future research. While current approaches may not yet yield superior results for textual data, they hold potential. Further exploration and development of these methods may lead to new techniques that better leverage the structure of text embeddings for improved clustering performance.

Model	ARI	AMI	Purity
	20newsgroups		
DEC + K-Means	0.140±0.041	0.327±0.045	0.343±0.049
DCN + K-Means	0.386±0.014	**0.583 ± 0.011**	**0.594 ± 0.011**
DipDECK + K-Means	0.194±0.020	0.392±0.0230	0.399±0.0230
Top-k SAE	**0.405 ± 0.008**	0.552±0.006	0.556±0.007
	YahooAnswers		
DEC + K-Means	0.090±0.004	0.163±0.002	0.159±0.004
DCN + K-Means	**0.278 ± 0.010**	**0.403 ± 0.002**)	**0.362 ± 0.002**
DipDECK + K-Means	0.155±0.030	0.272±0.032	0.249±0.029
Top-k SAE	0.216±0.015	0.290±0.014	0.293±0.013

Table 7.2: Performance comparison of deep clustering methods on document embeddings for the `20newsgroups` and `YahooAnswers` datasets. Results show that DCN consistently outperforms other deep models, particularly on YahooAnswers, while Top-k SAE provides interpretable clustering through sparse latent representations. However, deep clustering methods generally underperform compared to traditional clustering using UMAP and HDBSCAN. These findings suggest that while deep clustering has potential, especially in other domains like image analysis, it does not yet surpass the effectiveness of traditional methods for clustering-based topic modeling on text embeddings.

7.3.3 Optimizing UMAP and HDBSCAN Parameters

The performance of the `Top2Vec` algorithm is sensitive to the hyperparameters of both UMAP and HDBSCAN. One of the most important parameters for UMAP is `n_neighbors`, which controls the balance between preserving local and global structure in the reduced embedding dimensions. A lower `n_neighbors` value preserves more local structure, which would capture granular semantic relationships in the embedding space. Conversely, a higher value of `n_neighbors` preserves more global structure, which would capture broader semantic relationships.

Since HDBSCAN relies on the density in the UMAP-reduced space, the choice of `n_neighbors` directly affects its ability to discern meaningful clusters from noise. In the extreme, too small values of `n_neighbors` can lead to too granular clusters that fragment semantically coherent regions where as large values might lead to semantically divergent areas to be merged close together. However, if tuned properly, the `n_neighbors` will control the tradeoff between presercing granular semantic information or broader themes.

Another important parameter in UMAP is `n_components`, which determines the reduced dimensionality of the embedding. Reducing the data to too few dimensions could discard important semantic structure, leading to an oversimplified representation. On the other hand, a higher number of components can preserve richer structure but might also be too sparse and HDBSCAN could still suffer from the curse of dimensionality. The optimal setting for `n_components` is a trade-off between dimensionality reduction and the preservation of discriminative semantic structure necessary for clustering with HDBSCAN.

In addition, HDBSCAN has an important hyperparameter, `min_cluster_size`, which sets the minimum number of points required to form a cluster and determines the granularity of the clustering. A lower `min_cluster_size` can lead to the formation of many small clusters, which might capture subtle subtopics but also risk creating spurious clusters that don't really reflect an underlying topic. In contrast, a higher `min_cluster_size` forces the algorithm to merge smaller clusters into larger, more robust ones, potentially overlooking meaningful distinctions between topics. The ideal value for `min_cluster_size` depends on a tradeoff between what amount of points are labelled as noise, the granularity of the clustering and the coherence of the resulting clusters.

HDBSCAN labels dense regions and also identifies areas of noise, which are generally sparser regions. Since we apply HDBSCAN on the UMAP-reduced embedding, we explore parameters for both simultaneously. The evaluation aims to find the parameters that allow HSDBSCAN to assign clusters to the most amount of points while also keeping semantically similar regions in the same clusters. We specifically focus on optimizing `n_neighbors`, `n_components`, and `min_cluster_size`. In our experiments, we assess clustering quality using several metrics:

- **Clustered Proportion:** This measures the proportion of data points assigned to a cluster rather than being labeled as noise. A high value is desirable, as an excessive number of unclassified points can indicate that the clustering parameters are suboptimal.
- **Cluster Homogeneity (Purity):** This metric evaluates how well the clusters contain data points from a single ground truth category. High homogeneity indicates that each cluster is coherent in terms of its semantic content.
- **ARI and AMI:** Both metrics assess the overall agreement between the clustering output and the known labels.

For optimal topic modeling results, all of these metrics should be high. It is particularly important to avoid cases where HDBSCAN assigns nearly all points to one or few clusters,

thereby inflating certain scores, while failing to capture the underlying topic structure. By tuning `n_neighbors` and `n_components` in UMAP, along with `min_cluster_size` in HDBSCAN, and observing the resulting changes in performance across these evaluation metrics, we can identify a parameter configuration that effectively balances the preservation of local structure with the retention of global structure, leading to accurate and interpretable topic representations.

For our experiments, we vary each hyperparameter between 2 and 100, evaluating all combinations by running each setting 10 times to account for the stochasticity of UMAP and HDBSCAN. We then compute the average values of the four key metrics: *Clustered Proportion*, *Purity*, *AMI*, and *ARI*.

We compute a single *Clustering Score* by taking the mean of the min-max scaled values of clustered proportion (C), purity (P), Adjusted Mutual Information (AMI), and Adjusted Rand Index (ARI). Each metric is scaled to the $[0, 1]$ range based on its minimum and maximum values across all parameter configurations using the following min-max scaling:

$$\tilde{M} = \frac{M - M_{\min}}{M_{\max} - M_{\min}}$$

where M is the original metric value, and $M_{\min}$ and $M_{\max}$ are the minimum and maximum values of that metric across all configurations. If the clustered proportion falls below 0.1, indicating that HDBSCAN has classified most points as noise, we set the score to 0, as the remaining metrics become unreliable. The Clustering Score for a given parameter configuration is then defined as:

$$\text{Clustering Score} = \begin{cases} 0 & \text{if } C < 0.1 \\ \text{mean}\left(\tilde{C}, \tilde{P}, \tilde{AMI}, \tilde{ARI}\right) & \text{otherwise} \end{cases}$$

where:

- C is the clustered proportion, i.e., the fraction of points assigned to clusters (not labeled as noise by HDBSCAN),
- P is the clustering purity,
- $\tilde{C}$, $\tilde{P}$, $\tilde{AMI}$, and $\tilde{ARI}$ are the min-max scaled values of each respective metric.

We then generate heatmaps for each combination of `n_neighbors`, `n_components`, and `min_cluster_size`. For each value of `n_neighbors`, `n_components`, and `min_cluster_size`, we also take the maximum Clustering Score observed across combinations with the other parameters to show the optimal value for each hyperparameter in a single plot.

We analyze the clustering results across three datasets: *20Newsgroups* (20 topics), *YahooAnswers* (10 topics), and *CCNEWS* (50 topics). Our analysis is based on the heatmaps shown in Figures 7.1, 7.2, and 7.3, as well as the maximum clustering score plots in Figures 7.4, 7.5, and 7.6. We focus on identifying the optimal values of the hyperparameters `n_neighbors`, `n_components`, and `min_cluster_size`, and examine how these values affect the clustering score.

n_neighbors In the *YahooAnswers* dataset (10 topics), clustering performance improves with increasing `n_neighbors`, peaking at 50. However, performance is already close to optimal at lower values around 15, suggesting that while global structure helps, even smaller neighborhood sizes are sufficient to capture the broad topic distinctions. For the *20Newsgroups* dataset (20 topics), the best results are achieved within a range of 15–25, where the balance between local neighborhood preservation and global structure enables clearer separation of mid-level topic granularity. In the *CCNEWS* dataset (50 topics), clustering scores remain high across a wide span of `n_neighbors` values from 15 to 75, with performance already near its peak at 15 and only marginally increasing beyond that. This suggests that the HDBSCAN can effectively find the dense areas that are semantically coherent even with more focus on local structure being preserved with smaller `n_neighbors` values. Increasing `n_neighbors` and capturing more global structure offers only incremental gains, beyond a value of 15, with the exception of the *20Newsgroups* dataset. Overall, these results indicate that the default `n_neighbors` value of 15 is a reasonable choice in most cases and can be further fine-tuned depending on the desire for finding local versus global structure.

n_components. The effect of `n_components`, which controls the dimensionality of the UMAP-reduced space, also reflects the amount of information preserved from the original dimension. In *YahooAnswers* (10 topics), performance steadily improves with higher values, peaking at 50, as greater dimensionality helps preserve the semantic structure across broader topic categories. Similarly, in *CCNEWS* (50 topics), performance peaks at 50 components, but is consistently high for smaller values. However, in *20Newsgroups* (20 topics), performance peaks earlier—around 15 to 25 components—and tends to decline beyond that likely due to the introduction of noise and sparsity in a smaller dataset with high

topic granularity. These results suggest that while higher dimensionality helps with broad or diverse topics, moderate values (15-25) are reasonable defaults, especially for smaller datasets.

min_cluster_size. Across all datasets, smaller values of `min_cluster_size`, particularly around 15-25, generally yield higher clustering scores. A value of 15 emerges as a strong default, as it gives a balance between granularity and cluster coherence. In datasets with a larger number of topics, such as *20Newsgroups* and *CCNEWS*, lower values allow HDBSCAN to identify fine-grained topic structures and avoid merging distinct topics into overly broad clusters. However, in datasets with fewer topics—such as *YahooAnswers*, which has only 10 ground-truth labels—larger values of `min_cluster_size` can perform better by reducing over-fragmentation and aligning more closely with the broader topic definitions. It is also important to note that evaluation metrics like AMI and ARI penalize excessive topic fragmentation, which can disproportionately affect results when the ground-truth label set is small. As such, lower values of `min_cluster_size` may still capture meaningful subtopics, even if they yield lower scores.

Figures 7.7, 7.8, and 7.9 show that increasing `min_cluster_size` consistently leads to a reduction in the average number of clusters across all three datasets. This is because HDBSCAN merges smaller clusters into larger ones as the minimum size threshold increases, thereby reducing clustering granularity. Conversely, lower values of `min_cluster_size` enable HDBSCAN to identify a greater number of finer-grained clusters, which can capture more nuanced subtopics and semantic distinctions. Therefore this parameter serves as a useful control for adjusting topic resolution: smaller values lead to detailed, fragmented topic structures, while larger values produce coarser, more general topics.

In addition to controlling the trade-off between local and global structure in the UMAP embedding, the `n_neighbors` parameter also implicitly controls the number of clusters discovered by HDBSCAN. Smaller values of `n_neighbors` emphasize local structure in the low dimensional space, enabling the HDBCSAN to identify many small, dense clusters that often correspond to fine-grained or specific subtopics. As `n_neighbors` increases, UMAP increasingly preserves global structure, smoothing over local variations and leading to more consolidated clusters. This inverse relationship between `n_neighbors` and the number of clusters is consistent across all the datasets. As shown in Figures 7.10, 7.11, and 7.12, increasing `n_neighbors` results in a decrease in the average number of clusters, regardless of the value of `min_cluster_size`. This trend demonstrates the role of `n_neighbors` in adjusting the resolution of topics: low values lead to detailed and fragmented topic structures, while high values encourage the emergence of coarser, high-level themes.

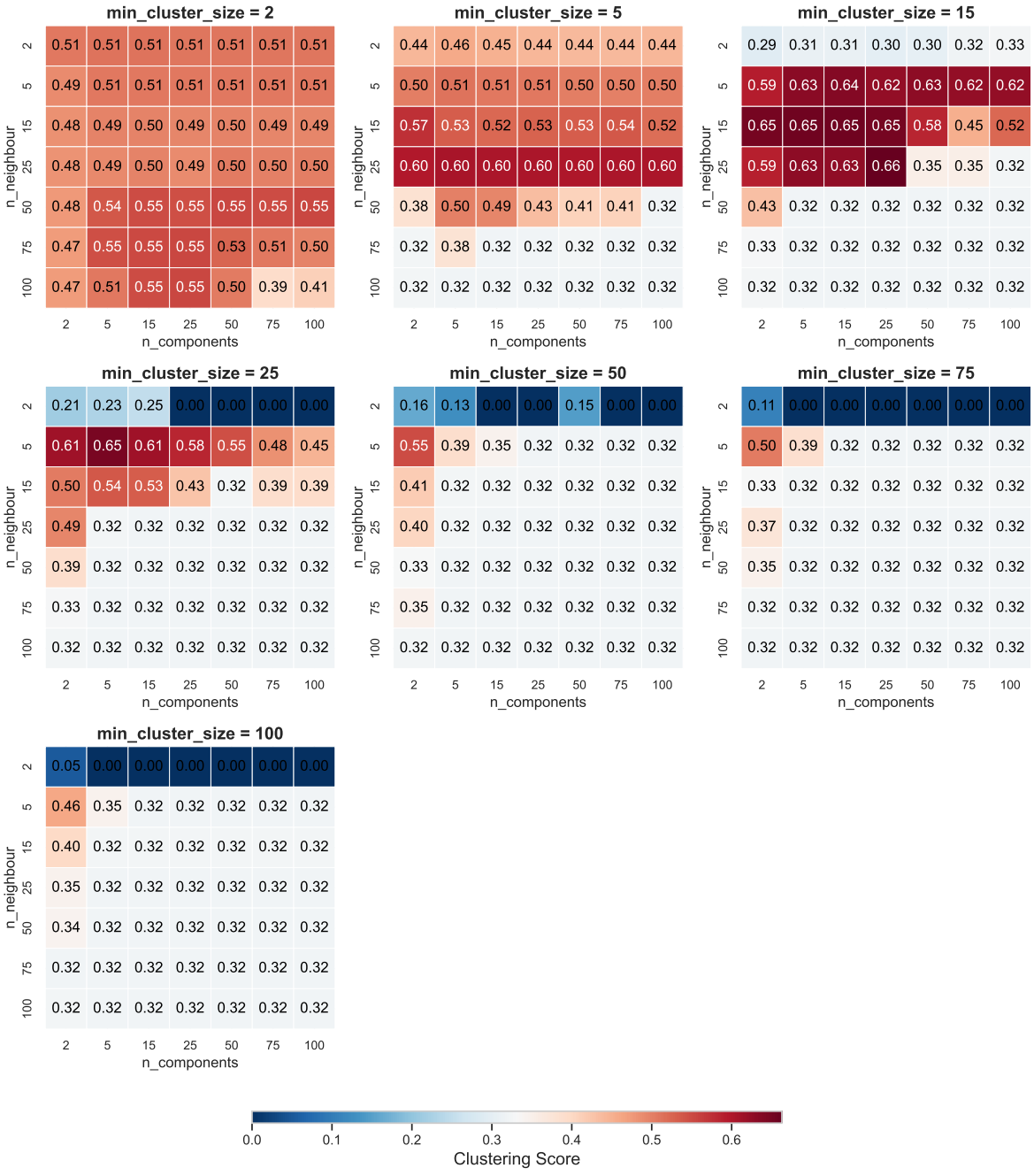


Figure 7.1: Heatmaps showing the Clustering Score across combinations of UMAP and HDBSCAN parameters on the *20Newsgroups* dataset. Each subplot corresponds to a different value of `min_cluster_size` (ranging from 2 to 100). The axes represent values of `n_neighbors` and `n_components`, and the color intensity reflects the Clustering Score, computed as the mean of min-max scaled clustering quality metrics (Clustered Proportion, Purity, AMI, and ARI). These heatmaps reveal how topic modeling performance varies with different granularity and structural preservation settings.

