CBS
COPENHAGEN
BUSINESS SCHOOL
HANDELSHØJSKOLEN

Data-Driven Podcast Advertising: A Novel Framework

Exploring the Potential of Topic Modelling and Text Segmentation for Native Ad Placement

Authors
Oskar Munck af Rosenschöld (149873)
Ramon Habtezghi (149885)

Supervisor
Daniel Hain

Master Thesis

MSc in Business Administration & Data Science
Department of Digitalization
Copenhagen Business School

15th of May 2023

Total pages: 64 (86 normal pages)

Total characters: 164,650

Abstract

The growing podcast industry necessitates innovative advertising strategies, prompting this study to explore the use of data science methods to enable native advertisement placement, in which ads align with the surrounding content. This research seeks to locate advertisement spots at points of topical shifts as well as assign meaningful topics to the content surrounding those shifts in podcast transcripts. A transformer-based clustering approach, integrated with a text segmentation algorithm, is developed for this purpose, advancing previous literature in text segmentation. By modelling the Spotify Podcast Dataset, the developed methodology's ability to identify meaningful advertisement spots in podcasts and assign topics from the corpus to these segments is validated. This proof-of-concept study not only technically enables native advertising but also proposes a business framework for its monetization, outlining potential integration into podcast platforms. The study also positions the relevance of the methodology in relation to network effects and platform theory.

Keywords: Cluster-Based Topic Modelling, Text Segmentation, TopicTiling, BERTopic, Podcasts, Native Advertisement

Acknowledgements

We would like to express our deepest gratitude to our supervisor, Daniel Hain, for his invaluable guidance and unwavering support throughout this thesis. Dr. Hain's expertise in Natural Language Processing and academic research has been instrumental in shaping this study. His valuable feedback, insightful discussions, and challenging questions have greatly contributed to the quality of our work.

We would also like to thank Spotify for providing us with access to the Spotify Podcast Data, which has been crucial in conducting our research.

Finally, we extend our heartfelt thanks to our families and friends for their unwavering support and encouragement throughout this journey. Their belief in us and their continuous encouragement has been a constant source of motivation.

Contents

1 Introduction

In recent years, podcasts have emerged as a prevalent and influential medium for sharing information, stories, and ideas, and their rapid growth is a testament to their ability to engage audiences through long-form audio content (Berry, 2016). As a result, this medium has become an essential platform for content creators, businesses, and advertisers alike, seeking to reach an increasingly diverse and engaged audience (Sullivan, 2019). With the plethora of podcasts available, there is a growing need for effective tools and techniques to analyse, understand, and capitalize on the vast aggregation of data generated by podcasts and other types of digital media.

One critical aspect of understanding the podcast industry is the importance of platform theory, which explains how digital platforms operate and reshape industries (Parker et al., 2016). The basic notion of a platform is to facilitate the exchange of value between participants of a market (Hagiu & Wright, 2015; Van Alstyne et al., 2016). When participants of a market are connected, the nature of competition between platforms takes on properties of network effects and winner-takes-all dynamics (Katz & Shapiro, 1985; Eisenmann et. al, 2006). To harness these properties, platforms leverage vast amounts of data, algorithms and data science which begets a thorough understanding of their participants' needs and directs the development of the platform's value proposition (Parker et al., 2016). Platforms have revolutionized various sectors, including media (Netflix), retail (eBay), and transportation (Uber), by providing a centralized infrastructure that facilitates transactions between producers and consumers. In the podcast domain, platforms like Spotify and Apple Podcasts have become indispensable, offering a wide range of services that empower podcast creators and enable storage, discovery, distribution, and content monetization (Sullivan, 2019).

As platforms play an increasingly vital role in the digital landscape, their reliance on algorithms and artificial intelligence (AI) has become a key factor to remain competitive (Iansiti & Lakhani, 2020). By employing advanced data analysis and machine learning (ML) techniques, these platforms can better understand user preferences, enhance content discovery, and provide personalized recommendations at scale (Sullivan, 2019). The implementation of AI-driven solutions not only enhances the user experience and the competitive advantage of the platforms but also offers valuable insights for content creators, enabling them to optimize their content and reach a broader audience.

Strong value propositions are essential for platforms to attract and retain users, thereby fostering network effects that drive the platform's success and growth (Parker et al., 2016). By developing compelling monetization strategies, platforms can amplify these network effects, leading to a self-reinforcing cycle of increased user engagement and revenue gen-

eration (van Alstyne et. al., 2016). With the growing popularity of podcasts, advertising has become an important source of revenue for both creators and platforms, estimated to stand for over 1.33 billion USD yearly (Whitner, 2023). The effectiveness of advertising relies heavily on the ability to target the right audience and provide relevant content, as evidenced by 69 percent of podcast listeners agreeing that podcast ads make them aware of new products and services (Whitner, 2023; Lambrecht & Tucker, 2013).

Currently, podcast advertisements are sourced from companies that have either contacted the creators or the other way around and are inserted at the point of production. Two popular approaches to podcast advertising are by inserting a pre-recorded ad or having the creators talk about the product in a dedicated section of the podcast (Bezbaruah & Brahmbhatt, 2023). This approach requires the creators themselves to source advertisers and identify suitable placements of the ads in the content.

By leveraging AI and advanced analytics, platforms can better understand user behaviour, preferences, and content, enabling more precise targeting and improved delivery of advertisements (Provost & Fawcett, 2013). An example of a podcast platform leveraging such technologies is Spotify which is investing in its podcast advertising capabilities (Spotify, 2022). By using the wealth of the user and consumption data available to them they aim to make insights, personalization, and scale feasible for advertisers and content creators by introducing Dynamic Ad Insertion (DAI) which leverages the company's full digital suite of planning, reporting and measurement capabilities (Leung, 2020). DAI aim to increase advertisers' ability to target audiences with precision at scale and provide more understanding of the efficiency and impact of ad placement and delivery by streaming different ads to listeners of the same podcast based on personal user data (Spotify, 2022). However, this approach requires large amounts of behavioural user data and is therefore not a pursuable strategy for new platforms as they have not been able to collect such data yet. It is also a strategy which cannot be applied to new users as the platform has not gathered enough data on those users yet (Géron, 2019).

The advent of transformers and large language models has revolutionized the field of natural language processing (NLP) and computational linguistics, offering unprecedented capabilities in understanding and generating human-like language. These models, such as BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018), GPT (Generative Pre-trained Transformer) (Radford et al., 2018), and their subsequent iterations, have demonstrated remarkable success in various NLP tasks, including sentiment analysis, machine translation, and named entity recognition.

These models have the potential to significantly impact various aspects of the podcast ecosystem, including advertisement placement, targeting, and content discovery. By leveraging the power of transformer-based models, this study will show it is possible to segment podcast transcripts based on content, effectively pinpointing ideal advertisement spots within the primary content. In addition to advertisement placement, transformer-based models offer the potential for improved topic modelling of podcasts. Topic modelling is an ML technique that aims to uncover the underlying themes and topical patterns within a collection of documents (Blei et. al., 2003). By applying models like BERT, we aim to extract accurate and granular topics from the podcast content, enabling a deeper understanding of the content which can be leveraged for advertisement.

The combination of topic modelling and text segmentation can contribute significantly to podcast monetization strategies and remove the need of using users' behavioural data. By accurately identifying possible advertisement spots and understanding the content of different parts of podcast episodes, creators, advertisers, and platform providers can allocate targeted advertisements that are related to the content at scale and without the use of personal user data. Platforms adding such capabilities would strengthen their revenue opportunities and value proposition towards creators and advertisers. This is the primary business application for the proposed methodology of this study.

In this thesis, we explore the application of transformer-based topic modelling and text segmentation techniques to develop such capabilities. As input data, we use podcast transcripts gathered from the Spotify Podcast Dataset, the largest corpus of transcribed speech data within the domain (Clifton et. al., 2020). Specifically, we propose a modelling framework where we leverage a BERT-based method for topic modelling and develop a novel approach for segmenting transcripts based on topic shifts to identify advertisement spots. Iterating and experimenting with various configurations and methods, we strive to demonstrate the effectiveness and potential of the method in addressing the challenges associated with analysing podcast data. Our objective is to further the advancement of useful tools for podcast platforms and advertisers while establishing a foundation for subsequent research in this domain.

The assumption that advertisements are most effective when placed at topic shifts is unexplored but partly supported by the importance of context, consumer attention, and engagement in advertising (van Reijmersdal et al., 2009). Ritter and Cho (2009) found that advertisement placement in the middle of the primary content leads to a higher sense of intrusiveness and irritation in the listener. Integrating ads relevant to the ongoing discussion at topical transitions instead may lead to greater attentiveness and receptiveness among audiences, as the promotional content becomes relevant and seamlessly aligns with the natural flow of the ongoing discussion.

1.1 Research Questions

The following research questions are designed to guide our exploration of how to improve podcast monetization strategies by leveraging modern data science methodologies.

The overarching research question guiding this study is:

> *How can data science be leveraged to place relevant advertisements in podcasts?*

Considering the broad scope of the main research question, it is essential to delineate the objectives that a data science-driven framework must accomplish to place relevant advertisements in podcasts. Considering the idea that advertisement placements are ideally located at topical shifts, such a framework must be able to detect such shifts. Additionally, the framework must identify the topics discussed to enable relevant advertisements to be inserted. To explore these aspects, two sub-questions have been defined to guide this study.

> *RQ 1: How can data science methods locate topical shifts in podcast transcripts?*
> *RQ 2: How can data science methods find topics that are meaningful for podcast advertisement?*

1.2 Thesis Structure

This section presents an overview of the thesis structure and the objectives for each section, aiming to guide the research process and facilitate the reader's understanding.

In the *Introduction*, we have presented our thesis's motivation and research objective and provided a brief overview of the methodology we employ for podcast transcript analysis and segmentation.

In the *Literature Review*, we review state-of-the-art methodologies, tools, and best practices in topic modelling and text segmentation, situating our approach within the broader research context. Additionally, we explore business-oriented perspectives relevant to our approach, such as digital media, big data, AI, platform economy, network effects, and monetization strategies in the podcast ecosystem.

The *Theory* chapter introduces and discusses the ML and NLP concepts and models used in this study.

In the *Methodology* chapter, we present our study's research design and approach, including philosophy and strategy. We also give a technical explanation of our novel approach to text segmentation and outline the details of the algorithms used.

In the *Results* chapter, we present the performance of our methodology, including evaluation metrics as well as examples of output from the methodology when applied to

individual podcast transcripts.

In the *Discussion* chapter, we answer the research questions and delve into the interpretation and implications of the results, comparing them to existing research, and identifying limitations. The chapter also provides a reference for future research in the area.

The *Conclusion* provides a summary of the thesis research objectives and highlights the main findings from our study.

2 Literature Review

This section will first go through the business-oriented perspectives relevant to this study, including the platform theory and network effects, organisation challenges related to AI, monetization and advertisement strategies on platforms and the dynamics of podcasts. This broader context will help explain the implications and opportunities of leveraging data science methods in the podcast ecosystem and motivate the business framework. Secondly, this section will detail techniques, applications, and related work on topic modelling and text segmentation, and present the research gap to which this study aims to contribute to.

2.1 Digital Media: Platforms, AI, and Monetization Strategies

2.1.1 Multi-Sided Platforms and Network Effects in the Digital Age

The rise of digital platforms has transformed the way businesses and consumers interact, driving the growth of multi-sided platforms (MSPs) that enable various user groups to exchange value and engage in transactions (Rochet & Tirole, 2003; Hagiu, 2009). In the context of digital media, these platforms often operate as intermediaries between content creators, advertisers, and consumers, fostering a complex ecosystem characterized by network effects and interdependencies (Parker et al., 2016; Evans & Schmalensee, 2016).

Historically, MSPs have played a critical role in traditional media industries, such as newspapers, television, and radio, where they facilitated the distribution of content and advertising (Hagiu, 2009). However, the advent of the internet and digital technologies has amplified the importance of MSPs and network effects, enabling the emergence of new platforms like social media, streaming services, and media platforms (Parker et al., 2016; Evans & Schmalensee, 2010). These digital platforms have reshaped the media landscape, offering new opportunities for content creation, consumption, and monetization, while also presenting challenges related to competition, regulation, and user privacy (Hagiu & Wright, 2015).

Network effects occur when the value of a product or service increases with the number of users (Katz & Shapiro, 1985). Bresnahan (1998) explains that as the user base of a platform grows, the users find new ways of leveraging the capabilities of the platform. If platform owners are responsive to the new user dynamics, new capabilities can be built which adhere to those, further user growth can be generated as a result of the

platform increasing its value proposition. This is called creating a virtuous cycle that can lead to rapid growth and market dominance (Katz & Shapiro, 1985; Van Alstyne et al., 2016, Bresnahan, 1998). The virtuous cycle of positive network effects can reinforce the value proposition of MSPs by increasing the level of user engagement on the platform and acting upon it (Caillaud & Jullien, 2003). Furthermore, network effects can enable MSPs to establish powerful competitive advantages, making it difficult for rivals to enter the market and challenge their position (Eisenmann et al., 2006). These winner-takes-all dynamics, a characteristic of network effects, leads to a scenario in which a single dominant player can capture the majority of the market share (Eisenmann, et. al, 2006; McIntyre & Chintakananda, 2014).

The study of MSPs has attracted significant attention from researchers and practitioners alike, with various theoretical and empirical contributions shedding light on the drivers, dynamics, and implications of these phenomena (Parker et al., 2016; Hagiu & Wright, 2015). Key platform ecosystem dynamics involve factors such as openness, multi-homing, and cross-side complementarities (Parker et al., 2016). Boudreau (2010) define open platforms as platforms allowing third-party developers to create and contribute applications, one of the more prominent examples being Apple and their App Store. He investigated the relationship between platform openness and innovation, finding that open platforms typically attract a larger and more diverse set of applications, thereby enhancing network effects and user value. Cennamo and Santalo (2013) examined the competitive dynamics of digital platforms, emphasizing the roles of multi-homing and cross-side complementarities in determining market outcomes. Multi-homing occurs when users or providers engage with multiple platforms simultaneously, while cross-side complementarities arise when the value of one side of the platform (e.g., consumers) increases with the growth of the other side (e.g., service providers). These dynamics impact the competitive landscape and the capacity of platforms to attract and retain users, ultimately shaping the success of digital platforms in the market (Cennamo & Santalo, 2013).

A key aspect of digital MSPs is their reliance on data and algorithms to deliver personalized content and targeted advertising, driving user engagement and revenue generation (Goldfarb & Tucker, 2011). As digital platforms continue to grow and amass vast amounts of user data, they can leverage advanced analytics and ML techniques to extract value from the data to optimize their offerings (Provost & Fawcett, 2013). In the context of podcast platforms, this includes the use of NLP and information retrieval techniques to analyse podcast transcripts and inform content recommendations, advertising strategies, and platform design (Clifton et. al., 2020).

2.1.2 The Rise of Big Data & AI

The digital media landscape has experienced a massive increase in the volume of data, offering numerous opportunities for businesses to capitalize on insights, improve decision-making, and stay competitive (Iansiti & Lakhani, 2020; McAfee & Brynjolfsson, 2012). This section explores the business implications of the big data and AI revolution, focusing on revenue opportunities and the need for companies to reshape their business models.

The big data revolution is transforming management and decision-making in businesses, emphasizing the need for organizations to adapt their business models and strategies to

remain competitive in the data-driven economy (McAfee and Brynjolfsson, 2012; Davenport, Barth & Bean, 2012; LaValle et. al.,2011). Companies that have successfully embraced data-driven decision-making have benefitted from improved operational efficiency, better customer targeting, and increased profitability (McAfee and Brynjolfsson, 2012). Manyika et al. (2011) further underscore the impact of big data on various sectors of the economy and identify five ways in which big data can create value for organizations: improving transparency, enabling experimentation, segmenting customers, replacing human decision-making with automated algorithms, and fostering data-driven innovations.

As discussed by Iansiti and Lakhani (2020) is the emergence of "super-platforms," which are organizations that leverage AI and network effects to create powerful ecosystems spanning multiple industries. These super-platforms, such as Amazon, Alibaba, and Google, exploit AI to optimize their operations, expand their reach, and increase their dominance in various sectors. By integrating AI with their platform-based business models, these companies gain a significant competitive advantage, driving innovation and growth in the platform economy. The authors suggest that to effectively compete with or coexist alongside these super-platforms, organizations must develop new strategies that exploit AI-driven opportunities and build synergies within the broader ecosystem (Iansiti & Lakhani, 2020).

Iansiti and Lakhani (2020) introduce the concept of the "AI Factory," which represents the new operating model for businesses that want to exploit AI-driven opportunities. This model comprises four key components: data pipeline, algorithms, modular architecture, and cloud-based infrastructure. It's argued that the convergence of these components has accelerated the development and adoption of AI tools, reshaping industries, economies, and societies. To remain competitive in this rapidly evolving environment, the authors underscore the importance of cultivating new leadership styles that prioritize innovation, agility, and "digital intuition"—an understanding of how AI and digital technologies influence business processes and create additional value.

The authors also delve into the complex interplay between AI and network effects in shaping the digital platform economy as network effects play a crucial role in augmenting the impact of AI-driven innovations. By harnessing the wealth of data generated through user interactions, AI systems can refine their algorithms, improve personalization, and offer enhanced user experiences. As more users join the platform, the AI's performance and the platform's value proposition strengthen, attracting an even larger user base and reinforcing the network effects. The authors stress that AI not only benefits from network effects but also accelerates them, resulting in a potent feedback loop that bolsters the platform's competitive advantage. This reciprocal relationship between AI and network effects serves as a defining characteristic of the digital platform economy, influencing competitive dynamics and propelling the rapid evolution of industries and markets (Iansiti & Lakhani, 2020).