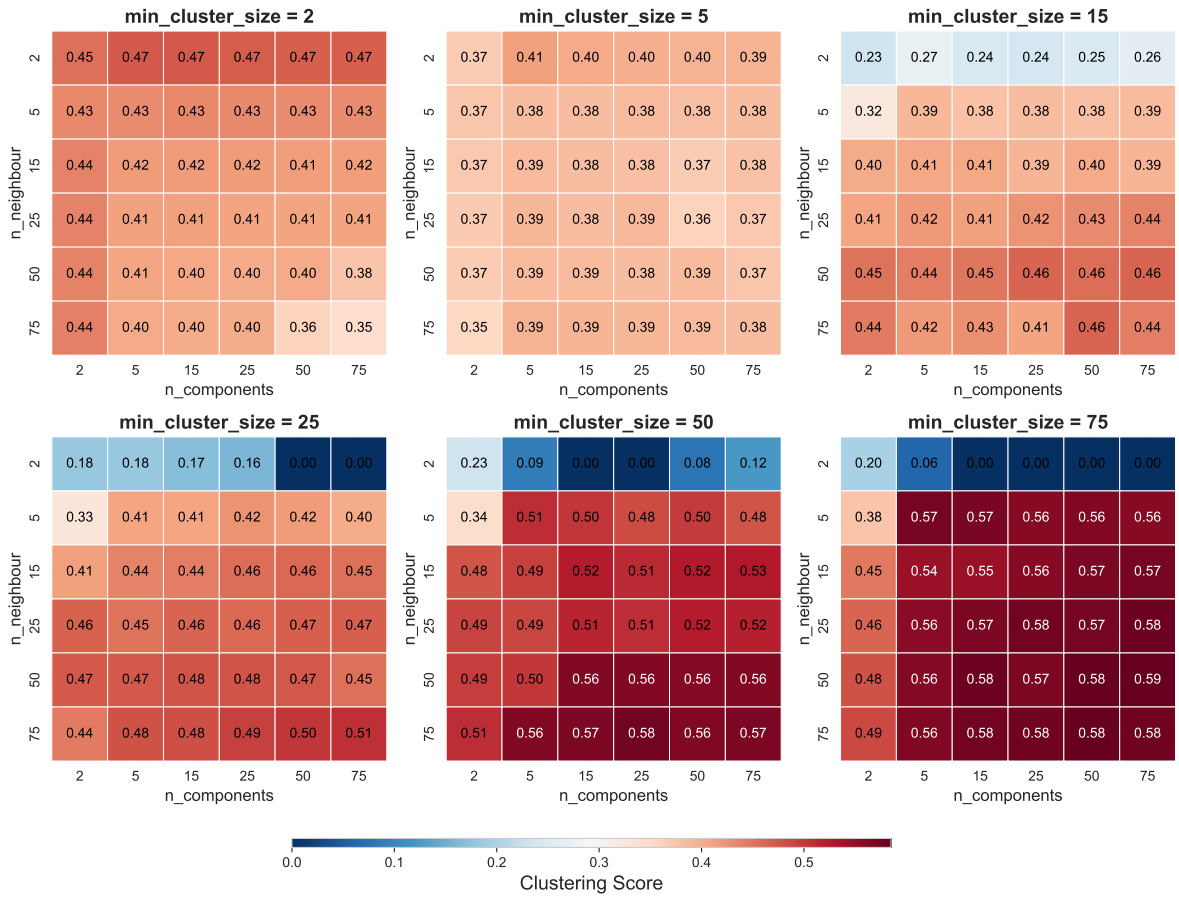


Figure 7.2: Heatmaps showing the Clustering Score across combinations of UMAP and HDBSCAN parameters on the *YahooAnswers* dataset. Each subplot corresponds to a different `min_cluster_size` value (2, 5, 15, 25, 50, 75, and 100). Color intensity indicates the Clustering Score, based on the average of min-max scaled metrics (Clustered Proportion, Purity, AMI, and ARI). These heatmaps reveal how topic modeling performance varies with different granularity and structural preservation settings.

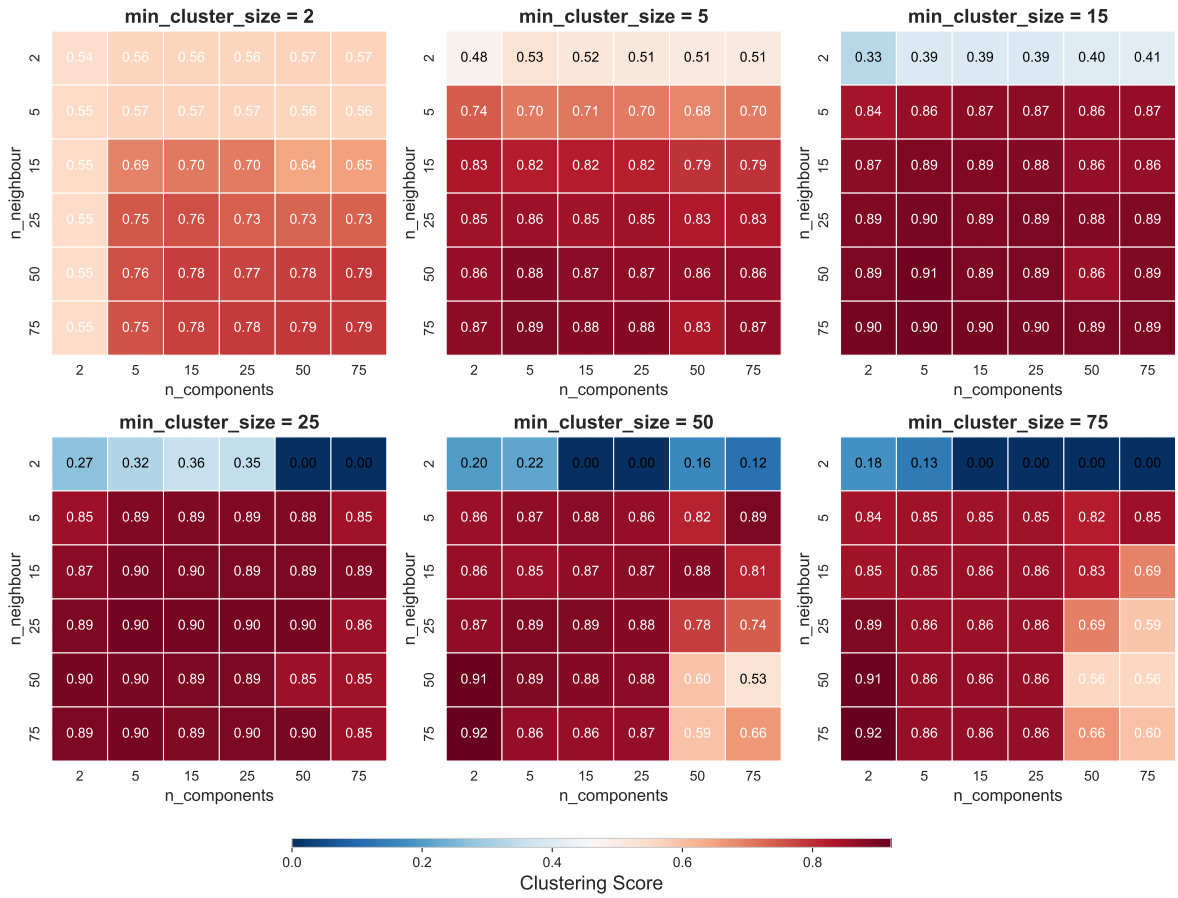


Figure 7.3: Heatmaps showing the Clustering Score across combinations of UMAP and HDBSCAN parameters on the *CCNEWS* dataset. Each subplot corresponds to a different `min_cluster_size` value (2, 5, 15, 25, 50, 75, and 100). Color intensity indicates the Clustering Score, based on the average of min-max scaled metrics (Clustered Proportion, Purity, AMI, and ARI). These heatmaps reveal how topic modeling performance varies with different granularity and structural preservation settings.

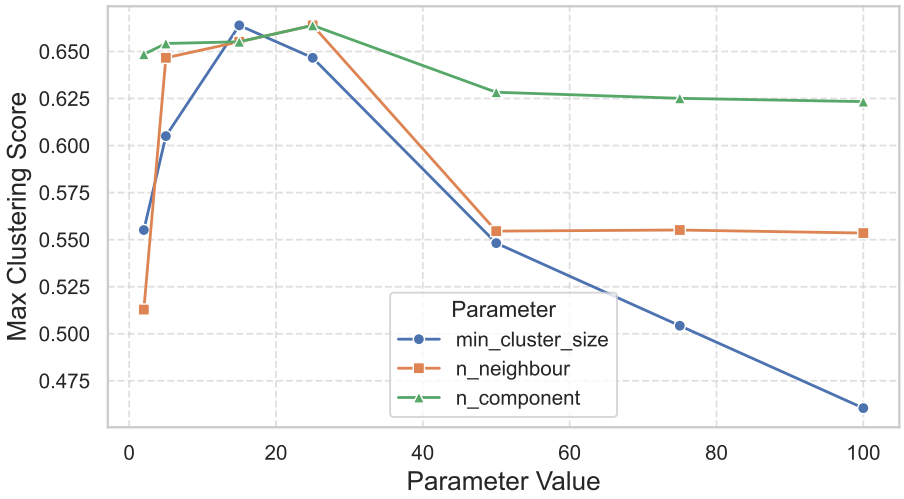


Figure 7.4: Maximum Clustering Score achieved for each individual hyperparameter: `n_neighbors`, `n_components`, and `min_cluster_size` on the *20Newsgroups* dataset. For each value along the x-axis, the other two parameters are varied across their full range, and the best (maximum) Clustering Score is recorded.

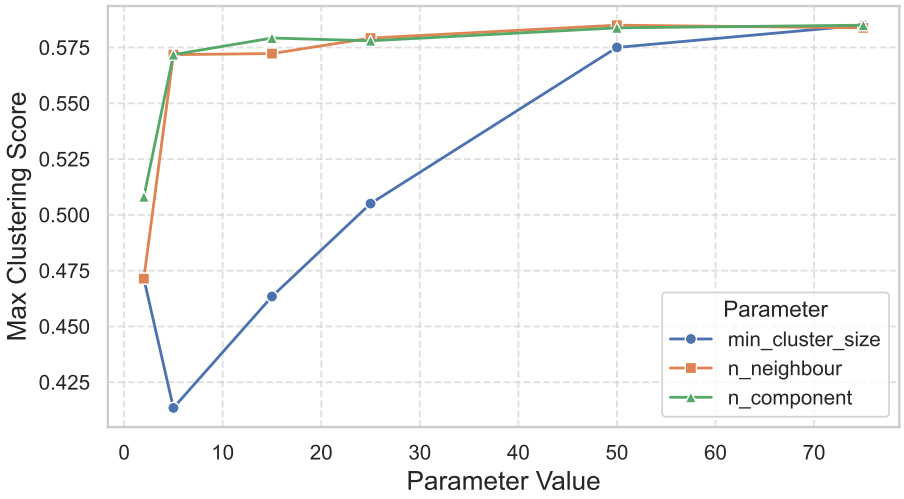


Figure 7.5: Maximum Clustering Score achieved for each individual hyperparameter:`n_neighbors`, `n_components`, and `min_cluster_size` on the *YahooAnswers* dataset. For each value along the x-axis, the other two parameters are varied across their full range, and the best (maximum) Clustering Score is recorded.

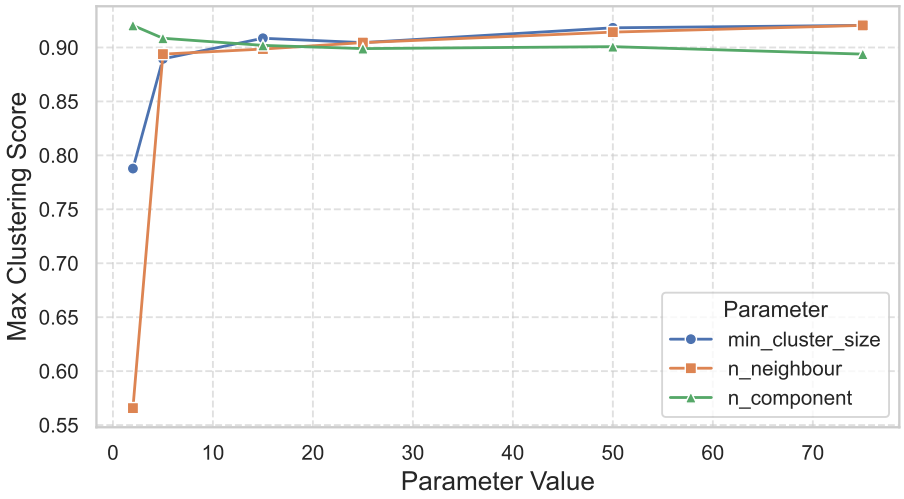


Figure 7.6: Maximum Clustering Score achieved for each individual hyperparameter: `n_neighbors`, `n_components`, and `min_cluster_size` on the *CCNEWS* dataset. For each value along the x-axis, the other two parameters are varied across their full range, and the best (maximum) Clustering Score is recorded.

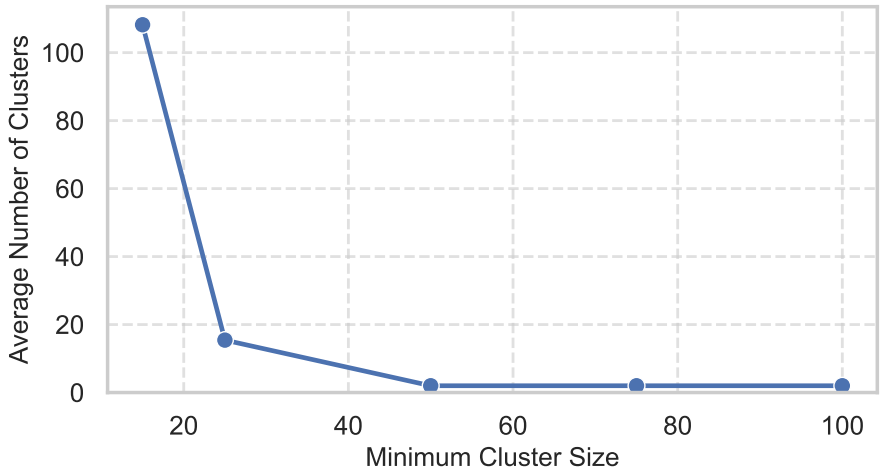


Figure 7.7: Average number of clusters produced by HDBSCAN on the *20Newsgroups* dataset as a function of `min_cluster_size`.

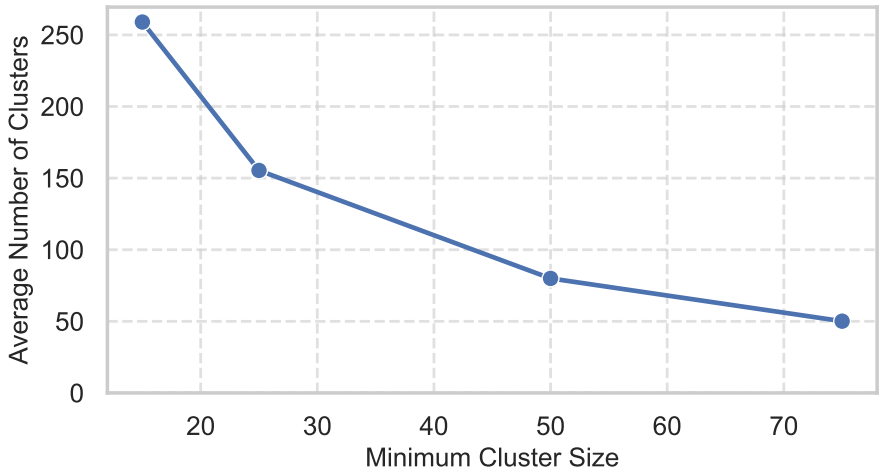


Figure 7.8: Average number of clusters produced by HDBSCAN on the *YahooAnswers* dataset as a function of `min_cluster_size`.