The emergence of the platform economy, as discussed by Parker, Van Alstyne, and Choudary (2016), further underscores the importance of data-driven insights and network effects in shaping business strategies and models. Platforms excel by harnessing data, algorithms, and network effects to deliver value to users and simultaneously transform traditional industries. As companies navigate the digital media landscape and the platform economy, it is crucial to merge big data insights with an understanding of strate-

gic principles within a platform economy.

2.1.3 Monetization and Advertising Strategies in Digital Media

The platformisation and digital transformation of the media industry have spurred a shift in monetization and advertising strategies, with an increasing focus on data-driven targeting and personalization to maximize user engagement and revenue (Goldfarb & Tucker, 2011; Sullivan, 2019). As traditional advertising models struggle to adapt to the fragmentation and proliferation of digital content, new approaches have emerged that leverage the vast amounts of user data, advanced analytics capabilities, and AI available to digital platforms (Provost & Fawcett, 2013; Lambrecht & Tucker, 2013).

As digital platforms have become increasingly important in the global economy, businesses have been compelled to adapt their strategies to remain competitive and relevant in the digital age. This has led to the emergence of various monetization models, such as advertising, subscription, and transaction-based models, which enable platforms to generate revenue by facilitating the exchange of value between different user groups (Hagiu & Wright, 2015; Van Alstyne et al., 2016). In the case of digital media platforms, advertising plays a particularly important role in the monetization process, as it allows content creators to generate revenue while offering their content for free or at a lower cost to consumers (Goldfarb & Tucker, 2011).

Historically, advertising in the media industry has been characterized by mass-market campaigns and broad targeting, with limited opportunities for personalization and measurement (Napoli, 2011). However, the rise of digital platforms and the increasing availability of granular user data has enabled the development of more sophisticated advertising models, such as programmatic advertising, real-time bidding, and native advertising (Lambrecht & Tucker, 2013; Athey et al., 2018). These models seek to optimize the delivery and effectiveness of ads by aligning them with user preferences, behaviours, and contexts, often using ML and predictive analytics (Goldfarb & Tucker, 2011; Provost & Fawcett, 2013).

Native advertising has emerged as a significant trend in the online advertising space, with Campbell and Marks (2015) examining its impact on user engagement. The authors found that native ads can be more effective than traditional display ads in specific contexts. Native advertisements are designed to blend seamlessly with the surrounding content, thereby minimizing disruption to the user's online experience (Campbell & Marks, 2015). This approach aims to enhance user engagement by delivering ads that are contextually relevant, visually consistent with the platform, and less intrusive compared to traditional display ads. As a result, native advertising has the potential to improve ad effectiveness, generate higher click-through rates, and foster better brand recall, ultimately offering a more appealing advertising solution for both brands and consumers.

In the context of podcast platforms, the growing popularity and diversity of podcast content present unique opportunities and challenges for monetization and advertising. While traditional sponsorship and pre-roll ad formats remain prevalent, there is increasing interest in exploring more dynamic and data-driven advertising strategies that can leverage podcast transcripts and other unstructured data sources to inform targeting, personal-

ization, and ad placement. For example, podcast platforms such as Spotify, have begun experimenting with DAI, which enables the delivery of targeted ads based on user demographics, listening history, and other contextual factors in streaming media (Leung, 2020).

2.1.4 The History of Podcasts & Podcast Advertising

The podcasting landscape has witnessed rapid growth and diversification over the past two decades, evolving into a prominent digital mass medium that has reshaped the way we consume and interact with audio content (Bonini Baldini, 2015). Podcasts started to emerge in the early 2000s, when the fusion of syndicated digital audio files and portable media players, such as Apple's iPod, gave rise to the term "podcasting" (Berry, 2006). Since then, the medium has experienced a resurgence, often referred to as the "second age" of podcasting, characterized by increased production, consumption, and monetization (Bonini Baldini, 2015). The growth in podcast consumption can be attributed to several factors, including increasing access to smartphones, improved internet connectivity, and the rise of streaming services (McHugh, 2016). Additionally, the diverse range of podcast genres, from news and storytelling to educational content and comedy, caters to a wide array of listener interests, further driving the medium's popularity (Bonini Baldini, 2015).

This rise of podcasts has coincided with a shift towards platformisation, as multi-sided platforms leverage network effects to facilitate interactions between podcast creators, advertisers, and consumers (Van Dijck et al., 2018). Podcast platforms, such as Spotify and Apple, play a critical role in content discovery, distribution, and storage, shaping the overall podcast ecosystem (Sullivan, 2019). One key aspect of platformisation is the monetization of podcasts through advertising. Podcast advertisements have emerged as an effective marketing channel, with brands capitalizing on the medium's highly engaged and loyal listener base. Advertisers are increasingly adopting new strategies, leveraging the accumulated data and network effects on platforms to allow for increased discovery and personalized advertisement (Sullivan, 2019).

The traditional model of podcast advertising typically involves the insertion of pre-recorded ads, known as "baked-in" ads, directly into the podcast content (Bezbaruah & Brahmbhatt, 2023). While this approach can create a more natural listening experience, it lacks the flexibility and targeting capabilities offered by Dynamic Ad Insertion (DAI). DAI has emerged as a disruptive technology in the podcast advertising landscape, enabling the delivery of targeted ads based on user demographics, user behaviour, and other contextual factors. This innovative approach to advertising has transformed the way brands engage with their audiences in the podcasting space, offering more personalized and relevant ad experiences (Leung, 2020).

DAI enables real-time stitching of ads into podcast episodes, replacing or supplementing baked-in ads with dynamically generated content tailored to individual listeners. DAI technology relies on advanced algorithms that analyse listener data, such as geographic location, device type, and content preferences, to deliver targeted ads with higher relevance and engagement potential (Leung, 2020). This data-driven approach has been shown to improve ad performance, leading to higher click-through rates and increased return on investment for advertisers. The adoption of DAI has also opened new revenue

streams for podcast creators and platforms, enabling them to monetize their back catalogues by inserting fresh and relevant ads into older episodes (Leung, 2020). This method requires extensive historical and current user behavior data, which may not be accessible to new platforms. Moreover, targeting new users becomes challenging as the platform has not yet amassed sufficient data on them (Géron, 2019). An unexplored avenue of DAI involves using it for native advertisement placement. This approach reduces entry barriers, as it doesn't require extensive user data but still ensures high user engagement with ads (Campbell & Marks, 2015). By focusing on podcast content instead of user data, platforms can facilitate targeted advertising while preserving user privacy and capitalizing on network effects.

2.2 Related Work

In this section, related work within topic modelling and text segmentation will be presented where techniques, methods and applications will be detailed for each component.

2.2.1 Topic Modelling

Topic modelling was long performed with techniques based on probabilistic topic modelling such as Latent Semantic Analysis (LSA) which was first presented by (Deerwester et al., 1990). LSA was presented to improve information retrieval which describes the task of matching a search query with the best matching document of a corpus. LSA constructs a term frequency matrix which is then reduced in dimensionality using singular value decomposition (SVD). SVD factorise the original matrix into three new matrices, $M = U\Sigma V^T$, from which the documents can be interpreted as embedded in topic space in the U matrix. The S matrix is a diagonal matrix containing the eigenvalues that rank the topics' importance. The V matrix when multiplied with Σ gives the describing words for each topic. LSAs largest contribution to the field was by introducing an efficient dimensionality reduction technique which combated earlier problems with the curse of dimensionality (heuristically explained as a $m \times n$ feature matrix where $m < n$) of the preceding uni-gram models.

Latent Dirichlet Allocation (LDA) was later presented in 2003 in the field of ML as a way of modelling collections of discrete data such as collections of texts (Blei et al., 2003). This approach assumes that each document is modelled as a mixture of a finite set of underlying topics. Each topic is in turn modelled as an infinite set of multinomial word probabilities conditioned on the topic. From that, each document is interpreted as probabilistically generated from the underlying topics. LDA is an unsupervised technique, and the topics are found by finding the distribution of topics that maximise the likelihood of the documents in the corpus.

Both LSA and LDA employ the bag-of-words (BoW) technique for text representation (Deerwester et al., 1990; Blei et al., 2003). BoW simplifies semantics by assigning a document's topic based on word counts within the document and its relationship to other documents in the corpus. However, the technique does not account for entities described

by multiple tokens, despite efforts to construct multi-gram lexicons for BoW methods (Mikolov et al., 2013). Furthermore, BoW disregards semantic and syntactic relationships between words in a sentence and overlooks polysemy, a concept where a word can have multiple meanings depending on the context (Pustejovsky, 1995).