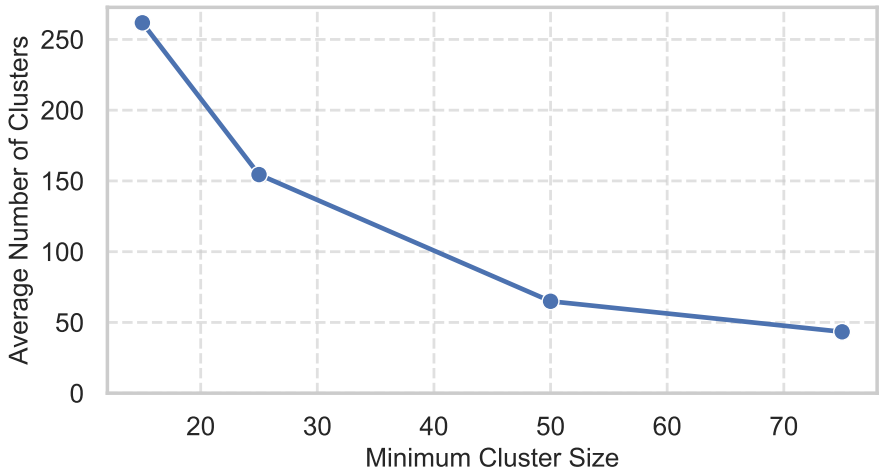


Figure 7.9: Average number of clusters produced by HDBSCAN on the *CCNEWS* dataset as a function of `min_cluster_size`.

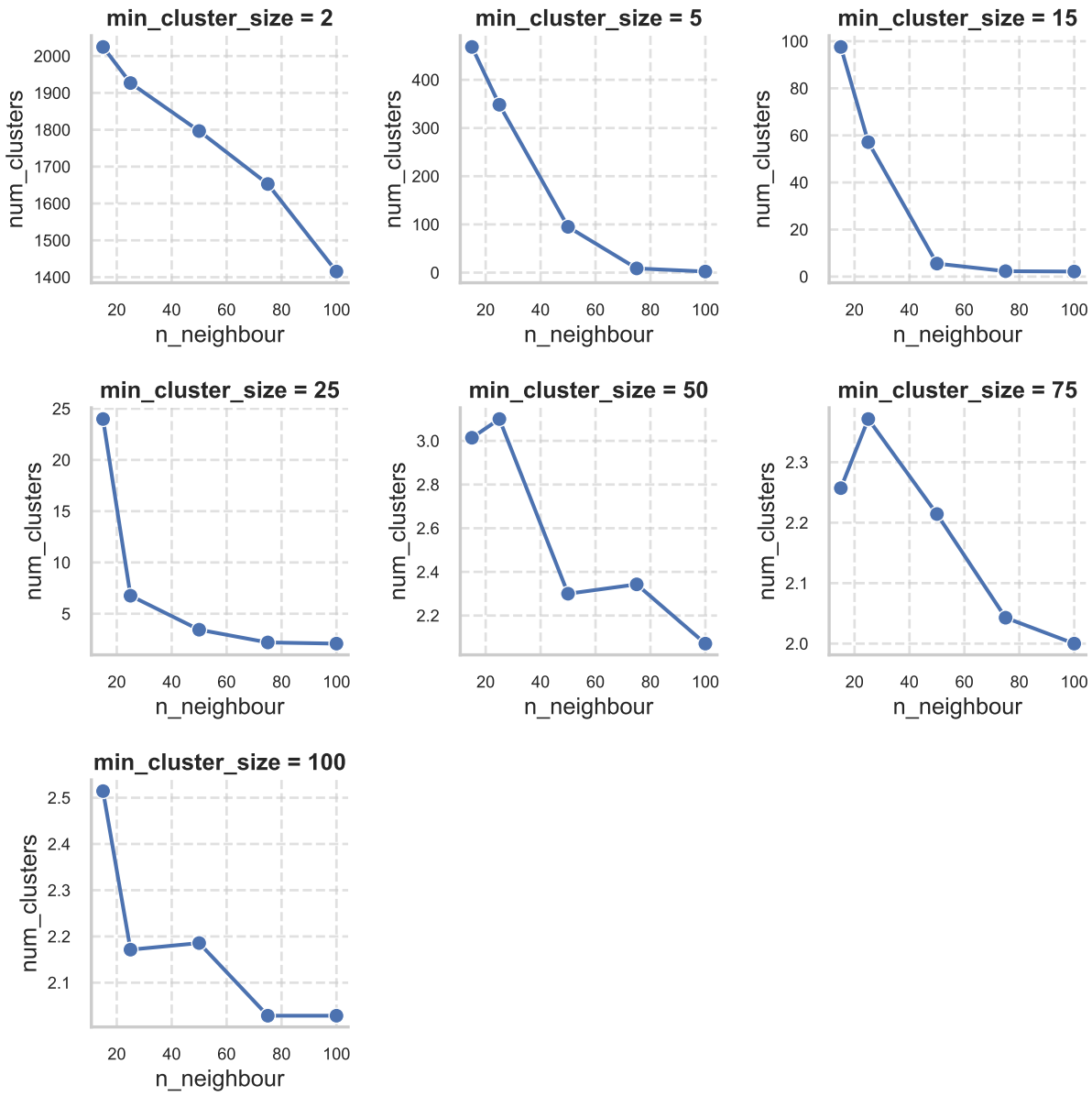


Figure 7.10: Average number of clusters (*num_clusters*) identified by HDBSCAN on the *20Newsgroups* dataset as a function of `n_neighbors`, plotted separately for each value of `min_cluster_size`.

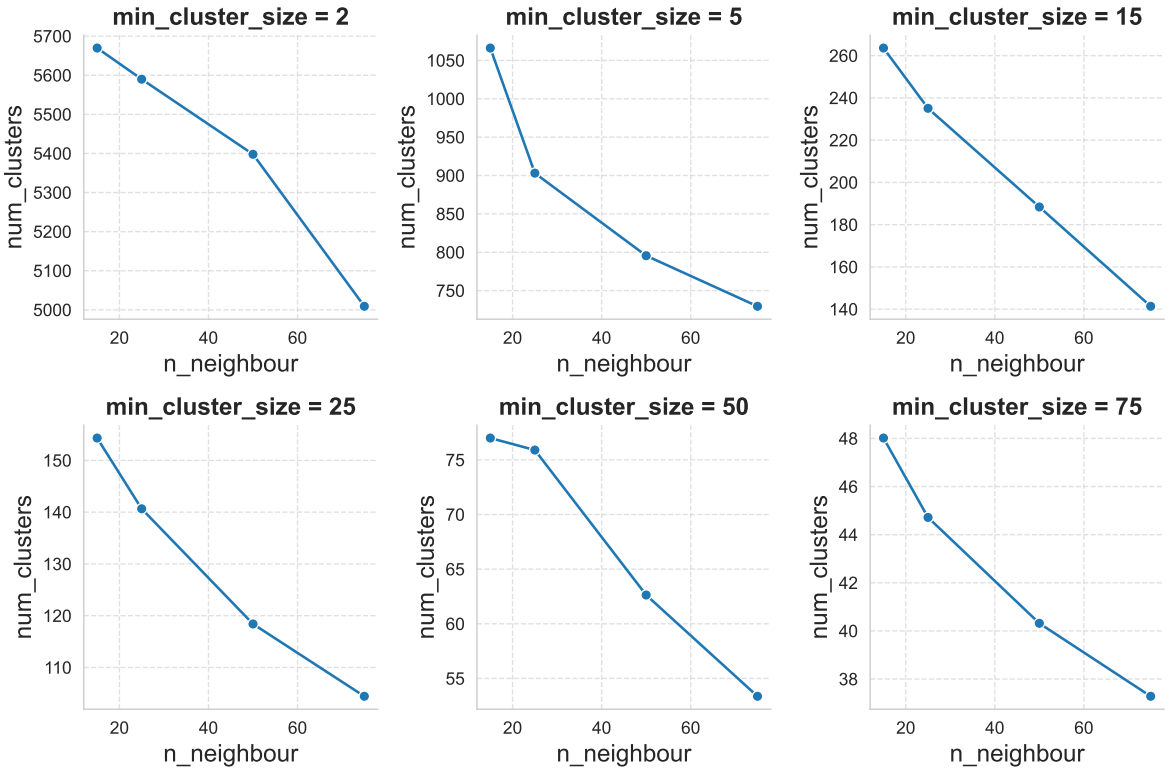


Figure 7.11: Average number of clusters (*num_clusters*) identified by HDBSCAN on the *YahooAnswers* dataset as a function of `n_neighbors`, plotted separately for each value of `min_cluster_size`.

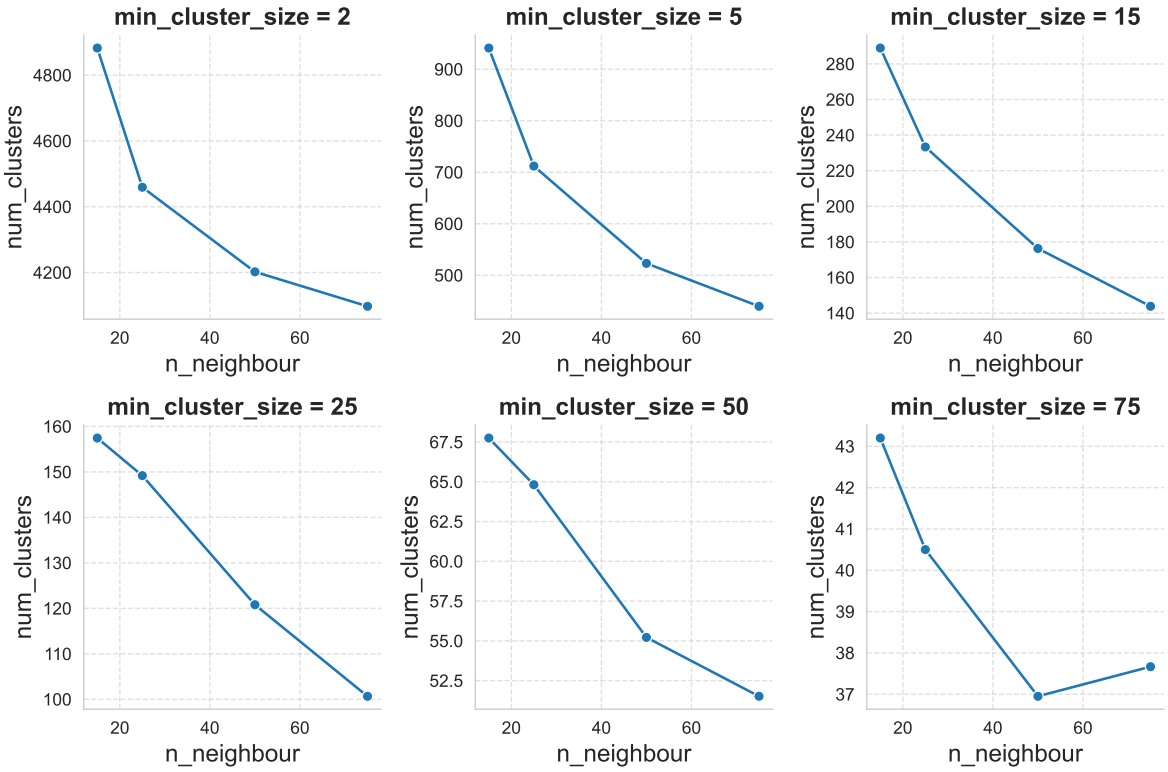


Figure 7.12: Average number of clusters (*num_clusters*) identified by HDBSCAN on the *CCNEWS* dataset as a function of `n_neighbors`, plotted separately for each value of `min_cluster_size`.

Conclusion. Our experiments demonstrate that the performance of Top2Vec is highly dependent on the selection of UMAP and HDBSCAN hyperparameters. While the optimal values can vary based on dataset characteristics, such as the number and granularity of topics, certain patterns are consistent. Across all datasets, a `n_neighbors` value of 15 serves as a reliable default, with higher values offering marginal improvements, particularly for datasets with fewer, broader topics. Similarly, moderate values of `n_components` (15–25) are effective for most datasets, with higher dimensionality offering benefits mainly for more diverse or coarse-grained topic structures. For `min_cluster_size`, values around 15 offer a good balance between capturing fine-grained clusters and avoiding over-fragmentation, though larger values may help when the number of ground truth topics is low. Overall, our results support using 15 as a baseline value for all three hyperparameters, with targeted tuning depending on the dataset's size and topic distribution to optimize clustering

effectiveness. Recommended ranges for each dataset are summarized in Table 7.3.

Dataset	n_neighbors	n_components	min_cluster_size
20Newsgroups	15–25	15–25	15
YahooAnswers	15–70	15–75	15–75
CCNEWS	15–75	15–75	15–75

Table 7.3: These ranges reflect empirical findings from our experiments and highlight how dataset characteristics such as topic granularity and size influence optimal settings. A baseline of 15 for all three parameters is generally robust, but tuning within these ranges may yield improved clustering performance tailored to the dataset.

7.4 Evaluating Hierarchical Clustering Approaches

We demonstrated in Figures 7.10, 7.11, 7.12 how `n_neighbors` and `min_cluster_size` can be used to find different resolution of clusters by varying the tradeoff between local and global structure in UMAP. Additionally we showed in Figures 7.1, 7.2 and 7.3 that high clustering scores can be achieved for those configurations of UMAP and HDBSCAN showing that they all capture coherent semantic structures just at different resolutions.

To evaluate how tuning UMAP to capture more global structure affects performance, we use labeled data from *20newsgroups*, *YahooAnswers*, and *CCNEWS*. These datasets contain coarse grained labels that are expected to be reflected in the global structure. We evaluate how well the resulting UMAP-reduced embeddings preserve label separability using several clustering validation metrics. Given that our dataset includes ground-truth labels, we treat these labels as cluster assignments to analyze the structure of the UMAP reduced space. Specifically, we compute the *Calinski-Harabasz score* [20], which measures the ratio of between class to within class dispersion, higher values indicate better-defined class clusters. The *Davies-Bouldin score* [30] captures intra-class similarity relative to inter-class separation, where lower scores reflect tighter and more distinct groupings. Lastly, the *Silhouette score* [96] evaluates the relative closeness of each sample to its own class compared to other classes, with higher values signaling improved separability. These metrics enable a comprehensive analysis of how well the reduced representation space captures the underlying class structure.

Figures 7.13–7.15 demonstrate the impact of increasing the `n_neighbors` parameter in UMAP on clustering quality for the *20Newsgroups*, *YahooAnswers*, and *CCNEWS* datasets.

We assess the structure of the reduced embedding space using the Calinski-Harabasz, Davies-Bouldin, and Silhouette scores. Across all datasets, increasing `n_neighbors`, which preserves more global structure in the reduced embedding space, generally leads to improvements in clustering quality. For each dataset, higher values of `n_neighbors` correspond to increasing Calinski-Harabasz and Silhouette scores, as well as decreasing Davies-Bouldin scores, indicating more distinct and coherent clusters. These consistent trends across datasets support our hypothesis that tuning UMAP to better preserve global structure helps capture higher-level semantic groupings aligned with dataset labels, making it a useful strategy for hierarchical topic modeling at coarser resolutions.

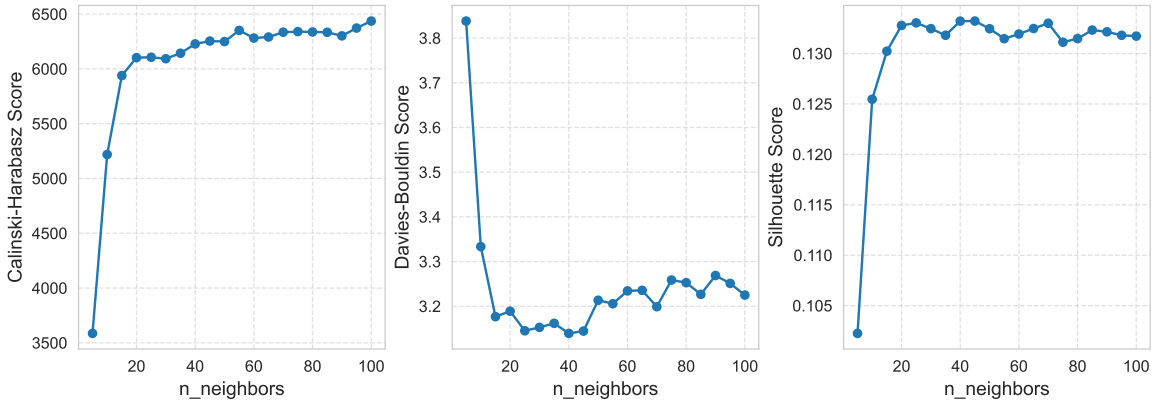


Figure 7.13: Local vs. Global UMAP structure evaluation on the *20newsgroups* dataset.