Non-negative Matrix Factorization (NMF) has emerged as another popular technique for topic modelling, offering an alternative to LSA and LDA. Introduced by Lee and Seung (1999), NMF is a dimensionality reduction method that decomposes a non-negative data matrix into two lower-dimensional non-negative matrices, effectively capturing the underlying structure of the data in a more interpretable manner. In the context of topic modelling, NMF operates on a term-document matrix, factorizing it into a term-topic matrix and a topic-document matrix, which represent the term weights for each topic and the topic weights for each document, respectively. NMF's non-negativity constraint leads to a parts-based representation, resulting in more interpretable topics compared to LSA, which can produce negative weights for terms and topics. This non-negativity property aligns with the inherent nature of text data, as word frequencies and topic proportions cannot be negative. Additionally, NMF is more flexible in handling polysemy, as it allows terms to have different weights in multiple topics, thus better capturing the context-dependent meanings of words. Like LSA and LDA, NMF also relies on the BoW technique.

Despite its advantages, NMF is computationally more demanding compared to LSA, as it employs iterative optimization algorithms to find the best factorization (Lee & Seung, 1999). Moreover, NMF does not inherently capture the generative process of documents like LDA, which models each document as a mixture of underlying topics (Blei et al., 2003). Nonetheless, NMF has been successfully applied in various topic modelling applications, demonstrating its effectiveness in capturing meaningful and interpretable topics from large-scale text corpora (Berry et al., 2007).

Building on traditional topic modelling techniques such as LDA, LSA, and NMF, recent research has focused on addressing the limitations of these approaches, particularly regarding capturing semantic relationships in text and not relying on a BoW representation. Novel approaches have been proposed that leverage text embeddings and clustering algorithms to generate more accurate and semantically meaningful topics (Sia et al., 2020; Grootendorst, 2022). Sia et al. (2020) explored the use of pre-trained embeddings, such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2018), in combination with clustering algorithms to improve topic modelling. By representing text with dense vector embeddings, they were able to capture semantic relationships between words and alleviate the limitations of BoW-based methods. This approach demonstrated the potential of using embeddings for topic modelling and paved the way for further advancements in the field.

One such advancement is BERTopic, a topic modelling technique proposed by Grootendorst (2022) that leverages BERT embeddings and clustering algorithms to generate topics. BERTopic separates the topic representation process from the clustering process, giving it a modular nature that allows the approach to improve as embedding techniques and pre-trained Large Language Models (LLM) continue to advance. In BERTopic, documents are first converted into dense vector representations using BERT or other transformer-based language models, effectively capturing the semantic information in the text. Next,

the dimensionality of the embeddings is reduced using UMAP, preserving the local and global structure of the data. Finally, a clustering algorithm, such as HDBSCAN is used to group the embeddings into clusters that represent the topics. When creating topic representations, the author presents a modified, class-based TF-IDF method which allows for the generation of topic-word distributions for each cluster of documents.

BERTopic and similar approaches offer several advantages over traditional topic modelling techniques. By utilizing dense vector embeddings and advanced clustering algorithms, they can better capture semantic relationships, handle polysemy, and work effectively with shorter texts. Moreover, these approaches can be easily fine-tuned or extended by using other transformer-based language models to adapt to specific domains or languages. As a result, BERTopic and related methods represent a promising direction for future research and development in topic modelling.

2.2.2 Text Segmentation

Text segmentation is a critical step in many NLP tasks, as it allows for the partitioning of documents or corpora into meaningful units such as sentences, paragraphs, or topical sections. Over the years, numerous techniques have been developed to tackle the challenges associated with text segmentation, ranging from rule-based methods, and sliding window approaches to ML-based algorithms and deep learning architectures (Eisenstein & Barzilay, 2008; Riedl & Biemann, 2012; Arnold et al., 2019). In this section, we explore the evolution of text segmentation techniques and their application in various domains, focusing on their relevance and potential for the purposes of this study.

One of the earliest text segmentation techniques is the rule-based approach, which relies on heuristics and linguistic patterns to identify segment boundaries (Hearst, 1994). For instance, Hearst (1994) introduced TextTiling, an algorithm that segments texts into multi-paragraph units based on lexical cohesion. By calculating the similarity between BoW vectors of adjacent blocks of text using a sliding window, TextTiling identifies boundaries based on the cosine similarity. While rule-based methods like TextTiling have demonstrated effectiveness in some cases, they often struggle to adapt to varying linguistic styles and domains.

TopicTiling, a method for unsupervised topic segmentation, was proposed by Riedl and Biemann (2012) to address some of the shortcomings observed with TextTiling. The approach is a modification of the well-known TextTiling method and is based on a distributional hypothesis and utilizes the topic assignments from LDA to determine topic boundaries within the text. In TopicTiling, the text is divided into equal-sized blocks or tiles. The algorithm computes the similarity between adjacent blocks of text by calculating a coherence score, and topic boundaries are identified by detecting local minima in the scores. This method has been successfully applied to various conversational data types, such as telephone conversations, interviews, and meetings, demonstrating its potential for analysing complex and unstructured text (Rield & Biemann, 2012).

A more sophisticated approach to text segmentation involves the use of more advanced statistical models, such as Hidden Markov Models (HMMs) and Bayesian models. For example, Beeferman et al. (1999) proposed a text segmentation method based on an unsu-

pervised HMM, which models the relationships between words and the underlying topics. Similarly, Eisenstein and Barzilay (2008) developed a Bayesian unsupervised model that captures both local and global topic shifts, outperforming existing methods at that time on various datasets. Although ML-based techniques have shown promise in addressing the challenges of text segmentation, they may still require substantial computational resources and can be sensitive to the choice of features and model parameters (Riedl & Biemann, 2012).

Deep learning has demonstrated significant potential in text segmentation, enabling the detection of complex patterns and representations from large data sources (LeCun et al., 2015). Various deep learning architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) along with their advanced variants, including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), for modelling the sequential nature of text and capturing long-range dependencies (Hochreiter & Schmidhuber, 1997; Cho et al., 2014), have made significant strides in addressing text and topic segmentation challenges.

Building upon these advancements, Arnold et al. (2019) introduced SECTOR, a specialized neural model for coherent topic segmentation and classification. This model addresses the issue of coherence in topic segmentation by jointly modelling segmentation and classification tasks using a hierarchical attention mechanism. By effectively leveraging both local and global contexts to learn latent topic representations and identify segment boundaries, SECTOR outperforms previous state-of-the-art methods in topic segmentation and classification tasks. However, as the model was developed to support machine reading systems, its application and performance on conversational data is unknown (Arnold et al., 2019).

The segmentation of conversational data presents unique challenges due to its unstructured nature, the presence of multiple speakers, and the varying speaking styles and language usage (Eisenstein & Barzilay, 2008). Several studies have explored the segmentation of conversational data, such as telephone conversations, meetings, and interviews (Galley et al., 2003; Riedl & Biemann, 2012). For example, Galley et al. (2003) introduced LcSeg, a Bayesian model designed for unsupervised topic segmentation of conversational data. This model considers speaker turns and other conversational cues, enabling effective detection of topic shifts in dynamic and interactive settings, such as podcasts or interviews.

Text segmentation techniques have seen significant evolution over the years, with recent progress in deep learning and transformer-based models offering new possibilities for addressing challenges associated with more complex data. Although existing methods have shown success across various domains and settings, their applicability in this study is an area warranting further exploration. Harnessing the capabilities of transformer-based models like BERT and GPT could pave the way for more effective and efficient approaches to text segmentation, capable of adapting to the intricacies of the data. Ultimately, this could lead to a deeper understanding of its content and potential applications in content discovery, recommendation, and monetization.

2.3 Research Gap

In this section, we outline the research gap that our study seeks to address. Drawing on insights from the previous sections, we discuss each element of our proposed approach individually and identify the gaps that emerge in the interplay between both components in the context of this study.

The BERTopic framework, formally proposed in 2022, has limited research on its applications, particularly in the context of the type of data used in this study. Existing work has primarily focused on the method's generalizability, such as Arabic topic modelling (Abuzayed & Al-Khalifa, 2021) or its performance on multi-domain short texts (de Groot et al., 2022). Additionally, comparisons between BERTopic and other topic modelling techniques, like LDA, LSA, and NMF, have been conducted (Egger & Yu, 2022), with BERTopic showing promising results in applications for short-text and across domains. However, most prior work has centred on short-text data, encompassing social media posts, open-text comments, or brief news articles (Egger & Yu, 2022; de Groot et al., 2022). This points to a research gap in the potential application of BERTopic for longer documents, documents with varying characteristics, and downstream tasks such as segmentation.

The research gap also extends to the area of text segmentation. While earlier approaches, such as TextTiling (Hearst, 1994) and TopicTiling (Riedl & Biemann, 2012), have been effective in some cases, they often struggle to adapt to varying linguistic styles and domains. This limitation could stem from TopicTiling's foundation in LDA which performs best on written language data. This limitation opens for the exploration of different architectures of TopicTiling where capabilities of transformer-based models like BERT could be integrated. Such models can capture richer semantic information and context compared to BoW-based methods like LDA.

Exploring the integration of topic probabilities generated by transformer-based topic models into the TopicTiling method can potentially overcome some of the under-looked aspects of previous related work. By leveraging contextualized sentence embeddings and improved topic representations, such modification of the TopicTiling method may yield enhanced performance in segmenting text, particularly for the inherent complexities of the podcast medium. This research gap offers an opportunity to advance text segmentation by combining the strengths of transformer-based topic modelling and segmentation techniques in a novel way.

The research gap is further strengthened by its application in a business context detailed in the first section of the literature review (2.1). As podcast platforms such as Spotify are attempting to build out their capabilities within DAI and native advertisement placement, it further underscores the possible practical implications of this study and the associated gap in both academia and industry. This also resonates with the implications of platform theory, such as leveraging algorithms and data to reinforce and harness network effects and create a self-reinforcing cycle of increased engagement and revenue opportunities for platform participants.

3 Theory

In this section, the prevalent concepts and techniques within ML and NLP relevant to the modelling framework in this study will be discussed. There are various methods and application areas within the field which would need a broader introduction, but this section has been limited to only those necessary for the context and understanding of this study.

3.1 Tokenization

Tokenization is a crucial pre-processing step in NLP and text mining, responsible for breaking down raw text data into individual words, called tokens, which can then be used for various analyses such as topic modelling, sentiment analysis, or information retrieval (Manning, 2009). The development of tokenization techniques has been shaped by a variety of factors, including the evolution of computational linguistics, the increasing diversity and complexity of languages and writing systems, and the growing need for more accurate and efficient text analysis tools. A basic understanding of tokenization is a pretext for understanding the inner workings of embeddings and transformers which will be described later in this chapter. Tokenization is used in this study to generate the topic representations which will be described in chapter 4.

Early tokenization approaches often relied on simple rule-based methods, such as splitting text on whitespace characters or using regular expressions to identify word boundaries. While these methods were effective for many tasks and languages with well-defined word boundaries, they struggled to handle more complex cases, such as agglutinative languages, multi-word expressions, and domain-specific jargon (Manning & Schütze, 1999).

To address these challenges, researchers have developed more advanced tokenization techniques, such as unsupervised segmentation algorithms, which leverage statistical patterns in the data to identify likely word boundaries (Goldsmith, 2001). These methods often employ probabilistic models, such as n-gram models or Hidden Markov Models, to capture the distribution of character sequences within a given corpus and iteratively refine their segmentation hypotheses (Chen et al., 1999).

In recent years, deep learning approaches have emerged as a promising direction for tokenization, leveraging the ability of neural networks to learn complex patterns and representations from large-scale data (Goodfellow et al., 2016). For example, sub-word tokenization techniques, such as Byte Pair Encoding (BPE) and WordPiece, have gained

popularity for their ability to handle out-of-vocabulary words and improve the efficiency of text analysis tasks (Sennrich et al., 2016; Wu et al., 2016). These methods operate by iteratively merging the most frequent character pairs to create a fixed-size vocabulary of sub-word units, which can then be used to segment input text into tokens.

The choice of tokenization method can significantly impact the performance of downstream NLP tasks, as it determines the granularity and quality of the input data for subsequent analysis (Eisenstein, 2013). Therefore, it is essential to select a suitable tokenizer that can accurately and efficiently process the language and domain-specific characteristics of the given text data.

3.2 Embeddings

In mathematics, embeddings describe the mapping of one instance of some mathematical structure into another instance. The embedding is given by some structure preserving an injective map such that $f : X \hookrightarrow Y$ (Hocking & Young, 1988). In the context of NLP, X signify a sequence of tokens and Y signifies an n-dimensional dense vector which is used to describe the sequence. The dense vector representations of the documents are subsequently used in downstream ML tasks as input. The goal is to embed the semantic and syntactic meaning of a word, sentence, or document into a dense vector of real numbers which retains the relation to other words, sentences, or documents in the corpus (Mikolov et. al., 2013).

There are various approaches for achieving this, but the desired properties of an embedding model remain the same no matter the approach. These include non-confluence, robustness against lexical ambiguity, demonstration of multifacetedness, reliability and good geometry (Wang, et al., 2019). Non-confluence refers to the property of being able to encode the same word in different representations, such as singular or plural, in a meaningful way such that it differentiates between them. Robustness against lexical ambiguity refers to the problem raised by polysemy. Demonstration of multifacetedness points to the property of considering different properties of the word, such as the phonetic, syntactic, and morphological properties. If the property of a word changes, so should its embedding. Reliability refers to the fact that the model should yield embeddings of the same quality across multiple iterations of training, as the word vectors are initialised randomly. The last property of a good embedding model is that of good geometry. This property points to the need for a good spread in the embedding space such that the space is used efficiently.

Traditional embeddings, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), generate static, context-independent representations, which may lead to limitations in capturing the nuanced meanings of words in different contexts. Contextual embeddings address these limitations by considering the surrounding context of words, phrases, or sentences in a document, rather than just the individual tokens themselves (Peters et al., 2018; Devlin et al., 2018). These embeddings have gained significant attention due to their ability to capture context-dependent semantics, leading to improved performance in a wide range of NLP tasks (Peters et al., 2018; Devlin et al., 2018).

One of the most well-known and widely used contextual embedding models is BERT, proposed by Devlin et al. (2018). BERT is a pre-trained model that captures contextual information from both sides of a given token, resulting in rich contextual representations. BERT has shown to achieve state-of-the-art results in numerous NLP tasks. Using BERT to generate embeddings for clustering tasks has previously been evaluated to outperform certain tokenization techniques across various clustering algorithms (Subakti et al., 2022) as well as providing a lower runtime and computational complexity in some cases (Sia et al., 2020). Another notable contextual embedding model is ELMo, introduced by Peters et al. (2018). ELMo generates contextual embeddings using deep bidirectional representations from an input text, significantly enhancing the performance of downstream NLP applications. However, as ELMo is based in a recurrent neural network architecture, its attention span for generating context is limited due to problems with quadratic computational complexity. These models, along with others such as RoBERTa (Liu et al., 2019), GPT-2 (Radford et al., 2019), and GPT-3 (Brown et al., 2020), have advanced the field by addressing the limitations of previous embedding techniques and introducing novel embedding construction methods.

However, it is important to note that contextual embeddings also come with challenges, such as the increased computational complexity associated with their large-scale models and the need for substantial amounts of training data (Peters et al., 2018; Devlin et al., 2018; Brown et al., 2020). Despite these challenges, contextual embeddings have demonstrated remarkable success in capturing the desired properties of an embedding model, including non-confluence, robustness against lexical ambiguity, demonstration of multifacetedness, reliability, and good geometry (Peters et al., 2018; Devlin et al., 2018; Radford et al., 2018). As a result, contextual embeddings have become an essential component in modern NLP applications, enabling more accurate and context-aware representations of words, sentences, and documents (Devlin et al., 2018).

3.3 Transformers: A Paradigm Shift in NLP

Before the development of transformers, recurrent neural networks (RNNs) and convolutional neural networks (CNNs) were the dominant approaches to processing sequential data, such as text. However, these models were limited in their ability to capture long-range dependencies and scale efficiently (Vaswani et al., 2017). To address these limitations, Vaswani et al. (2017) introduced the transformer architecture, depicted in Figure 3.1, which relies on self-attention mechanisms to model dependencies within a sequence without the need for recurrence or convolutions.

The transformer architecture is characterized by its use of multi-head self-attention, a mechanism that allows the model to weigh the importance of different parts of the input sequence to generate contextualized representations (Vasani et al., 2017). This self-attention mechanism is achieved by calculating a series of dot products between query, key, and value vectors, which are derived from the input embeddings. The resulting attention scores are then used to create a weighted sum of the value vectors, producing context-aware representations for each token in the sequence.

Another distinguishing feature of the transformer models is its use of positional encoding,

which allows the model to incorporate information about the relative positions of tokens within a sequence (Vasani et al., 2017). Since the self-attention mechanism is permutation invariant, it cannot inherently capture positional information (Vasani et al., 2017). To address this issue, the authors introduced positional encodings to the input embeddings at the bottom of both the encoder and decoder stacks.