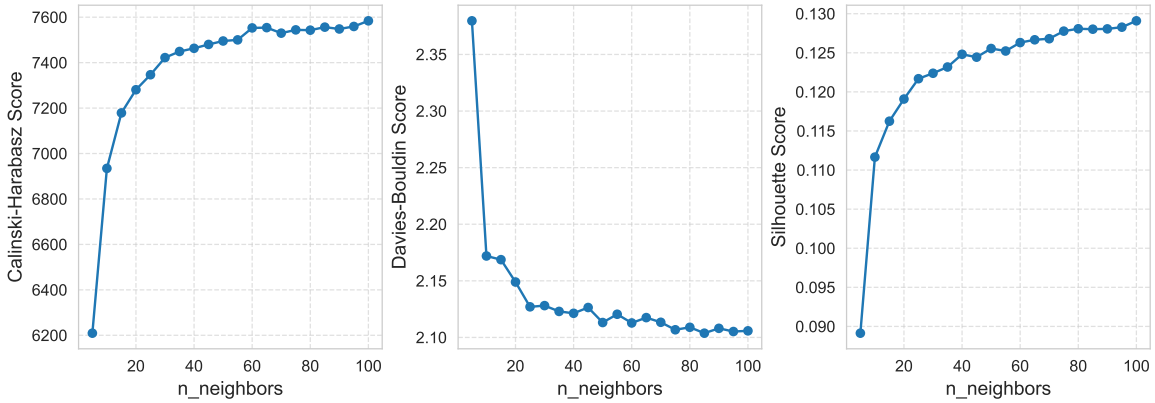


Figure 7.14: Local vs. Global UMAP structure evaluation on the *YahooAnswers* dataset.

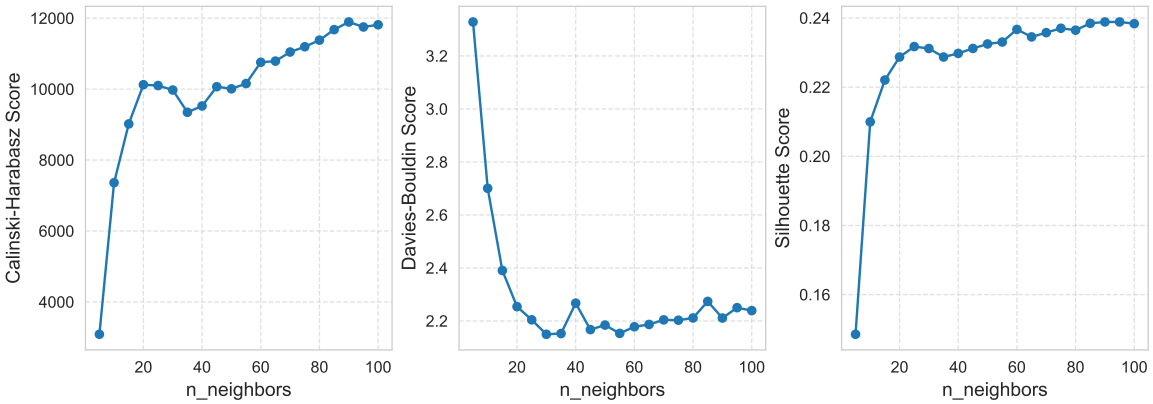


Figure 7.15: Local vs. Global UMAP structure evaluation on the *CCNEWS* dataset.

Across all datasets, increasing `n_neighbors`, which preserves more global structure, generally improves clustering structure. These consistent trends support our hypothesis that tuning UMAP to preserve global structure helps capture higher-level semantic groupings aligned with dataset labels. This makes it a useful strategy for hierarchical topic modeling at coarser resolutions.

As discussed in 4.7, HDBSCAN with a higher `min_cluster_size` will not aggregate coarse-grained clusters found by smaller `min_cluster_size` it instead finds the most persistent and dense regions, therefore it will not find a coarse to fine-grain topic hierarchy. Thus, higher values of `min_cluster_size` uncover stable dense regions rather than aggregates of previously discovered granular clusters, capturing persistence rather than semantic coverage.

In order to demonstrate that, and evaluate the hiearchy generated by HDBSCAN for the purposes of topic modeling we compare the clusters found at different levels of `min_cluster_size` and and assess the coverage of fine-grained clusters by those discovered at higher values. Specifically, we measure how many of the fine-grained clusters make it into the high level clusters and evaluate their diversity. We also measure the clustering score at each cluster resolution.

We additionally perform the same analysis using our proposed two stage approach. The approach first uses UMAP followed by HDSCAN to find the granular clusters in the dataset and computes their topic vectors. Then we UMAP-reduce the topic vectors and perform agglomerative clustering using the low-dimensional topic vectors. Unlike HDBSCAN, this method produces a complete hierarchical tree without pruning, ensuring

that all fine-grained topics are retained in coarser representations, therefore allowing for multi-resolution analysis of the topic space.

To perform this analysis we use the *20newsgroups*, *YahoAnswers*, and *CCNEWS* datasets. We create a hierarchy of three levels using HDBSCAN and our proposed approach. For each level we calculate how much of the first level clusters are in the higher level clusters. We also calculate the diversity of each level as well as the clustering score. The motivation behind computing the semantic diversity of each level is to to further compare the impact of the incompleteness of the HDBSCAN hierarchy.

To quantify the semantic diversity within each topic, we use *Mean Pairwise Distance* computed over the set of document vectors assigned to a topic. This metric measures the average cosine distance between all pairs of document vectors within a topic cluster, providing a direct and interpretable indication of semantic spread. A higher mean distance reflects greater internal variability and thus higher diversity within the topic. We can use this measure evaluate diversity within the topic hierarchy, enabling comparison across clustering levels.

In these experiments, we assess clustering quality using the following metrics:

- **Level-0 Covered:** Measures the proportion of fine-grained (Level-0) clusters whose documents are represented in higher-level clusters.
- **Mean Pairwise Distance:** The mean pairwise cosine distance between cluster document vectors. Higher values indicate more distinct clusters, and thus, greater semantic separation among topics.
- **Clustering Score:** A composite metric that accounts for proportion of clustered points, purity, AMI and ARI.

HDBSCAN Hierarchy				
Level	#Clusters	Level-0 Covered	Avg Pairwise Dist	Clustering Score
0	299.6 ± 7.3	1.000 ± 0.000	0.575 ± 0.003	0.489 ± 0.003
1	77.2 ± 5.1	0.660 ± 0.017	0.664 ± 0.006	0.544 ± 0.014
2	22.0 ± 1.7	0.189 ± 0.025	0.767 ± 0.010	0.653 ± 0.012
Agglomerative Hierarchy				
Level	#Clusters	Level-0 Covered	Avg Pairwise Dist	Clustering Score
0	282.5 ± 11.87	1.000 ± 0.000	0.615 ± 0.004	0.436 ± 0.013
1	80.0 ± 0.0	1.000 ± 0.000	0.727 ± 0.004	0.604 ± 0.011
2	20.0 ± 0.0	1.000 ± 0.000	0.814 ± 0.006	0.606 ± 0.015

Table 7.4: Comparison of hierarchical clustering on the *20newsgroups* dataset. While HDBSCAN improves diversity at higher levels, it loses fine-grained topics. The agglomerative method retains full coverage with competitive performance.

HDBSCAN Hierarchy				
Level	#Clusters	Level-0 Covered	Mean Pairwise Dist	Clustering Score
0	300.9 ± 3.0	1.000 ± 0.000	0.647 ± 0.003	0.368 ± 0.002
1	78.8 ± 3.6	0.685 ± 0.020	0.754 ± 0.003	0.423 ± 0.006
2	21.0 ± 1.2	0.089 ± 0.020	0.856 ± 0.004	0.514 ± 0.010
Agglomerative Hierarchy				
Level	#Clusters	Level-0 Covered	Mean Pairwise Dist	Clustering Score
0	328.2 ± 10.8	1.000 ± 0.000	0.671 ± 0.004	0.297 ± 0.008
1	80.0 ± 0.0	1.000 ± 0.000	0.795 ± 0.004	0.426 ± 0.008
2	20.0 ± 0.0	1.000 ± 0.000	0.889 ± 0.006	0.473 ± 0.008

Table 7.5: Comparison of hierarchical clustering on the *YahooAnswers* dataset. While HDBSCAN improves diversity at higher levels, it loses fine-grained topics. The agglomerative method retains full coverage with competitive performance.

HDBSCAN Hierarchy				
Level	#Clusters	Level-0 Covered	Mean Pairwise Dist	Clustering Score
0	300.0 ± 1.6	1.000 ± 0.000	0.463 ± 0.002	0.578 ± 0.005
1	77.8 ± 2.2	0.767 ± 0.014	0.560 ± 0.004	0.715 ± 0.004
2	29.0 ± 1.2	0.122 ± 0.012	0.645 ± 0.007	0.725 ± 0.005
Agglomerative Hierarchy				
Level	#Clusters	Level-0 Covered	Mean Pairwise Dist	Clustering Score
0	305.2 ± 17.9	1.000 ± 0.000	0.491 ± 0.003	0.537 ± 0.015
1	80.0 ± 0.0	1.000 ± 0.000	0.617 ± 0.005	0.672 ± 0.034
2	20.0 ± 0.0	1.000 ± 0.000	0.712 ± 0.008	0.584 ± 0.022

Table 7.6: Comparison of hierarchical clustering on the *CCNEWS* dataset. While HDBSCAN improves diversity at higher levels, it loses fine-grained topics. The agglomerative method retains full coverage with competitive performance.

Summary of Results. The comparative analysis across the *20Newsgroups*, *YahooAnswers*, and *CCNEWS* datasets demonstrates differences between the HDBSCAN-based and agglomerative hierarchies. Detailed metrics for each dataset and method are presented in Tables 7.4, 7.6, and 7.6. The HDBSCAN hierarchies exhibit a sharp decline in *Level-0 Coverage* as we move to coarser clustering levels. This indicates that many fine-grained clusters are not preserved in higher levels, highlighting the incomplete nature of HDBSCAN's pruning-based hierarchy construction. In contrast, the agglomerative approach—built on top of granular clusters ensures a full representation of the fine-grained topic space.

This limitation in coverage for HDBSCAN is further reflected in the *semantic diversity* scores. Although diversity increases with coarser levels in both methods as expected, HDBSCAN consistently shows lower diversity compared to the agglomerative hierarchy at corresponding levels. This shows that HDBSCAN clusters are more compact but less representative of the broader semantic space, owing to its focus on persistent dense regions rather than semantic coverage.

Finally, while HDBSCAN achieves marginally higher **clustering scores**, particularly in the coarsest levels, this comes at the expense of representational completeness. The agglomerative method, though sometimes slightly lower in clustering score, offers a more comprehensive multi-resolution topic hierarchy. This trade-off demonstrates the key benefit

of our proposed approach: a complete semantic hierarchy suited for topic exploration.

7.5 Topic Modeling Results

7.5.1 Evaluated Models

We evaluate LDA, ETM, CTM, `Top2Vec`, BERTopic and our proposed model. To level the playing field between the neural methods we use the same SBERT model, `all-mpnet-base-v2` , which is based on MPNet [99] for all of them, except ETM which uses `word2vec`. We use Comparing and Optimizing Topic Models is Simple (OCTIS) to evaluate models with C_{NPMI}, C_V, and C_{WE}. Our goal is to compare our approach to topic modeling baselines like LDA while also comparing to current state of the art approaches.

7.5.2 Model Training

OCTIS is a framework designed to standardize the evaluation of topic modeling approaches and enable reproducibility of results. It provides an integrated environment where users can select topic modeling algorithms, including traditional models like LDA and neural-based models. OCTIS includes a variety of evaluation metrics, covering coherence, diversity, and accuracy. It also supports the integration of custom models, metrics, and datasets, making it a good tool for evaluating topic models. We trained LDA and ETM using OCTIS for standardization and reproducibility.

We trained CTM, BERTopic and `Top2Vec` using their respective github implementations[1][2][3] as they are not available in the OCTIS framework.

Due to the computational cost and time required for training multiple models across a range of topics and evaluating with BERTScore and other evaluation metrics, we opted to train each model once for each configuration.

7.5.3 Model Parameters

To train LDA and ETM, we used OCTIS [104] and their suggested model parameters. For CTM, we use the suggested parameters from the paper [14] and their GitHub[1]. For

[1]https://github.com/MilaNLProc/contextualized-topic-models
[2]https://github.com/MaartenGr/BERTopic
[3]https://github.com/ddangelov/Top2Vec

BERTopic [42], due to the little information on parameters in the paper, we use the default values from their GitHub[2]. For `Top2Vec` [6], we use the suggested parameters from the paper and their GitHub[3]. All relevant evaluated model parameters are shown in Table 7.7.

Model Parameters

Parameter	Value
LDA	
decay	0.5
gamma_threshold	0.001
iterations	50
ETM	
num_epochs	100
t_hidden_size	800
t_hidden_size	800
rho_size	300
embedding_size	300
activation	relu
dropout	0.5
lr	0.005
optimizer	adam
batch_size	128
wdecay	1
CTM	
embedding_model	`all-mpnet-base-v2`
hidden_sizes	(100, 100)
dropout	0.2
learn_priors	True
batch_size	64
lr	2×10^{-3}
momentum	0.99
solver	adam
num_epochs	100
BERTopic	
embedding_model	`all-mpnet-base-v2`
top_n_words	10
umap_n_neighbours	15
umap_n_components	5
hdbscan_min_cluster_size	10
Top2Vec	
embedding_model	`all-mpnet-base-v2`
umap_n_neighbours	15
umap_n_components	5
hdbscan_min_cluster_size	15
min_count	50
C-Top2Vec	
embedding_model	`all-mpnet-base-v2`
umap_n_neighbours	50
umap_n_components	5
hdbscan_min_cluster_size	15
min_count	50
window_size	50
stride	40
smoothing_window_size	3

Table 7.7: Model parameters for all the evaluated models.

We train LDA, ETM, CTM, `Top2Vec`, BERTopic and `C-Top2Vec` on the *20newsgroups* dataset for 20, 60, 100 and 140 topics, and on the *Yahoo! Answers* dataset for 10, 60, 100 and 140 topics. We average the results over four runs for each dataset and report the mean scores. We start with 20 topics for the `20newwsgroups` dataset because it originally consists of 20 topic labels. We start with 10 topics for the `YahooAnswers` dataset due to it having 10 topics in its labels. Our goal is to demonstrate the performance of the models across a range of topic sizes, to assess their performance for different topic granularities.

7.5.4 Topic Coherence Evaluation

To evaluate topic coherence, we use the top 10 words of each topic for LDA, ETM, CTM, `Top2Vec`, and BERTopic. For our approach, we used the top 10 phrases of each topic. To evaluate the phrases with C_{NPMI}, C_{V}, and C_{WE}, we split up the ordered phrases into words and take the top 10, as these methods require single-word descriptions. For C_{SBERT} we use the entire phrases for our approach.

Table 7.8 shows that `C-Top2Vec` outperforms all other models on C_{NPMI} and C_{SBERT} while remaining competitive in the other metrics, demonstrating that `C-Top2Vec` produces more coherent topics. `Top2Vec` has better performance than our approach on C_{WE} results, this is likely because we split up the phrases into single words and take only the top 10 words. This reduces the coherence as can be demonstrated by the C_{SBERT} score which allows for measuring the coherence using the entire phrases.

Model	C_{NPMI}	C_V	C_{WE}	C_{SBERT}
20newsgroups				
LDA	-0.068	0.460	0.026	0.244
ETM	-0.004	0.508	0.042	0.244
CTM	0.032	0.615	0.046	0.269
BERTopic	0.027	0.540	0.043	0.260
Top2Vec	-0.070	0.561	**0.126**	0.480
C-Top2Vec	**0.138**	**0.684**	0.116	**0.514**
YahooAnswers				
LDA	-0.086	0.374	0.023	0.235
ETM	-0.010	0.426	0.022	0.249
CTM	0.050	**0.601**	0.038	0.294
BERTopic	0.028	0.486	0.052	0.290
Top2Vec	-0.068	0.494	**0.106**	0.473
C-Top2Vec	**0.094**	0.532	0.090	**0.506**

Table 7.8: Topic coherence comparison across models.