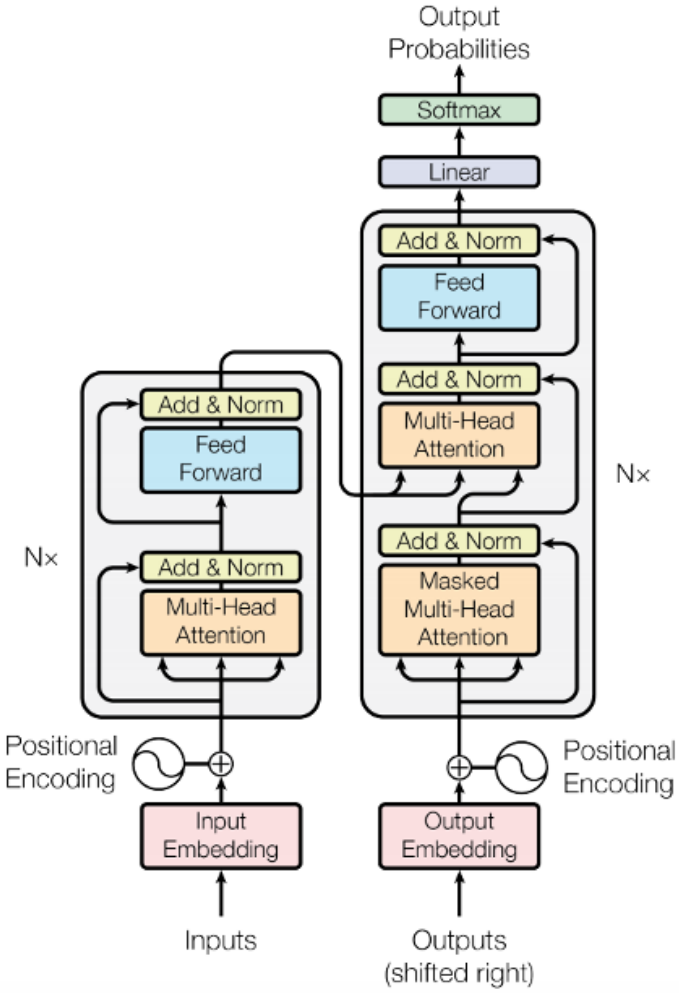

Figure 3.1: *Overview of the Transformer model architecture (Vaswani et al., 2017, Figure 1).*

One of the key advantages of the transformer architecture over previous models is its ability to process input sequences in parallel, as opposed to sequentially. This enables transformers to achieve significantly faster training times and greater scalability compares to RNNs and CNNs, which are inherently sequential in nature (Vaswani et al., 2017). Additionally, the self-attention mechanism allows transformers to effectively model long-range dependencies, overcoming the limitations of RNNs and CNNs in this regard.

The transformer architecture/model has laid the foundation for a new generation of pre-trained language models, such as BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and RoBERTa (Liu et al., 2019) which have achieved state-of-the-art performance across a wide range of NLP tasks. These models leverage the power of transformers to generate contextualized representations that can be fine-tuned for specific tasks, resulting in significantly improved performance compared to traditional feature-based and task-specific models.

3.3.1 Sentence Transformers

The widespread success of transformer-based models has led to the development of specialized architectures designed to generate meaningful embeddings for sentences or short paragraphs. Sentence Transformers, proposed by Reimers and Gurevych (2019), is one such model that builds upon the BERT architecture to generate semantically rich sentence embeddings through a Siamese network approach.

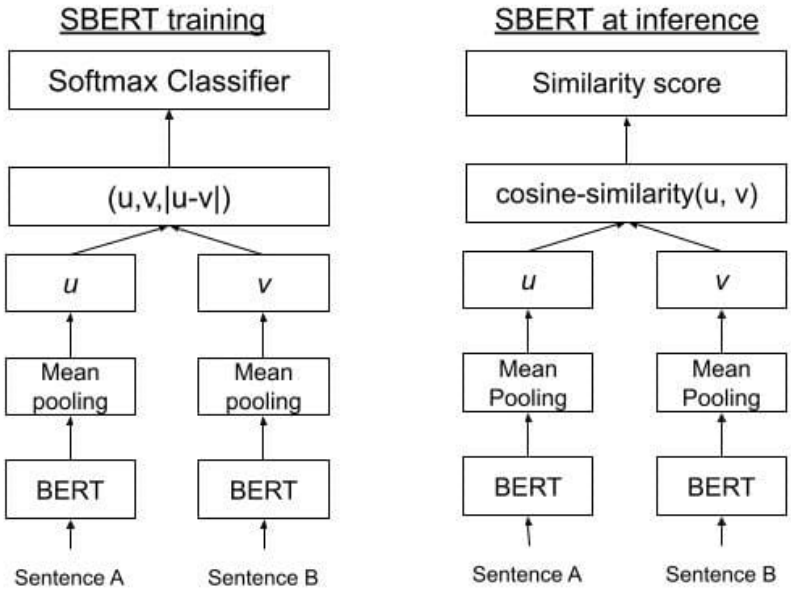

Figure 3.2: *Illustration of the Siamese network architecture in SBERT (Reimers & Gurevych, 2019, Figure 1 & Figure 2).*

The primary motivation behind sentence transformers is the need for an efficient and effective method to generate fixed-size embeddings for sentences (Reimers & Gurevych, 2019). While BERT excels at capturing contextual information within sentences, its architecture is not designed to produce fixed-size embeddings for entire sentences (Reimers & Gurevych, 2019). Instead, it generates embeddings for individual tokens, making it challenging to use BERT embeddings for tasks that require sentence-level representations, such as semantic textual similarity or clustering (Reimers & Gurevych, 2019). To address this limitation, the authors introduced a Siamese network architecture that leverages pre-trained BERT models as seen in Figure 3.2. In a Siamese network architecture, two input sequences are independently processed by a shared BERT model, producing contextualized token embeddings. These embeddings are then aggregated using a pooling operation, such as mean or max pooling, to generate fixed-size vector representations for each sentence. The resulting sentence embeddings are then compared using a similarity measure, such as cosine similarity or a learned distance metric, to determine the semantic relatedness of the input sequences.

A key advantage of the approach is that it enables the generation of sentence embeddings in a computationally efficient matter. By using pre-trained BERT models, a shared network architecture, it can effectively capture the contextual information encoded within sentences while significantly reducing the computational overhead associated with processing individual tokens (Reimers & Gurevych, 2019). The authors demonstrated the

effectiveness of the modified architecture by evaluating it on a range of semantic textual similarity tasks. The results showed that Sentence Transformers outperformed, at the time, state-of-the-art models, such as InferSent and Universal Sentence Encoder, highlighting the power of their Siamese BERT-network approach.

3.3.2 Long-Sequence Transformers

The quadratic complexity of the self-attention mechanism in the original transformer architecture has limited their applicability to long sequences due to the computational overhead and memory requirements (Beltagy et al., 2020; Zaheer et al., 2020). To address this limitation, researchers have proposed long-sequence transformers, such as Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020), designed to handle longer input sequences more efficiently.

Longformer is a transformer architecture that replaces the standard self-attention mechanism with a more efficient variant called "sliding window attention" (Beltagy et al., 2020). This approach allows Longformer to handle sequences of up to 4096 tokens, significantly extending the capabilities of traditional transformers. The sliding window attention mechanism is based on the observation that not all token pairs need to be attended to in long documents. Instead, the model focuses on local context by attending a fixed-size window around each token. Essentially, each token only needs to attend to a subset of the other tokens and with the network being multi-layered it manages to cover the entire sequence after going through all layers.

As an extension of the mechanism, a dilated sliding window attention pattern is also introduced (Beltagy et al., 2020). There, each token's attention positions are separated by d empty spaces which further increases the receptive field without increasing computation. The receptive field refers to the range or scope of input elements (words, tokens, positions) that a model can consider when processing a specific element in the input sequence. Additionally, Longformer incorporates global attention to selected tokens, ensuring that critical information from distant parts of the input sequence is considered. This combination of local and global attention considerably reduces the computational complexity of the model, enabling it to process longer sequences with minimal impact on memory and computational resources (Beltagy et al., 2020). A conceptual view of the different attention mechanisms and how they attend to tokens is seen in Figure 3.3.

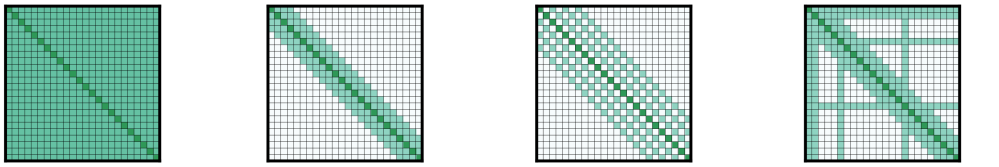

Figure 3.3: *Illustration of the attention mechanism in Longformer (Beltagy et al., 2020, Figure 2). The input sequence has length n, and the matrix represents the size of the matrix of similarity scores of size n^2. The green cells represent the positions with which each token computes a similarity score. Different attention mechanisms generate different fields of attention as shown in the illustration. The three mechanisms to the right have a smaller computational and memory complexity which enables longer sequences to be processed.*

Big Bird is another long-sequence transformer from Google (Zaheer et al., 2020) that addresses the quadratic complexity issue by introducing a spare attention mechanism called "random attention". The random attention mechanism allows Big Bird to attend to a small, fixed number of randomly selected tokens from the input sequence, in addition to attending to the local context through a fixed-size window. The sparsity pattern in the attention matrix enables Big Bird to maintain linear complexity with respect to the sequence length, allowing it as well to process input sequences of up to 4096 tokens (Figure 3.4). Additionally, the random attention mechanism is designed to keep the ability of the traditional transformer architecture to understand and process complex relationships within a sequence of tokens. As such it is complementary to the traditional transformer architecture as it boosts the model's ability to process longer sequences while preserving the core features of the model.

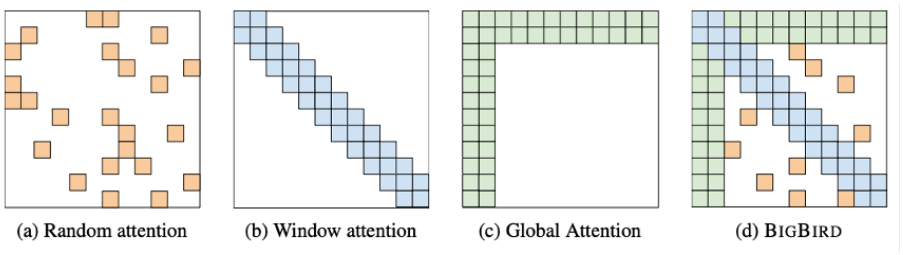

Figure 3.4: *Illustration of the attention patterns in BigBird (Zaheer et al., 2020, Figure 1).*

Both Longformer and Big Bird, have demonstrated their effectiveness in handling long input sequences across a range of NLP tasks, including question-answering, documents summarization, and language modelling (Beltagy et al., 2020; Zaheer et al., 2020). They have outperformed traditional transformers and other long-sequence models, such as Transformer-XL (Dai et al., 2019) and RoBERTa (Liu et al., 2019), on tasks involving long documents or extensive contexts. The architectures represent a meaningful advancement within the field and the introduction of new and efficient attention mechanisms has opened new possibilities for research and development in areas such as document understanding, conversation analysis, and large-scale information extraction (Beltagy et al., 2020; Zaheer et al., 2020).

3.3.3 Large Language Models & GPT

Language model pre-training has become an essential aspect of contemporary NLP, especially with the introduction of large-scale models such as GPT (Radford et al., 2018), and PaLM from Google (Chowdhery et al., 2022). The development of these models has transformed the field, yielding significant performance improvement across various NLP tasks. Especially the GPT family has seen rapid advancements with multiple iterations. GPT-2 expanded upon the original GPT model by increasing the number of parameters and training data size (Radford et al., 2019). This improved performance on a wide range of NLP tasks, including machine translation, summarization, and question-answering, among others. GPT-2's ability to generate coherent and contextually relevant text samples demonstrated the potential of large-scale pre-trained language models.

The subsequent release of GPT-3 pushed the boundaries even further, with a model containing 175 billion parameters, making it the largest transformer model at the time (Brown

et al., 2020). GPT-3 showcased remarkable capabilities in tasks such as few-shot learning, where it could adapt to new tasks with minimal fine-tuning or additional training data. The model's ability to understand context and generate contextually relevant responses provided a significant leap in the field.

Building on the success of GPT-3, researchers have continued to explore the limits and potential applications of large-scale pre-trained language models. The growing interest in these models has led to the development of more efficient training and fine-tuning techniques, as well as the exploration of task-specific architectures and optimization strategies (Brown et al., 2020). Additionally, there has been a focus on understanding the model's inner workings, such as examining the attention mechanisms and the representations learned during pre-training (Brown et al., 2020). This increased understanding has facilitated the development of more effective transfer learning techniques, allowing the models to be more easily adapted to various domains and tasks.

3.4 Dimensionality Reduction

Dimensionality reduction is a critical component in the field of NLP, as it simplifies the representation of high-dimensional data into a lower-dimensional space, preserving the underlying structure and relationships between data points (van der Maaten & Hinton, 2008). Dimensionality reduction techniques can be categorized into linear and non-linear methods. Linear methods, such as Principal Component Analysis (PCA) (Pearson, 1901) and Linear Discriminant Analysis (Fisher, 1936), are among the earliest and most widely used techniques for reducing high-dimensional data. PCA identifies the axes with the highest variance in the data, enabling the transformation of the original data into a lower-dimensional space while preserving most of the original variability by linearly projecting the data from the higher to the lower dimensional space. Linear Discriminant Analysis on the other hand, is a supervised dimensionality reduction technique which aims to maximize the separability between classes in the lower-dimensional space, making it particularly useful for supervised classification tasks.

In the context of NLP, Singular Value Decomposition (SVD) has been a popular linear dimensionality reduction technique, as it can be applied to non-square matrices, such as term-document matrices (Géron, 2019). Latent Semantic Analysis (LSA), an early application of SVD in NLP, demonstrated that SVD could capture latent semantic structure in text data and improve information retrieval performance (Deerwester et al., 1990).

Linear methods have limitations when dealing with complex, non-linear relationships in high-dimensional data. To address these limitations, researchers developed non-linear dimensionality reduction techniques, such as t-Distributed Stochastic Neighbour Embedding (t-SNE) (van der Maaten & Hinton, 2008) and Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018). These techniques can capture an underlying manifold structure in high-dimensional data, providing improved visualizations and representations of complex relationships.

t-SNE, a popular non-linear technique, minimizes the divergence between two probability distributions: one that measures pairwise similarities in the high-dimensional space

and another that measures pairwise similarities in the lower-dimensional space (van der Maaten & Hinton, 2008). This method has been widely adopted in various domains, including NLP, for its ability to produce meaningful visualizations and low dimensional embeddings.

UMAP, a more recent non-linear dimensionality reduction technique, is built on the foundations of Riemannian geometry and algebraic topology (McInnes et al., 2018). UMAP constructs a graph representation of the high-dimensional data and optimizes a low-dimensional representation to preserve the topological structure of the original graph. UMAP has gained popularity due to its superior performance in preserving both local and global structures and its scalability to large datasets.

In the context of topic modelling, dimensionality reduction plays a vital role in simplifying high-dimensional text data representations, such as word embeddings, to facilitate the extraction of meaningful topics and reduce computational complexity. By reducing the dimensionality of text data, topic modelling algorithms can more effectively identify latent patterns and relationships in the underlying semantic space. As seen in Sia et al., (2020) where PCA was used as the dimensionality reduction algorithm or in BERTopic (Grootendorst, 2022) which utilized UMAP, the choice of method is essential for accurate and meaningful topic modelling.

Dimensionality reduction techniques have evolved significantly over the years, from linear methods like PCA and LDA to more advanced non-linear techniques like t-SNE and UMAP. These techniques play a pivotal role in topic modelling, simplifying high-dimensional text data representations and enabling the effective extraction of meaningful topics.

3.5 Clustering

Clustering in NLP plays a central role in various applications, such as document summarization, information retrieval, and topic modelling (Xu & Wunsch, 2005). The primary goal of clustering is to group similar objects together, thereby reducing the complexity of the data and revealing underlying patterns or structures.

One of the earliest clustering algorithms is the K-means algorithm, which was first introduced by MacQueen (1967). K-means is a centroid-based clustering method that partitions data into K clusters by minimizing the within-cluster sum of squared distances. Despite its simplicity and ease of implementation, the K-means algorithm has certain limitations, such as its sensitivity to values set at initialization and the requirement to specify the number of clusters in advance (Xu & Wunsch, 2005).

A widely used alternative to K-means is the hierarchical clustering method, which can be either agglomerative or divisive (Murtagh & Contreras, 2012). Agglomerative hierarchical clustering starts with each data point as a separate cluster and iteratively merges the closest pairs of clusters until only one cluster remains. Divisive hierarchical clustering, on the other hand, starts with a single cluster containing all data points and successively splits the cluster into smaller ones. The major advantage of hierarchical clustering is

that it does not require the number of clusters to be predefined; however, it has a higher computational complexity compared to K-means (Xu & Wunsch, 2005). Additionally, it assumes that each cluster has roughly equal number of observations and that the variance of the distribution of each variable is spherical (Géron, 2019).

In recent years, density-based clustering methods, such as DBSCAN (Ester et al., 1996) and OPTICS (Ankerst et al., 1999), have gained popularity due to their ability to identify clusters of arbitrary shapes and sizes. These algorithms rely on the density of data points to determine cluster boundaries and allow for data to be classified as noise. While density-based clustering methods offer several advantages over traditional techniques, they may be sensitive to the choice of hyperparameters, such as density thresholds and distance metrics (Ester et al., 1996).

In NLP, clustering has been extensively employed for document and text clustering tasks (Steinbach et al., 2000). For instance, Cutting et al. (2017) introduced the Scatter/Gather method, which utilizes a combination of agglomerative hierarchical clustering and K-means to cluster documents based on their term-frequency vectors. Furthermore, clustering techniques have been designed specifically for topic modelling, with Latent Dirichlet Allocation (LDA) being a prominent example (Blei et al., 2003). LDA is a generative probabilistic model that assumes documents are composed of latent topics, which are in turn characterized by a distribution of words. By inferring the topic proportions for each document and the word distributions for each topic, LDA effectively clusters documents based on their latent topics.

Recently, BERTopic (Grootendorst, 2020), a transformer-based topic modelling framework, has emerged as a promising approach to clustering text data. BERTopic combines the power of transformer-based language models, such as BERT, with novel clustering techniques like hierarchical density-based clustering (HDBSCAN) (Campello et al., 2013). The advantage of HDBSCAN over DBSCAN is that the researcher does not need to define the global minimum-difference epsilon (density threshold) hyperparameter that controls what the algorithm should consider as dense areas (Campello et al., 2013). By leveraging the semantic representations learned by transformer models and removing the assumptions of spherical, linearly separable clusters and a need for a global density threshold, the BERTopic framework provides an approach to cluster text data with higher accuracy and granularity compared to conventional methods (Grootendorst, 2020).