Table 7.10 and 7.9 show a more detailed analysis with number of topics breakdown.

Topics	20	60	100	140
LDA				
C_{NPMI}	-0.024	-0.063	-0.097	-0.086
C_V	0.496	0.478	0.436	0.432
C_{WE}	0.027	0.033	0.020	0.026
C_{SBERT}	0.230	0.222	0.220	0.222
ETM				
C_{NPMI}	-0.005	-0.003	-0.005	-0.001
C_V	0.511	0.504	0.508	0.508
C_{WE}	0.045	0.045	0.045	0.035
C_{SBERT}	0.241	0.244	0.245	0.245
CTM				
C_{NPMI}	0.06	0.03	0.026	0.014
C_V	0.645	0.617	0.607	0.592
C_{WE}	0.051	0.047	0.046	0.041
C_{SBERT}	0.278	0.267	0.268	0.264
BERTopic				
C_{NPMI}	0.001	0.023	0.04	0.044
C_V	0.478	0.517	0.572	0.593
C_{WE}	0.016	0.046	0.055	0.056
C_{SBERT}	0.233	0.253	0.271	0.283
Top2Vec				
C_{NPMI}	-0.041	-0.072	-0.075	-0.093
C_V	0.615	0.568	0.544	0.516
C_{WE}	**0.155**	**0.12**	**0.116**	0.111
C_{SBERT}	0.511	0.484	0.469	0.458
C-Top2Vec				
C_{NPMI}	**0.153**	**0.145**	**0.122**	**0.13**
C_V	**0.721**	**0.683**	**0.665**	**0.666**
C_{WE}	0.129	0.113	0.109	**0.112**
C_{SBERT}	**0.542**	**0.515**	**0.502**	**0.497**

Table 7.9: Topic coherence scores across models on textit20newsgroups

Topics	10	60	100	140
LDA				
C_{NPMI}	-0.084	-0.095	-0.078	-0.086
C_V	0.399	0.393	0.360	0.345
C_{WE}	0.019	0.029	0.025	0.020
C_{SBERT}	0.244	0.243	0.228	0.224
ETM				
C_{NPMI}	-0.020	-0.008	-0.004	-0.007
C_V	0.407	0.430	0.436	0.431
C_{WE}	0.030	0.012	0.017	0.027
C_{SBERT}	0.244	0.250	0.251	0.250
CTM				
C_{NPMI}	0.044	0.047	0.057	0.053
C_V	0.610	**0.615**	**0.602**	**0.576**
C_{WE}	0.006	0.062	0.046	0.037
C_{SBERT}	0.294	0.308	0.292	0.282
BERTopic				
C_{NPMI}	-0.015	0.030	0.049	0.046
C_V	0.421	0.467	0.518	0.539
C_{WE}	0.063	0.037	0.048	0.061
C_{SBERT}	0.261	0.265	0.303	0.331
Top2Vec				
C_{NPMI}	-0.112	-0.059	-0.046	-0.054
C_V	0.428	0.517	0.520	0.509
C_{WE}	0.081	0.107	**0.119**	**0.116**
C_{SBERT}	0.438	0.483	0.486	0.483
C-Top2Vec				
C_{NPMI}	**0.053**	**0.110**	**0.109**	**0.105**
C_V	**0.493**	0.554	0.543	0.536
C_{WE}	**0.045**	**0.112**	0.101	0.101
C_{SBERT}	**0.463**	**0.514**	**0.527**	**0.518**

Table 7.10: Topic coherence scores across models on *Yahoo! Answers*.

7.5.5 Topic Assignment Evaluation

To evaluate how well the models group documents into topics, we compare their topic assignment with the original topic labels of the datasets. For LDA, ETM, and CTM we use the document topic distribution and assign each document to the topic with the highest probability. `Top2Vec` and BERTopic only assign documents to a single topic, so we just use those assignments. For our approach, we combine the document topic probability and the document topic relevance by taking their element-wise multiplication and then assigning each document to the topic with the highest resulting score.

The results in Table 7.11 show the AMI and ARI results averaged over 5 runs comparing the true topic labels to assigned topic labels for each dataset. Our results are tied with `Top2Vec` with a very slight edge, over all other models.

Model	**AMI**	**ARI**
	20newsgroups	
LDA	0.228±0.022	0.084±0.019
ETM	0.431±0.017	0.258±0.012
CTM	0.275±0.012	0.102±0.003
BERTopic	0.430±0.025	0.095±0.016
Top2vec	**0.532±0.023**	**0.330±0.061**
C-Top2Vec	**0.529±0.029**	0.257±0.116
	YahooAnswers	
LDA	0.121±0.029	0.054±0.018
ETM	0.228±0.014	0.170±0.020
CTM	0.147±0.009	0.050±0.028
BERTopic	0.238±0.012	0.026±0.018
Top2Vec	**0.331±0.010**	**0.138±0.085**
C-Top2Vec	**0.348±0.029**	**0.150±0.114**

Table 7.11: Evaluation of topic assignment accuracy for various models on the *20newsgroups* and *YahooAnswers* datasets using Adjusted Mutual Information (AMI) and Adjusted Rand Index (ARI). Each model assigns documents to topics based on its internal strategy. Results show that `C-Top2Vec` and `Top2Vec` outperform other models, with `C-Top2Vec` achieving the highest AMI and remaining competitive on ARI across both datasets, demonstrating strong alignment between discovered and ground truth topic labels.

We inspected the distributions of true topic labels within the topic clusters and the cluster sizes for `Top2Vec` and our approach. We made two observations that explain the disparity in ARI. Our approach has more even-sized clusters and the cluster purity of true topic labels decreases as the number of topics are increased. The `Top2Vec` clusters are more unevenly distributed with large clusters dominating even as the number of topics are increased. These large clusters have purer true topic labels. Our observations show that our approach produces more specific and evenly sized topics as the number of topics increase whereas `Top2Vec` does not break up the dominant topics, resulting in the uneven size of topic clusters.

Tables 7.12 and 7.13 show a more detailed analysis with number of topics breakdown.

Topics	20	60	100	140
LDA				
ARI	0.095	0.100	0.088	0.051
AMI	0.235	0.258	0.224	0.196
ETM				
ARI	0.250	0.279	0.249	0.252
AMI	0.417	0.454	0.413	0.440
CTM				
ARI	0.167	0.114	0.092	0.069
AMI	0.278	0.274	0.277	0.270
BERTopic				
ARI	0.111	0.108	0.092	0.070
AMI	0.388	0.452	0.444	0.434
Top2Vec				
ARI	0.427	**0.336**	**0.291**	**0.268**
AMI	0.566	**0.537**	**0.517**	**0.507**
C-Top2Vec				
ARI	**0.445**	0.260	0.179	0.145
AMI	**0.575**	0.531	0.509	0.500

Table 7.12: Topic assignment accuracy across models on *20newsgroups*.

Topics	10	60	100	140
LDA				
ARI	0.085	0.051	0.045	0.037
AMI	0.171	0.109	0.105	0.100
ETM				
ARI	0.193	0.178	0.168	0.139
AMI	0.247	0.235	0.218	0.213
CTM				
ARI	0.095	0.049	0.031	0.023
AMI	0.148	0.161	0.144	0.136
BERTopic				
ARI	0.056	0.021	0.014	0.011
AMI	0.218	0.244	0.245	0.245
Top2Vec				
ARI	0.281	**0.125**	**0.084**	**0.064**
AMI	0.341	0.34	0.326	0.317
C-Top2Vec				
ARI	**0.343**	0.118	0.079	0.059
AMI	**0.396**	**0.344**	**0.329**	**0.322**

Table 7.13: Topic assignment accuracy across models on *Yahoo! Answers*.

7.5.6 Topic BERTScore Evaluation

To evaluate the model's topic coherence and how well the topics represent their underlying documents we use the top 10 topic words for LDA, ETM, CTM, BERTopic, and `Top2Vec` as the references. For `C-Top2Vec`, we use the top 10 phrases as the references. For the candidates, we use the top 50 documents of each topic. For LDA, ETM, CTM we select the top 50 candidates by taking the 50 documents with the highest probability of each topic. For BERTopic and `Top2Vec`, we select the 50 documents with the highest similarity score to each topic.

In order to select the top 50 candidates for our model, we find the top 50 most relevant documents to each topic by using the element-wise multiplication of the document topic probability and the document topic relevance. Furthermore, to evaluate our topic segmentation, we use only the segments from the document that are assigned to the topic, rather than using the entire document.

We use a combination of topic distribution and topic relevance to rank documents. The topic distribution measures how much of a document is about a given topic, ensuring that documents with large sections of topically relevant segments are ranked highly. However, a high topic distribution does not mean that a document is most representative of the topic, it only indicates that a large portion of the document's segments are about that topic. Topic relevance measures how semantically similar a segment is to the topic and ensures segments representative of the topic are ranked highly. Without considering topic distribution, relevance alone could favor small but highly similar segments that may not provide comprehensive coverage of the topic.

By multiplying these two measures, we effectively balance both coverage and semantic fidelity. Documents that contain a substantial portion of the topic and have highly relevant content will rank the highest, ensuring that our selection captures the most representative and informative documents for each topic. This combined approach leads to more accurate and interpretable topic-document associations compared to methods that rely solely on probability or similarity.

Traditional topic models such as LDA, ETM, and CTM give a probability distribution of topics in a document, however these methods do not explicitly identify which part of a document is most relevant to the topic. Models like BERTopic and `Top2Vec` assign documents to a single topic. By evaluating BERTScore on the specific segments assigned to the topic, our approach demonstrates a granular understanding of topic distribution within documents. This allows us to measure the coherence of topics at the segment level, rather than treating entire documents as single-topic entities. As a result, we show that we

can identify the most topically relevant segments of a document that belong to each topic. Since each approach assigns topic-document relevance differently, the top 50 documents will vary across models. This is intentional, as it allows us to assess how well each algorithm identifies and associates its topics with the most relevant documents.

To compute the contextualized token embeddings of topics and documents, as suggested by the official BERTScore GitHub[4], we use the `microsoft/deberta-xlarge-mnli` model [46] which has the best correlation with human judgments.

To compute BERTScore recall, BERTScore_R, we create contextualized token embeddings for each topics top 10 words and its top 50 documents. We then compute pairwise cosine similarity between the embeddings of the topic's top 10 words and the embeddings of the topic documents. We calculate BERTScore_R for each topic by taking the maximum of similarity scores for each of the topic's contextualized tokens and doing a weighted sum using the IDF values of each word as shown Figure 6.1. We then average all the scores from each topic to have a single BERTScore_R.

The results in Table 7.14 show that our approach outperforms all other models. The BERTScore_R demonstrates that our topics are significantly more coherent in relation to their underlying documents. The BERTScore_P shows that the documents are well represented by the topics.

Table 7.15 and 7.16 show a more detailed analysis with number of topics breakdown.

[4]https://github.com/Tiiiger/bert_score

Model	$\textbf{BERTScore}_R$	$\textbf{BERTScore}_P$	$\textbf{BERTScore}_{F1}$
20newsgroups			
LDA	0.384	0.316	0.344
ETM	0.378	0.250	0.300
CTM	0.398	0.294	0.336
BERTopic	0.356	0.290	0.318
Top2Vec	0.410	0.309	0.351
C-Top2Vec	**0.456**	**0.374**	**0.409**
YahooAnswers			
LDA	0.365	0.347	0.352
ETM	0.392	0.265	0.315
CTM	0.414	0.322	0.359
BERTopic	0.351	0.308	0.326
Top2Vec	0.387	0.344	0.362
C-Top2Vec	**0.439**	**0.399**	**0.416**

Table 7.14: Evaluation of topic-document coherence using BERTScore on the *20newsgroups* and *YahooAnswers* datasets. Each model is assessed based on how well its top topic descriptors (words or phrases) align with the top 50 most representative documents per topic. BERTScore_R reflects how well documents are covered by the topics, BERTScore_P reflects how well the topics are represented by the documents, and BERTScore_{F1} provides an overall harmonic mean. `C-Top2Vec` significantly outperforms all baselines across all metrics, indicating superior topic-document alignment and more coherent, interpretable topic structures.

Topics	20	60	100	140
LDA				
BERTScore_R	0.386	0.401	0.380	0.370
BERTScore_P	0.296	0.332	0.319	0.315
BERTScore_{F1}	0.333	0.361	0.345	0.337
ETM				
BERTScore_R	0.375	0.390	0.370	0.375
BERTScore_P	0.241	0.254	0.258	0.248
BERTScore_{F1}	0.294	0.306	0.306	0.298
CTM				
BERTScore_R	0.405	0.402	0.392	0.392
BERTScore_P	0.295	0.295	0.292	0.295
BERTScore_{F1}	0.339	0.339	0.333	0.335
BERTopic				
BERTScore_R	0.357	0.361	0.355	0.353
BERTScore_P	0.275	0.292	0.296	0.295
BERTScore_{F1}	0.309	0.321	0.322	0.320
Top2Vec				
BERTScore_R	0.415	0.413	0.4080	0.406
BERTScore_P	0.307	0.309	0.310	0.309
BERTScore_{F1}	0.352	0.352	0.351	0.349
C-Top2Vec				
BERTScore_R	**0.463**	**0.456**	**0.454**	**0.449**
BERTScore_P	**0.368**	**0.372**	**0.376**	**0.378**
BERTScore_{F1}	**0.409**	**0.408**	**0.409**	**0.409**

Table 7.15: Topic BERTScore evaluation on the *20newsgroups* dataset.

Metric	10	60	100	140
LDA				
BERTScore_R	0.362	0.367	0.367	0.363
BERTScore_P	0.298	0.363	0.364	0.364
BERTScore_{F1}	0.324	0.362	0.363	0.361
ETM				
BERTScore_R	0.409	0.390	0.391	0.378
BERTScore_P	0.265	0.267	0.266	0.263
BERTScore_{F1}	0.321	0.316	0.315	0.309
CTM				
BERTScore_R	0.412	0.415	0.416	0.413
BERTScore_P	0.314	0.325	0.324	0.323
BERTScore_{F1}	0.354	0.362	0.362	0.360
BERTopic				
BERTScore_R	0.354	0.358	0.348	0.343
BERTScore_P	0.280	0.315	0.319	0.318
BERTScore_{F1}	0.311	0.334	0.331	0.328
Top2Vec				
BERTScore_R	0.373	0.389	0.393	0.393
BERTScore_P	0.345	0.345	0.344	0.343
BERTScore_{F1}	0.356	0.363	0.364	0.364
C-Top2Vec				
BERTScore_R	**0.420**	**0.443**	**0.447**	**0.447**
BERTScore_P	**0.390**	**0.400**	**0.403**	**0.404**
BERTScore_{F1}	**0.403**	**0.418**	**0.421**	**0.422**

Table 7.16: Topic BERTScore evaluation on the *Yahoo! Answers* dataset.

7.5.7 Topic Information Gain

To evaluate the effectiveness of topic information gain as an evaluation method for topic quality, we compare PWI(T) scores on topics generated by `Top2Vec` and LDA across two datasets: `20 Newsgroups` and `Yahoo Answers`. Results are shown in Tables 7.17, 7.18, 7.19, and **??**.

Across both datasets, `Top2Vec` consistently achieves significantly higher total PWI(T) scores than LDA. On the `20 Newsgroups` dataset, `Top2Vec` yields a total PWI(T) of 996.6,

compared to 360.9 for LDA (Tables 7.17 and 7.18). On the `Yahoo Answers` dataset, the contrast is even more pronounced, with `Top2Vec` achieving 837.6 while LDA scores only 153.3 (Tables 7.19 and **??**). These results indicate that `Top2Vec` topics provide significantly more information about the documents they represent.

Qualitative inspection of the topics supports this finding. `Top2Vec` topics contain informative words and are generally semantically coherent. For instance, Topic 1 in Table 7.19 includes technical terms such as *overwrite*, *debugging*, and *compiler*, while Topic 2 groups finance-related terms like *securities*, *underwriter*, and *issuer*. These topics contain words that are both meaningful and specific to particular subsets of documents, contributing to high PWI(T) scores.

In contrast, many LDA topics contain stop-words or high-frequency terms with limited discriminative power. For example, Topic 10 in Table **??** contains words like *time*, *friend*, *said*, and *got*, which are uninformative across documents and therefore result in low $P(d|w)$ values and low PWI(T) scores. Even after stop-word removal, many LDA topics still exhibit low information gain due to semantic vagueness or lack of topic-specific vocabulary.

These findings demonstrate that PWI(T) provides a more comprehensive evaluation of topic models than coherence metrics alone. While coherence measures capture intra-topic word similarity, PWI(T) directly quantifies how well topic words describe their associated documents. By penalizing frequent, uninformative words and rewarding words that are specific to particular topic documents, PWI(T) offers a robust and interpretable measure of topic model quality.

Topic Number	Topic Words	PWI(T)
1	pitching, pitchers, pitcher, hitter, batting, hit, hitters, baseball, batters, inning	74.2
2	bike, ride, riding, bikes, motorcycle, bikers, helmet, riders, countersteering, passenger	71.9
3	circuit, voltage, circuits, resistor, signal, khz, impedance, analog, diode, resistors	69.1
4	centris, ram, mhz, quadra, nubus, vram, iisi, lciii, cpu, fpu	62.4
5	patient, symptoms, patients, doctor, disease, treatment, jxp, therapy, skepticism, physician	59.1
6	koresh, fbi, compound, batf, davidians, atf, waco, raid, fire, bd	54.7
7	israel, arab, arabs, israeli, jews, palestinians, israelis, war, peace, occupied	54.3
8	orbit, space, launch, orbital, satellites, lunar, shuttle, spacecraft, moon, earth	53.7
9	clipper, nsa, encryption, encrypted, secure, keys, crypto, algorithm, escrow, scheme	51.6
10	controller, drives, drive, ide, scsi, floppy, bios, disk, jumpers, esdi	50.3
11	windows, drivers, ati, cica, driver, exe, card, autoexec, mode, ini	50.1
12	car, engine, cars, ford, brakes, honda, tires, valve, wheel, rear	49.7
13	hockey, playoffs, nhl, game, season, team, playoff, teams, scoring, play	48.0
14	gun, guns, firearms, laws, weapons, handgun, crime, amendment, handguns, firearm	46.6
15	window, application, xlib, manager, openwindows, motif, server, xview, client, clients	43.6
16	jesus, christ, god, bible, church, scripture, christians, scriptures, christian, heaven	38.9
17	postscript, format, printer, fonts, files, formats, font, truetype, bitmap, image	36.7
18	shipping, sale, offer, condition, asking, brand, sell, obo, price, selling	36.0
19	atheists, belief, religion, beliefs, god, christianity, truth, religions, believe, atheist	34.4
20	please, mail, post, email, posting, address, thanks, reply, interested, appreciate	11.3
		996.6

Table 7.17: Per-topic and total topic information gain (PWI(T)) for `Top2Vec` on the *20 Newsgroups* dataset using 20 topics. Each row lists the top 10 topic words and their associated PWI(T) score, quantifying how informative the topic is with respect to its assigned documents. High-scoring topics include domain-specific and semantically coherent terms (e.g., technical, financial, or geopolitical language), demonstrating that `Top2Vec` effectively captures document-specific vocabulary. The total PWI(T) score of 996.6 far exceeds that of LDA, highlighting the superiority of `Top2Vec` in producing informative and discriminative topics.

Topic Number	Topic Words	PWI(T)
1	la, pit, gm, det, bos, tor, pts, chi, vs, min	49.9
2	hz, cx, ww, uw, qs, c_, pl, lk, ck, ah	47.0
3	ax, max, pl, di, tm, ei, giz, wm, bhj, ey	42.6
4	db, period, goal, play, pp, shots, st, power, mov, bh	27.9
5	mk, mm, mp, mh, mu, mr, mj, mo, mq, mx	27.7
6	health, medical, new, study, research, disease, cancer, use, patients, drug	19.0
7	armenian, people, said, one, armenians, turkish, went, us, children, turkey	17.3
8	dos, windows, drive, card, system, disk, mb, scsi, pc, mac	16.7
9	file, program, window, files, image, jpeg, use, windows, display, color	15.7
10	government, president, law, would, mr, israel, state, rights, fbi, states	13.4
11	god, jesus, bible, church, christian, christ, christians, faith, lord, man	12.2
12	game, team, games, hockey, season, teams, league, nhl, new, players	12.0
13	space, nasa, earth, launch, shuttle, orbit, moon, satellite, solar, mission	11.8
14	edu, ftp, graphics, available, pub, image, mail, com, version, also	9.7
15	would, know, anyone, get, thanks, like, one, please, help, could	8.5
16	key, use, data, system, one, information, may, encryption, used, number	7.5
17	people, would, one, think, know, like, say, even, see, way	7.3
18	one, car, would, like, get, time, much, also, back, power	7.1
19	edu, com, please, list, mail, sale, send, email, price, offer	4.0
20	think, year, would, good, time, last, well, get, one, got	3.6
		360.9

Table 7.18: Per-topic and total topic information gain (PWI(T)) for LDA on the *20 Newsgroups* dataset using 20 topics, after stop-word removal. Each topic lists the top 10 words and its PWI(T) score, reflecting how much information the topic contributes about its associated documents. While some topics show modest discriminative power, many contain low-specificity or semantically vague terms, resulting in a relatively low total PWI(T) score of 360.9. Compared to `Top2Vec`, LDA yields significantly less informative and less focused topics, indicating weaker alignment between topic words and document-specific content.

Topic Number	Topic Words	PWI(T)
1	overwrite, rebooting, debug, debugging, reboot, executable, compiler, winxp, xp, winnt	112.2
2	securities, unpaid, equity, purchaser, payment, broker, underwriting, issuer, payable, underwriter	104.4
3	regimen, discomfort, inflammation, swelling, psoriasis, puffiness, inflammatory, irritation, edema, hypertension	98.1
4	realtionship, realationship, insecurities, confide, hurtful, inlove, clingy, friendship, bestfriend, friendships	92.5
5	song, sings, singer, sang, artist, duet, album, lyrics, ballad, vocalist	91.9
6	scripture, believers, righteousness, righteous, pious, spiritual, spirituality, sinful, worldly, discernment	82.2
7	team, players, game, teams, scoring, league, teammate, scorers, playoff, defensively	80.0
8	courses, subjects, curriculum, students, teaching, faculty, syllabus, academic, undergraduate, baccalaureate	78.6
9	war, leaders, politicians, government, democracy, political, terrorists, terrorism, partisan, policies	64.6
10	thus, constant, hence, surface, resulting, greater, therefore, becomes, occurs, larger	33.1
		837.6

Table 7.19: Per-topic and total topic information gain (PWI(T)) for `Top2Vec` on the *Yahoo Answers* dataset using 10 topics. Each topic's top 10 words and corresponding PWI(T) score are listed. The topics exhibit strong semantic coherence and specificity—e.g., technical (Topic 1), financial (Topic 2), and health-related (Topic 3) vocabularies—resulting in a high total PWI(T) of 837.6. These results reflect `Top2Vec`'s effectiveness in capturing highly informative, domain-relevant vocabulary that aligns closely with document content.

Topic Number	Topic Words	PWI(T)
1	team, game, world, win, cup, play, de, football, best, player	23.2
2	computer, yahoo, use, get, click, internet, free, com, need, windows	22.3
3	people, us, country, war, world, would, american, bush, government, america	18.0
4	one, water, two, would, light, number, energy, used, earth, use	16.9
5	body, weight, also, doctor, eat, blood, may, day, get, pain	15.7
6	www, com, http, find, song, name, know, anyone, org, music	14.2
7	get, money, school, would, need, work, pay, good, business, job	13.1
8	god, people, one, life, believe, jesus, word, many, would, us	12.5
9	like, know, get, think, would, want, people, good, really, go	9.1
10	time, like, friend, said, guy, back, would, one, years, got	8.3
		———
		153.3

Table 7.20: Per-topic and total topic information gain (PWI(T)) for LDA on the *Yahoo Answers* dataset using 10 topics, after stop-word removal. The table shows the top 10 words for each topic and their associated PWI(T) scores. Many topics include vague or generic terms that lack strong document-level specificity, resulting in a low total PWI(T) of 153.3. Compared to `Top2Vec`, LDA exhibits weaker topic-document alignment and lower informativeness, demonstrating the limitations of its probabilistic topic generation for this dataset.

7.5.8 LLM Topic Labels

The qualitative results shown in Figure 7.16 demonstrate the coherence and interpretability of topic labels generated using LLMs. Compared to NPMI-based phrase extraction, LLM-generated labels provide more descriptive and human-readable summaries that better capture the overarching themes of each topic. While phrase-based labels such as *"War on Terror"*, *"Christian Faith"*, and *"Operating System"* are often concise and relevant, they tend to reflect individual phrase occurrences and may lack the broader contextual understanding. In contrast, LLM-derived labels such as *"Domestic and Foreign Policy Issues"* or *"Comparative Religion with a Focus on Christianity"* offer higher-level abstractions that more clearly communicate the topic.

This distinction is further supported by the BERTScore$_R$ results shown in Table 7.21, where LLM-generated labels achieve the highest relevance scores across several topics. For instance, the label *"Computer Hardware and Video Graphics Cards"* scores 0.565,

outperforming phrase-based and `Top2Vec` methods on comparable topics. These results suggest that LLMs are capable of summarizing diverse and representative content into coherent summaries, forming semantically coherent and easily intepretable topic labels.

Overall, the qualitative evidence shows the strength of LLM-based labeling and suggest that it is a good direction for future research. Despite being more computationally intensive, this approach offers more control over label granularity and delivers more coherent summaries, especially for large or complex topic clusters. Future work can explore improvements in prompt engineering, document selection, and scalable inference techniques to further improve performance and applicability.

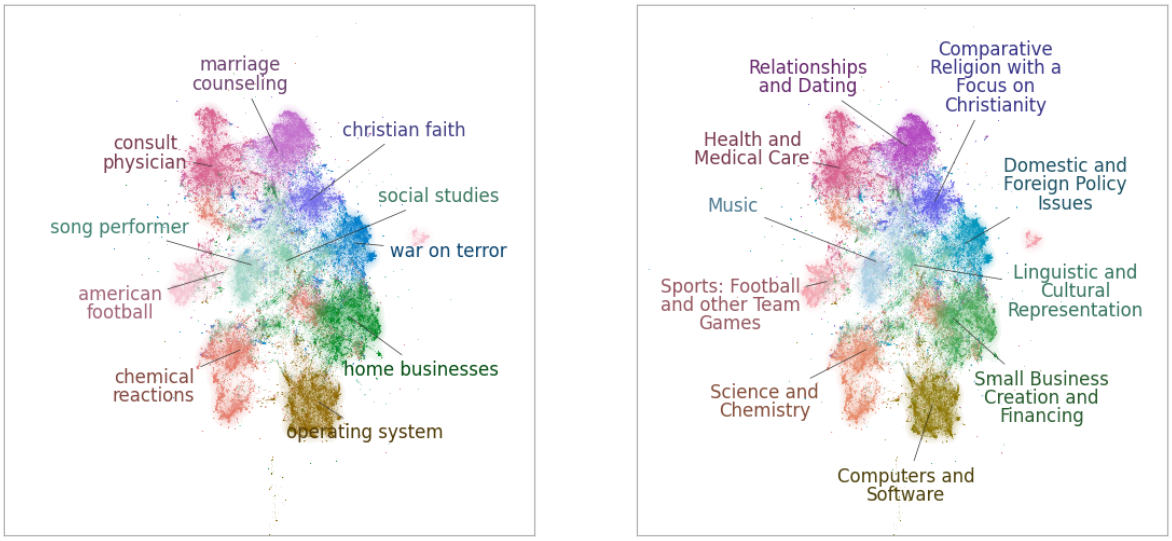


Figure 7.16: Comparison of topic labels on the *YahooAnswers* dataset. Phrase-based labels are generally coherent and readable, but LLM-generated labels tend to offer more descriptive and semantically rich summaries.

7.5.9 Qualitative Analysis

We present a qualitative assessment of the C-Top2Vec model by analyzing the structure and semantic quality of its topic representations. We show our sub-document vectors in a 2D UMAP plot along with the topic label assigned by `C-Top2Vec` for 20 topics on the *20newsgroups* dataset in Figure 7.17. In Figure 7.18, we show the topic spans assigned by `C-Top2Vec` visualized with `displacy` [52]. In Table 7.21, we show the top 2 and the bottom 2 topics based on their $\text{BERTScore}_{\text{R}}$ from the top performing models.

Topic Labels	$\textbf{BERTScore}_R$
C-Top2Vec - LLM labels	
Computer Hardware and Video Graphics Cards	0.565
Graphics Programming and Related Resources	0.518
Baseball Statistics and Team Standings	0.453
Debates and Diversity within Religion and Atheism	0.409
C-Top2Vec - Phrases	
turkish genocide, victims of the turkish, genocide of the muslims	0.564
sports car, sports cars, mustang gt, toyota celica, sport coupe, honda accord	0.515
christian doctrine, bible contradictions, biblical interpretation, biblical contradictions	0.412
introduction to atheism, religious sects, atheist position, religious persecution	0.383
Top2Vec	
graphics, graphical, freeware, tiff, bitmap, software, toolkits, fortran, adobe, pixmap	0.500
xmu, xterm, libxmu, xdm, xfree, openwindows, gui, xm, interface, xloadimage	0.448
propaganda, discussions, discussion, threatened, debate, rushdie, argument, arguments	0.359
prices, price, deals, sale, purchasing, sells, interested, offers, selling, buyer	0.351
CTM	
key, secure, keys, security, chip, encryption, government, use, algorithm, secret	0.480
dos, ftp, graphics, code, software, pub, available, files, unix, pc	0.457
hr, suggested, un, remain, frequently, abuse, covered, bj, structure, capable	0.316
hr, suggested, dod, un, remain, changes, consistent, connected, capable, ordered	0.315

Table 7.21: Top-2 and bottom-2 scoring topics (by BERTScore_R) across various models on the *20newsgroups* dataset. We compare topic labels generated by LLMs, phrase-based methods, and baseline models such as `Top2Vec` and CTM. LLM-generated labels achieve the highest BERTScore_R, however phrase-based methods also perform well and are only slightly less coherent.

UMAP Topic Cluster Visualization

Figure 7.17 visualizes the sub-document vectors projected into 2D using UMAP for the *20newsgroups* dataset. Each cluster corresponds to a discovered topic generated by `C-Top2Vec`, annotated with the nearest phrase label to each topic vector.

The clusters are clearly separated with minimal overlap, indicating strong topical coherence. Similar topics—such as `sports car` and `rec motorcycles`—appear adjacent to one another, supporting the model's ability to preserve local semantic structure. Densely packed regions further reflect intra-topic consistency, suggesting the windowed vector representation successfully captures discrete thematic units.

Topic Span Visualization

In Figure 7.18, we visualize the extracted topic spans within a sample document using `displacy`. C-Top2Vec accurately identifies thematically consistent segments related to

encryption and surveillance, with different thematic areas being segmented and labeled accurately.

This highlights the interpretability benefit of the model: it not only discovers topics but localizes them at the sub-document level. The ability to visualize where a topic resides within a document supports downstream use cases such as summarization and information retrieval.

Topic Label Coherence Analysis

In table 7.21 we present a comparison of topic coherence across the top performing models using $\text{BERTScore}_{\text{R}}$, highlighting several key trends. C-Top2Vec with LLM-generated labels achieves the highest coherence overall, producing specific and interpretable labels such as *Computer Hardware and Video Graphics Cards* (0.565). Following closely, C-Top2Vec with extracted phrases from the corpus also demonstrates strong performance, with coherent phrases that correpsond to specific topics. In contrast, Top2Vec, which relies on single words rather than phrases, shows moderate coherence; while some topics remain interpretable, coherence deteriorates at the lower end with vague or repetitive terms such as *prices, price, deals, sale* (0.351). CTM ranks the lowest, with its weakest topics dominated by generic or fragmented tokens (e.g., *hr, suggested, dod, un*), yielding the lowest BERTScores (approximately 0.315) and lacking semantic clarity. These results indicate that C-Top2Vec not only leads in top-performing topics but also preserves coherence better than other methods even in its lowest-scoring cases.

These findings qualitatively demonstrate the value of using language model-based representations as well as phrases for topic labeling. C-Top2Vec's ability to generate both coherent labels and interpretable phrase labels suggests it is more robust to topic ambiguity and noise, making it a more reliable tool for downstream tasks such as document summarization, or information retrieval tasks.

Conclusion These qualitative observations reinforce that C-Top2Vec and Top2Vec are capable of discovering highly interpretable and coherent topic structures. The visual separation of clusters, accurate span segmentation, and coherent topic labels together demonstrate the effectiveness of the model's performance.

7.5.10 Topic Span Effect Analysis

We assess our topic span segmentation with BERTScore using three variations: the complete document, solely the topic segments, and the non-topic segments. The same datasets and configuration as in the BERTScore Evaluation, Section 7.5.6, are used.

The results shown in Table 7.22, demonstrate that `C-Top2Vec` correctly selects the topic segments as the topic span $\text{BERTScore}_\text{R}$ remains high when non-topic spans are removed and is much lower when non-topic spans are used. This is because the $\text{BERTScore}_\text{R}$ for the extracted topic spans is almost the same to that of the full document in both datasets. Since BERTScore uses a maximum similarity, using on the highest similarity between token embeddings, its score reflects the most semantically relevant content within a text segment.

If the topic spans extracted by the model accurately capture the the topic specific segments, then removing the non-topic spans should not significantly reduce the $\text{BERTScore}_\text{R}$. This is what we observe: for *20newsgroups*, the $\text{BERTScore}_\text{R}$ is 0.457 for the full document and 0.456 for the topic spans. Likewise, in *YahooAnswers*, the scores are 0.440 and 0.439 respectively. The minimal difference confirms that the selected topic spans contain nearly all the semantical content that contributes to document-level topic similarity.

In contrast, when evaluating only the non-topic spans, the $\text{BERTScore}_\text{R}$ drops significantly, to 0.365 in *20newsgroups* and 0.313 in *YahooAnswers*. This large gap indicates that the non-topic segments lack key semantic content relevant to the topic confirming they are not topic segemnts.

These findings validate that `C-Top2Vec` accurately segments documents into topic segments, ensuring that its topic segmentation aligns with the most relevant parts of the document for each topic as measured by $\text{BERTScore}_\text{R}$.

7.5.11 Window Size Effect Analysis

We evaluate the impact of window size on our multi-vector document representation by calculating the Silhouette Score [96] for UMAP-reduced vectors, using the dataset's topic labels as a reference. Table 7.23 reports results for various window sizes across two datasets.

We observe that both very small windows with 15 tokens and using all tokens, which is equivalent to a single document vector, result in lower Silhouette Scores which indicates poor topic cluster separation. In contrast, a window size of 50 achieves the highest score on both datasets, suggesting an optimal trade-off between semantic granularity and contextual coherence.

Configuration	$\textbf{BERT}_{\textbf{Score}_R}$
20newsgroups	
Whole Document	0.457
Topic Spans	0.456
Non-topic Spans	0.365
YahooAnswers	
Whole Document	0.440
Topic Spans	0.439
Non-topic Spans	0.313

Table 7.22: Evaluation of topic span segmentation using BERTScore_R on the *20newsgroups* and *YahooAnswers* datasets. We compare the BERTScore_R of full documents, extracted topic spans, and non-topic spans. Results show that topic spans selected by `C-Top2Vec` achieve nearly identical scores to the full document, while non-topic spans yield substantially lower scores. This confirms that `C-Top2Vec` accurately segments documents and captures the semantically relevant content for each topic.

The window size determines how many contextualized token embeddings are pooled into each sub-document vector. Smaller windows may lack sufficient context, while larger windows dilute local topic specificity. A moderate window size (50 tokens) preserves semantic diversity without sacrificing coherence, leading to better-separated clusters in the embedding space.

These findings support the benefit of multi-vector document representations over single-vector approaches, especially when using an appropriately tuned window size.

7.6 Summary

This chapter evaluates `Top2Vec` and `C-Top2Vec` against traditional methods (LDA, ETM) and neural models (CTM, BERTopic), using the *20Newsgroups* and *Yahoo! Answers* datasets. The evaluation focuses on topic coherence, document-topic assignment, and topic document representativeness. In addition to comparing topic models, we systematically evaluate clustering methods, showing that UMAP followed by HDBSCAN outperforms both traditional and deep clustering models in preserving semantic structure. A combined clustering score (AMI, ARI, purity, and clustered proportion) was introduced to optimize UMAP and HDBSCAN parameters. We demonstrate that `C-Top2Vec` consistently out-

Window size	Silhoutte Score
20newsgroups	
15	0.140
25	0.182
50	**0.194**
100	0.192
All tokens	0.133
YahooAnswers	
15	0.144
25	0.152
50	**0.162**
100	0.127
All tokens	0.114

Table 7.23: Impact of window size on topic cluster separation for multi-vector document representations, evaluated using Silhouette Score on UMAP-reduced embeddings for the *20newsgroups* and *YahooAnswers* datasets. Results show that a window size of 50 tokens yields the highest Silhouette Scores on both datasets, indicating optimal topic separation. Very small windows (15 tokens) and full-document vectors perform worse, suggesting that moderate segmentation provides the best balance between local specificity and global context. These findings highlight the importance of window size tuning in segment-based document encoding.

performs other models across these metrics while effectively segmenting documents into meaningful topic spans. Insights into model performance, topic segmentation, and the impact of document representation are presented. The key points from the results are:

- **Datasets:** Evaluated on *20Newsgroups* and *Yahoo! Answers* which were chosen due to their size and number of topic labels
- **Experimental Setup:** Models evaluated include LDA, ETM, CTM, BERTopic, Top2Vec, and C-Top2Vec. Neural models used `all-mpnet-base-v2` embeddings, except ETM in order to create a level playing field.
- **Clustering Performance:** UMAP followed by HDBSCAN outperformed all traditional and deep clustering methods in preserving semantic structure. Deep clustering models like DEC, DCN, DipDECK, and Top-k SAE showed limited improvement over the baseline of K-means.

- **Hyperparameter Optimization:** A combined clustering score (AMI, ARI, purity, and clustered proportion) was introduced to optimize UMAP and HDBSCAN parameters.
- **Hierarchical Topic Modeling:** Increasing `n_neighbors` in UMAP in tandem with increasing `min_cluster_size` in HDBSCAN improves capturing of global structure, enabling coarse-to-fine topic resolutions and making it suitable for hierarchical topic modeling.
- **Two-Stage Clustering Yields Complete and Diverse Topic Hierarchies:** Our two-stage approach preserves all fine-grained topics across levels, resulting in full hierarchical coverage, higher semantic diversity, and comprehensive multi-resolution topic exploration.
- **Topic Coherence Evaluation:** Metrics evaluated were $C_{NPMI}, C_V, C_{WE}, C_{SBERT}$. C-Top2Vec outperformed models on most metrics or was not far behind other methods. This demonstrates its ability to generate coherent topics based on traditional evaluations of coherence.
- **Topic Assignment Evaluation:** Metrics evaluated are AMI and ARI. C-Top2Vec achieved the best AMI and ARI scores, demonstrating superior topic assignment relative to human topic labels.
- **Topic BERTScore Evaluation:** We propose using BERTScore for topic model evaluation, in order to evaluate both coherence and how representative topics are of their underlying documents. C-Top2Vec achieved the highest BERTScore recall and precision, indicating well-represented and coherent topics based on our proposed comprehensive evaluation.
- **Topic Span Effect Analysis:** Comparison of BERTScores for topic spans and non-topic validated the model's effectiveness in identifying relevant topic segments within a document.
- **Window Size Effect Analysis:** A window size of 50 produced the highest Silhouette scores, compared to single document vector and other granularities, demonstrating the effectiveness of multi-vector document representation.
- **Topic Information Gain:** We proposed PWI(T), a topic evaluation metric which evalutes topic coherence as well as how representative topics are of their underlying documents. We show that `Top2Vec` topics consistently had higher information gain than LDA across the datasets.

- **LLM Topic Labels:** LLM-generated topic labels were more descriptive than NPMI phrase-based labels, showing promise for improving topic interpretability.

In the next chapter, we conclude our findings and discuss limitations and directions of future work.

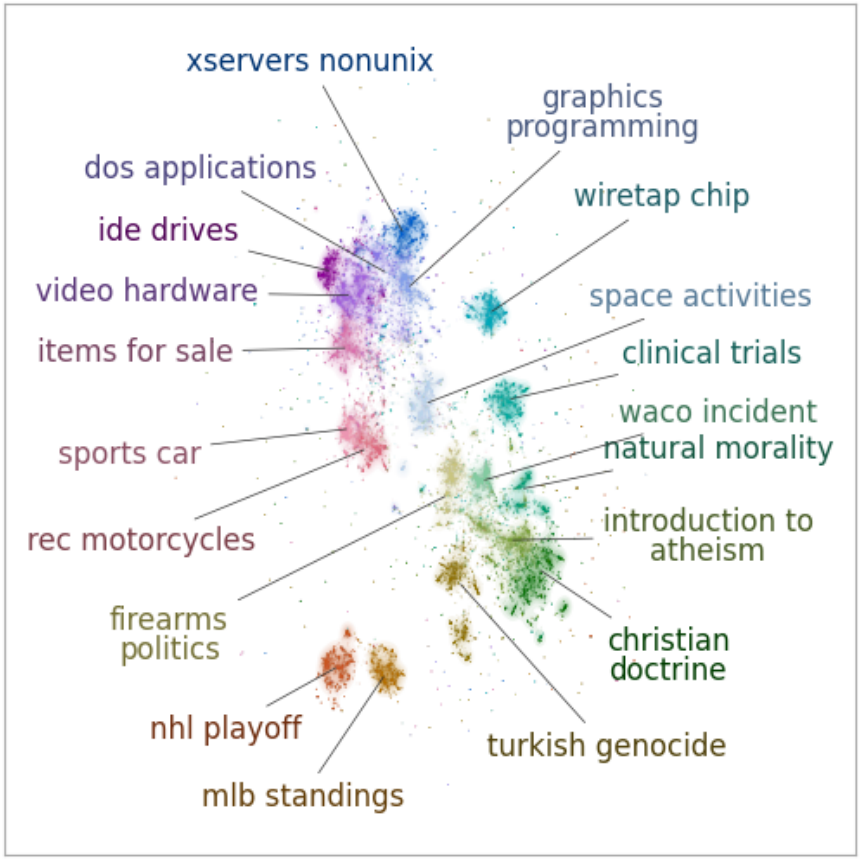


Figure 7.17: 2D UMAP projection of sub-document vectors on the *20newsgroups* dataset, with phrase-based topic labels from `C-Top2Vec`. The clusters exhibit strong topical coherence and spatial separation, illustrating effective semantic structuring of topics.

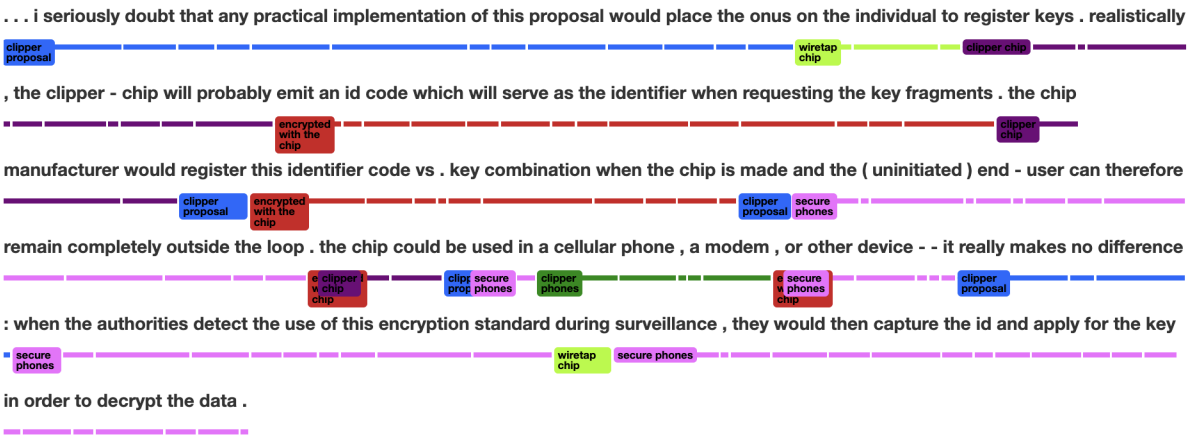


Figure 7.18: Visualization of topic spans within a document using `displacy`. The model identifies and segments semantically consistent regions, illustrating its ability to localize topics at the sub-document level and enhance interpretability for downstream tasks.

Chapter 8

Conclusion and Future Work

In this thesis, we redefine topic modeling by moving beyond the traditional view of topics as distributions over a vocabulary. We redefine topics to emphasize how informative they are of their underlying documents. We define a topic as a high-level, information-dense abstraction that captures key semantic content. It can be seen as a high-level summary, distilling the most important semantic elements of the underlying content. In order to satisfy this definition we propose a novel way to represent topics as well as evaluate topic models.

Traditional models, such as LDA, define topics as probability distributions over words and model documents as mixtures of these topic-word distributions. While useful, this approach prioritizes reconstructing word frequencies over semantic informativeness, often resulting in topics dominated by common or ambiguous words. To address these limitations, a range of neural topic modeling methods have been proposed, such as ETM and CTM. These models as well as others incorporate neural representations of text to improve semantic coherence and topic quality. However, despite these advances, most neural models still rely on a distributional definition of topics, suffer from interpretability issues, and often require pre-specifying the number of topics. In contrast, we propose representing topics as vectors in a joint semantic embedding space of documents and words. These *topic vectors* are computed from dense areas of semantically similar document embeddings and labeled using the nearest word or phrase embeddings, ensuring both informativeness and interpretability.

We introduce `Top2Vec`, a topic modeling approach that leverages distributed representations of text, manifold learning and density based clustering. It finds topic vectors based on density in the document vector space without requiring prior knowledge of the

number of topics in the corpus. It leverages joint embedding of topic vectors and word vectors to find informative topic labels without the need for stop-word removal, stemming or lematization. We extend it by proposing `C-Top2Vec` which leverages contextual token embeddings to create multi-vector document representations that capture topical information from each segment of a document. This approach segments documents into topic spans, enabling detailed and granular analysis of topics within documents. Further, it produces a document topic distribution and a document topic relevance score which allows for ranking of document segments according to their relevance to a specific topic. It also uses phrases to label topics rather than just words for improved interpretability. `Top2Vec` and `C-Top2Vec` support hierarchical topic reduction, enabling them to find topics at various levels of granularity.

We also argue that current evaluation methods are limited by focusing mainly on topic coherence metrics derived from word co-occurrence statistics. These fail to capture how well topics represent the documents to which they are assigned. To address this, we introduce new evaluation techniques, Topic Information Gain and BERTScore. They both evaluate both topic coherence and how informative topics are of their underlying documents overcoming previous gaps in evaluation. Together, our proposed representation and evaluation frameworks offer a more meaningful and accurate approach to topic modeling that matches our definition of topics. Our findings, based on a comprehensive set of topic modeling evaluation metrics, demonstrate that our approach outperforms current state-of-the-art models.

8.1 Thesis Contributions

The key contributions of this thesis are summarized as follows:

- **Top2Vec**: We introduced `Top2Vec`, a topic modeling approach that leverages semantic embeddings of documents and words to automatically discover topics as dense areas in the embedding space. It automatically finds the number of topics and find topic vectors. It does not require pre-defined number of topics, preprocessing like stop-word removal, or lemmatization. Topics are labeled using the nearest word vectors in the joint embedding space.
- **C-Top2Vec**: We extended `Top2Vec` with contextual token embeddings to develop `C-Top2Vec`, a model capable of producing multi-vector representations of documents and segmenting documents into topic spans by assigning topics at the token level. It enables fine-grained topic discovery and interpretability at the segment level.

- **Clustering Methods for Topic Modeling Evaluation**: Through a comprehensive study of clustering methods and dimensionality reduction techniques, we demonstrated that the combination of UMAP and HDBSCAN offers the most effective, scalable, and interpretable clustering for embedding-based topic modeling. We also used combined clustering score (AMI, ARI, purity, and clustered proportion) to optimize UMAP and HDBSCAN parameters.

- **Hierarchical Topics**: We explored several methods for reducing topics and building topic hierarchies: greedy sized based topic merging, UMAP and HDBSCAN parameter tuning for global versus local structure and finally our twos stage approach. This method first clusters documents into fine-grained topics with UMAP and HDBSCAN, then applies agglomerative clustering on the resulting topic vectors, which produces a full hierarchical tree that retains all topics, enabling multi-resolution exploration of the topic space without losing important granular clusters.

- **Phrase-based and LLM Topic Labels**: We improved topic interpretability through use of the topic vectors and finding nearest labels in the embedding space. This was done with word vectors and phrase-based labels using NPMI to find coherence phrases from the corpus. Lastly we explored the utility of LLM-generated labels for producing semantically rich topic names by leveraging topic vectors to find the most relevant documents and phrases to use in an LLM prompt.

- **Topic Span Identification and Relevance Ranking**: Our approach introduced token-level topic assignment, enabling document segmentation and ranking of topic-relevant spans which capabilities not supported by traditional models.

- **Evaluation Beyond Coherence**: We introduced a comprehensive evaluation framework, including BERTScore and Topic Information Gain, to assess both topic coherence and document representativeness. We demonstrated that `C-Top2Vec` achieves superior performance across coherence, clustering alignment (AMI, ARI), and BERTScore topic evaluation.

8.2 Research Questions Revisited

Based on the proposed methods and findings outlined in this thesis, we provide the following answers to the research questions:

- **RQ1 - Neural Topic Representations:** We demonstrate that the combination of UMAP and HDBSCAN leads to the best results for finding semantic structure in a semnatic embedding space allowing us to find topic vectors. By representing topics as vectors in a semantic embedding space, we demonstrated that distributed embeddings provide a powerful, interpretable, and flexible way to represent topics.
- **RQ2 - Informative Labels:** Through nearest neighbor word and phrase embeddings and LLM-based generation, we produced topic labels that are semantically meaningful and more interpretable than traditional ranked word lists.
- **RQ3 - Topic Hierarchies:** By adjusting clustering granularity, using greedy size-based topic reduction, and our two-stage approach we demonstrated multi-resolution topic discovery.
- **RQ4 - Topic Segmentation:** We proposed a novel method for segmenting documents into topic spans by assigning topics at the token level by leveraging contextualized embeddings and topic vectors, enabling fine-grained topic understanding within documents.
- **RQ5 - Topic Evaluation:** We introduced the metrics Topic Information Gain and BERTScore to assess how coherent topics are as well how well they represent underlying documents, addressing existing limitations of over-reliance on topic coherence evaluations.

8.3 Limitations

Although our use of UMAP and HDBSCAN provides strong performance, these methods can be computationally expensive on very large datasets. For very large datasets the use of a GPU may be required and for certain hyperparameter settings and computational complexity may be high. Additionally HDBSCAN in certain settings can be sensitive to extreme outliers and requires careful examination of the results.

`Top2Vec` model represents each document using a single embedding vector, which can limit its ability to capture multi-topic documents making it very useful for single topic or short documents but less useful for long documents containing multuple topics. While `C-Top2Vec` addresses this using a sliding window over contextual embeddings, it increases computational complexity and may require tuning of window size and stride.

Using pre-trained embedding models in neural topic modeling offers significant benefit by bringing external knowledge; however, these models may not fully capture the nuances of a domain specific corpus. Additionally, these models may encode biases from their training data and may not perform equally well across different domains, low-resource languages, or specialized vocabularies. All of these factors can potentially affect the accuracy of the topic models.

While topic vectors in a semantic embedding space offer a more flexible and informative representation than word distributions, the interpretability of dense vector spaces can be limited. Unlike probabilistic models where topic-word distributions have an explicit meaning, embedding distances are relative and not always easily interpretable.

Although we propose novel evaluation metrics like BERTScore and Topic Information Gain, BERTScore still depends on a pre-trained models and may not align perfectly with human judgment in all contexts. Additionally, there is a lack of comprehensive benchmarks for evaluating document-level topic segmentation. In general, topic model evaluation measures have limitations as they cannot fully capture the nuances of natural language. Model evaluation should not rely solely on automated metrics. Incorporating human judgment, via qualitative analysis, expert annotation, or user studies as well as task-based evaluations tailored to real-world applications, is essential for a more robust and reliable assessment. Ultimately, the effectiveness of any topic modeling approach is highly dependent on the specific use case and dataset; a custom evaluation strategy should always be adopted.

8.4 Future Work

8.4.1 Embedding Models

One key area for future work is finding the middle ground between learning text embeddings directly from the corpus, which may suffer from lack of information to learn meaningful representations versus using pre-trained embedding models which may not capture the nuances of a specific corpus domain. An unsupervised method to fine-tune a pre-trained embedding model may greatly improve topic modeling by balancing the benefits of general semantic knowledge with corpus-specific adaptation.

8.4.2 Clustering and Representation Learning

Although UMAP and HDBSCAN have proven to be a strong combination for clustering, deep clustering methods such as those based on sparse autoencoders have significant promise due to their ability to capture multiple topics per document. While these methods did not perform well in our initial experiments, a dedicated study may uncover strategies to improve representation learning, and better align latent dimensions with interpretable topics. Such advancements could enable deep clustering to serve as a more flexible and powerful alternative for multi-topic document modeling. SAEs in general hold a lot of promise due to their ability to assign multiple latent vectors to a single vector.

8.4.3 Topic Labeling

An important direction for future research is the formalization and optimization of LLM-based topic labeling, which has shown promising results in our preliminary implementation and experiments. This approach leverages our topic vectors to find the most representative documents and phrases to summarize a topic and generate a much more interpretable label. Future work could focus on formalizing the method by defining rigorous input selection criteria for documents and phrases, evaluating the impact of prompt structure and model architecture, and integrating mechanisms such as MMR for selecting diverse yet relevant example documents and phrases. Additionally, systematic evaluation through automatic metrics will be essential to assess improvements in interpretability, coherence, and generalization across domains.

8.4.4 Evaluation Metrics

Although BERTScore is a very useful and informative evaluation score for topic coherence and document-topic representativeness it has two major limitations. Firstly, it relies on pre-trained language models such as BERT, which may not fully align with the semantics of specific domains and can introduce biases or innacurate similarities in specialized domains. Second, its evaluation process is computationally expensive, $O(N^2)$, requiring token-level similarity comparisons between each topic word and every token in the associated documents. Finding more efficient evaluation metrics that accurately capture the specific semantics of a corpus domain would have a greatly improve the state of topic model evaluation. Another area to explore is the use of LLMs for evaluation of topics, given the difficulty of performing human evaluation.

8.5 Final Remarks

This thesis set out to challenge the foundations of topic modeling by redefining what a topic is, how it should be represented, and how it should be evaluated. Traditional topic models which are based on probabilistic distributions over vocabularies have been valuable for uncovering topics. However, their reliance on word co-occurrence statistics often fail to capture the rich semantic structures present in natural language, limiting both their interpretability and usefulness.

We proposed representing topics as *distributed vectors* within a joint semantic space of documents, words, and phrases. This distributed representation allows topics to emerge as dense regions in a continuous topic space, indicating an underlying topic shared by documents. The *topic vector* represents the most semantically central point of a topic, from which the most informative and contextually relevant words or phrases can be identified. This approach facilitates the discovery of topics that more accurately capture colloquial language usage.

Through distributed representations of topics, we unlock capabilities like automatic topic discovery, topic hierarchies, document topic segmentation, phrase-based labeling and use of topic vectors in retrieval and similarity-based operations. Our evaluation framework further supports this by measuring not only coherence but also how informative and representative topics are of their underlying documents.

By leveraging *distributed representations of topics*, we advance toward topic modeling language not only by what is said, but by what is meant.

References

[1] Aly Abdelrazek, Yomna Eid, Eman Gawish, Walaa Medhat, and Ahmed Hassan. Topic modeling algorithms and applications: A survey. *Information Systems*, 112:102131, 2023.

[2] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.

[3] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[4] Tuncer AKBAY. Modeling education studies indexed in web of science using natural language processing. *Instructional Technology and Lifelong Learning*, 3(2):129–143, 2022.

[5] Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th international conference on computational semantics (IWCS 2013)–Long Papers*, pages 13–22, 2013.

[6] Dimo Angelov. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*, 2020.

[7] Dimo Angelov and Diana Inkpen. Topic modeling: Contextual token embeddings are all you need. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13528–13539, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[8] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[9] Zhuang Bairong, Wang Wenbo, Li Zhiyu, Zheng Chonghui, and Takahiro Shinozaki. Comparative analysis of word embedding methods for dstc6 end-to-end conversation modeling track.

[10] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[11] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[12] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *Database Theory—ICDT'99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7*, pages 217–235. Springer, 1999.

[13] Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. An automatic approach for document-level topic model evaluation. In Roger Levy and Lucia Specia, editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 206–215, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[14] Federico Bianchi, Silvia Terragni, and Dirk Hovy. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 759–766, Online, August 2021. Association for Computational Linguistics.

[15] Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. Cross-lingual contextualized topic models with zero-shot learning. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1676–1683, Online, April 2021. Association for Computational Linguistics.

[16] Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. Cross-lingual contextualized topic models with zero-shot learning. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1676–1683, Online, April 2021. Association for Computational Linguistics.

[17] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[18] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of the Biennial GSCL Conference 2009*, 01 2009.

[19] Sophie Burkhardt and Stefan Kramer. Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model. *Journal of Machine Learning Research*, 20(131):1–27, 2019.

[20] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[21] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[22] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.

[23] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[24] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22, 2009.

[25] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941, 2014.

[26] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[27] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors, *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics.

[28] M Cover Thomas and A Thomas Joy. Elements of information theory. *New York: Wiley*, 3:37–38, 1991.

[29] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[30] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[31] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[33] Adji B Dieng, Francisco JR Ruiz, and David M Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 2020.

[34] Ran Ding, Ramesh Nallapati, and Bing Xiang. Coherence-aware neural topic modeling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 830–836, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[35] Thanh-Nam Doan and Tuan-Anh Hoang. Benchmarking neural topic models: An empirical study. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4363–4368, 2021.

[36] Caitlin Doogan and Wray Buntine. Topic model or topic twaddle? re-evaluating semantic interpretability measures. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3824–3848, Online, June 2021. Association for Computational Linguistics.

[37] Roman Egger and Joanne Yu. A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Frontiers in sociology*, 7:886498, 2022.

[38] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[39] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[40] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384, 2006.

[41] Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. Topics in semantic representation. *Psychological review*, 114(2):211, 2007.

[42] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[43] Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223, March 2017.

[44] John A Hartigan and Pamela M Hartigan. The dip test of unimodality. *The annals of Statistics*, pages 70–84, 1985.

[45] Philip J. Hayes and Steven P. Weinstein. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In *Second Annual Conference on Innovative Applications of Artificial Intelligence*, 1990.

[46] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*, 2021.

[47] John Healy and Leland McInnes. Uniform manifold approximation and projection. *Nature Reviews Methods Primers*, 4(1):82, 2024.

[48] Alexander Hinneburg, Charu C Aggarwal, and Daniel A Keim. What is the nearest neighbor in high dimensional spaces? *Proceedings of the 26th VLDB Conference*, 2000.

[49] Geoffrey E Hinton et al. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

[50] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[51] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88, 2010.

[52] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. spacy: Industrial-strength natural language processing in python. 2020.

[53] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[54] Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. Is automated topic model evaluation broken? the incoherence of coherence. *Advances in neural information processing systems*, 34:2018–2033, 2021.

[55] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

[56] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[57] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[58] Qiang Jipeng, Qian Zhenyu, Li Yun, Yuan Yunhao, and Wu Xindong. Short text topic modeling techniques, applications, and performance: a survey. *arXiv preprint arXiv:1904.07695*, 2019.

[59] Bradley Karas, Sue Qu, Yanji Xu, and Qian Zhu. Experiments with lda and top2vec for embedded topic discovery on social media data—a case study of cystic fibrosis. *Frontiers in Artificial Intelligence*, 5:948313, 2022.

[60] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[61] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[62] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *Acm transactions on knowledge discovery from data (tkdd)*, 3(1):1–58, 2009.

[63] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[64] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86, Berlin, Germany, August 2016. Association for Computational Linguistics.

[65] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *ArXiv*, abs/1405.4053, 2014.

[66] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[67] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755):788–791, 1999.

[68] Collin Leiber, Lena GM Bauer, Benjamin Schelling, Christian Böhm, and Claudia Plant. Dip-based deep embedded clustering with k-estimation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 903–913, 2021.

[69] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[70] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[71] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021.

[72] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pages 281–298. University of California press, 1967.

[73] Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

[74] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov 2017.

[75] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[76] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[77] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[78] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[79] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[80] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[81] David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 262–272, 2011.

[82] Khalil Mrini, Can Liu, and Markus Dreyer. Rewards with negative examples for reinforced topic-focused abstractive summarization. In Giuseppe Carenini, Jackie Chi Kit Cheung, Yue Dong, Fei Liu, and Lu Wang, editors, *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 33–38, Online and in Dominican Republic, November 2021. Association for Computational Linguistics.

[83] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual confer-*

ence of the North American chapter of the association for computational linguistics, pages 100–108, 2010.

[84] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[85] Charles O'Neill, Christine Ye, Kartheik Iyer, and John F Wu. Disentangling dense embeddings with sparse autoencoders. *arXiv preprint arXiv:2408.00657*, 2024.

[86] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1895–1901. IJCAI/AAAI Press, 2016.

[87] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[88] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[89] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[90] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[91] Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann Cun. Efficient learning of sparse representations with an energy-based model. *Advances in neural information processing systems*, 19, 2006.

[92] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP*

Frameworks, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[93] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[94] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, S Yu Philip, and Lifang He. Deep clustering: A comprehensive survey. *IEEE transactions on neural networks and learning systems*, 2024.

[95] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408, 2015.

[96] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[97] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985.

[98] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020.

[99] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[100] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[101] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*, 2017.

[102] Asbjørn Steinskog, Jonas Therkelsen, and Björn Gambäck. Twitter topic modeling by tweet aggregation. In *Proceedings of the 21st nordic conference on computational linguistics*, pages 77–86, 2017.

[103] Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260, 2003.

[104] Silvia Terragni, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano, and Antonio Candelieri. OCTIS: Comparing and optimizing topic models is simple! In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 263–270. Association for Computational Linguistics, April 2021.

[105] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[106] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[107] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[108] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1073–1080, New York, NY, USA, 2009. Association for Computing Machinery.

[109] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[110] Henry Widdowson. J.R. Firth, 1957, papers in linguistics 1934–51. *International Journal of Applied Linguistics*, 17:402 – 413, 10 2007.

[111] Xiaobao Wu, Thong Nguyen, and Anh Tuan Luu. A survey on neural topic models: Methods, applications, and challenges. *Artificial Intelligence Review*, 57(2):1–30, 2024.

[112] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[113] Jinming Xing, Dongwen Luo, Chang Xue, and Ruilin Xing. Comparative analysis of pooling mechanisms in llms: A sentiment analysis perspective. *arXiv preprint arXiv:2411.14654*, 2024.

[114] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR, 2017.

[115] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. Multilingual universal sentence encoder for semantic retrieval. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online, July 2020. Association for Computational Linguistics.

[116] Wenting Yi, Siqi Bu, Hiu-Hung Lee, and Chun-Hung Chan. Comparative analysis of manifold learning-based dimension reduction methods: A mathematical perspective. *Mathematics*, 12(15):2388, 2024.

[117] Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. Multi-view document representation learning for open-domain dense retrieval. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[118] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.

[119] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[120] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[121] Zihan Zhang, Meng Fang, Ling Chen, and Mohammad Reza Namazi Rad. Is neural topic modelling better than clustering? an empirical study on clustering with contextual embeddings for topics. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3886–3893, Seattle, United States, July 2022. Association for Computational Linguistics.