4 Methodology

This chapter presents the methodology used to address the research questions and details the techniques utilized for data collection, processing, modelling, and evaluation. The subsequent sections aim to elaborate on the selected research philosophy, approach and design that guide our study and answer the research questions.

4.1 Research Philosophy

Research philosophy is a critical aspect of any research project, as it encompasses the underlying beliefs, assumptions, methodology, and interpretation of findings. It provides a framework for understanding the nature of reality, knowledge, and the methods used to acquire that knowledge (Saunders et al., 2009). In the context of our research questions and thesis project, identifying and discussing the research philosophy is essential to ensure the coherence and validity of the chosen approach.

There are four main research philosophies outlined in Saunders et al. (2019) research onion model: positivism, interpretivism, pragmatism, and realism. Positivism emphasizes the use of objective, quantitative methods to study social phenomena and aims to uncover generalizable laws. In contrast, interpretivism focuses on understanding the subjective meanings and experiences of individuals through qualitative research methods. Realism, on the other hand, assumes that reality exists independently of human perceptions, and researchers seek to uncover this reality with appropriate methods.

In the context of our study, pragmatism emerges as a particularly relevant research philosophy. Pragmatism is centred around the idea that research should be guided by practical concerns and the most effective means to answer the research questions (Saunders et al., 2009). This philosophy allows for the integration of both quantitative and qualitative research methods, enabling researchers to utilize the most suitable techniques for addressing their research objectives. Pragmatism encourages flexibility and adaptability in the research process, recognizing that different approaches may be required to understand different aspects of a phenomenon. Adopting a pragmatic research philosophy enables us to draw on various methods and techniques, ensuring that the chosen approach is well-suited to the unique challenges and opportunities presented by our research questions and the nature of the associated real-world context.

4.2 Research Approach

The research approach provides a roadmap for the implementation of a study, outlining the methods and techniques employed to address the research questions. In line with the pragmatic research philosophy, the study adopts a primarily exploratory research approach, focusing on the practical applications of topic modelling and text segmentation techniques in the context of the chosen dataset. This approach allows for the examination of potential solutions to real-world problems while remaining open to multiple perspectives and methods, in accordance with the principles of pragmatism.

The two research questions that guide this exploratory study are:

> *RQ 1: How can data science methods locate topical shifts in podcast transcripts?*
> *RQ 2: How can data science methods find topics that are meaningful for podcast advertisement?*

To address these questions, the study employs a mixed-methods approach, combining quantitative analysis of topic modelling and text segmentation techniques with qualitative insights into their potential applications for monetization strategies. This strategy enables a comprehensive understanding of the research problem, as well as the identification of practical solutions that can be applied within the domain.

The choice of an exploratory research approach is informed by the nature of the research questions, which seek to uncover new insights and understandings of data science applications in digital media. Exploratory research is particularly suitable for investigating novel or under-explored research areas, as it allows researchers to remain flexible and adaptive in their data collection and analysis (Saunders et al., 2009). Moreover, the pragmatic philosophy underlying the approach emphasizes the importance of generating actionable knowledge that can be leveraged in real-world contexts.

The study begins by employing topic modelling techniques to analyse transcribed podcast transcripts, exploring their potential to uncover meaningful patterns in the data. The quantitative analysis provides valuable insights into the structure and content of the data, as well as the performance of the chosen topic modelling method in capturing these aspects. Subsequently, the research shifts its focus to text segmentation, reconfiguring an existing segmentation method to fit into the study's two-step approach and identify segments based on topic shifts. This process involves a combination of quantitative and qualitative evaluation, as the effectiveness of the segmentation method is assessed both in terms of evaluation metrics and its ability to generate meaningful and coherent segments and topic assignments of the data.

Finally, the study explores the potential applications of the generated segments in the context of real-world applications. This component of the research involves a more qualitative assessment of the findings, examining how the techniques can be leveraged to develop effective advertising strategies for platforms operating in the domain.

In summary, the exploratory research approach adopted in this study allows for a broad

examination of the research questions, incorporating both quantitative and qualitative methods to create a full understanding of the problem area. This approach aligns with the pragmatic research philosophy, emphasising the generation of actionable knowledge and the development of practical solutions to address real-world challenges.

4.3 Research Design

The research design serves as a blueprint for the study, outlining the process through which we address the research questions and accomplish the study's objectives. In the context of our research, we have adopted the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework (Shearer, 2000) to guide the study's design and execution. This choice aligns with the pragmatism research philosophy, as it enables the exploration of techniques and methodologies to derive meaningful insights regarding the application of the chosen methods.

The CRISP-DM framework comprises six iterative phases, which ensure a comprehensive and rigorous approach to data mining while maintaining flexibility to adapt to the specific needs of the project. The first phase, business understanding, involves understanding the problem domain, defining the project's objectives, and identifying relevant stakeholders. This phase entails a thorough examination of the challenges and opportunities associated with the analysis of the subject matter and is described in the literature review chapter. In the data understanding phase, a dataset is collected, and EDA is performed to identify trends, patterns, and potential issues that could impact the analysis. Following this, the data preparation phase includes data cleaning, preprocessing, and transformation to ensure the data is in a suitable format for analysis. The modelling phase involves the development and training of models to identify patterns and relationships in the data. This stage includes the implementation of specific modelling methods to analyse the dataset and generate meaningful insights. After modelling, the evaluation phases assess the performance and validity of the models, ensuring they meet the project's objectives. In this study, the effectiveness of the techniques is evaluated through various performance metrics to confirm the suitability for addressing the research questions. Finally, the deployment phase involves the integration of the developed models into the existing system or process. This phase entails the application of the findings to improve strategies, algorithms, and user experiences. The deployment phase is not within the scope of this study and hence not applicable

4.4 Data Collection

In this section, the data understanding stage of CRISP-DM is outlined. More specifically, we will elaborate on the sources, methods, and considerations involved in acquiring a suitable dataset for our study.

4.4.1 Spotify Podcast Dataset

The dataset used in this study is provided by the music streaming company Spotify and is called The Spotify Podcast Dataset which is the largest dataset of its kind (Clifton et. al., 2020). Other datasets of naturally occurring spoken language are available but are magnitudes of size smaller than this dataset (Canavan et. al., 1997; Du Bois & Englebretson, 2005; Hasebe, 2015). It is comprised of transcriptions of 105,360 podcast episodes uniformly sampled from the Spotify podcast database between the period 1st of January 2019 and the 1st of March 2020. The podcasts are from a wide range of domains, regions and audio quality and have a wide range of durations. Episodes in the podcasts contain content from multi-speaker contexts, interviews, non-speech segments and spontaneous dialogue making its nature quite complex and differ from earlier datasets in that regard as they have been from predominantly structured or scripted contexts with single-speaker content (Clifton et. al., 2020). In total, the dataset contains transcripts of more than 59,000 hours of podcast recordings and more than 603,000,000 words.

Clifton et al. (2020) has applied some filtering when creating the dataset. This filtering includes language, duration, consumption data and speech content ratio. The language element is filtered in two ways. First by selecting only podcasts where the creator had specified in the metadata that the language is English. Second, the authors ran the episode descriptions provided by the creators in a language identification algorithm, filtering out any episode where the language was identified as positively non-English. Since podcasts may contain multilingual content, this still left a selection of 20 language tags for the podcasts in the 100,000 podcasts dataset where eleven represent different variations of English, for example [‘en-US’] and [‘en-IN’] representing English spoken in the USA and in India, respectively. The second part of the filtering is done on the duration element. Here, the authors differentiate between professionally produced and non-professionally produced podcasts. For the professionally produced, representing productions from creators such as Spotify Studios and Gimlet, no restriction on time was set, leaving 125 podcasts exceeding 90 minutes and setting the longest podcast in the selection to 305 minutes. For the non-professionally produced podcasts however, a time limit was set to 90 minutes to limit the amount of noise, which according to the authors is more prevalent in long non-professionally produced podcasts. Third, the authors have used consumption data of the episodes from the first month after their release as a naïve proxy for high-quality content. They concluded that this approach was successful, effectively filtering out defective and noisy episodes. Lastly, they filtered out podcasts containing less than 50 percent of speech content.

Each episode is a part of a show which is run by a creator. Some shows are represented with more than one episode while others are represented by a single episode. There is a total of 18,290 podcast shows in the dataset with the largest show represented by 1,072

episodes and the smallest by one. There are a total of 8,632 shows represented by a single episode and 16,354 shows represented by less than ten episodes.

The episodes vary in length and word count as can be seen in Figure 4.1. The vertical lines describe the cumulative count of episodes up until a threshold. The distribution is skewed towards episodes with low word count and short length. In the right plot, the filtering made on length by the authors becomes evident as the count plummets at 90 minutes. The average word count is 5,732 and the average duration is 31.7 minutes. On average, 169 words are spoken per minute across the dataset.

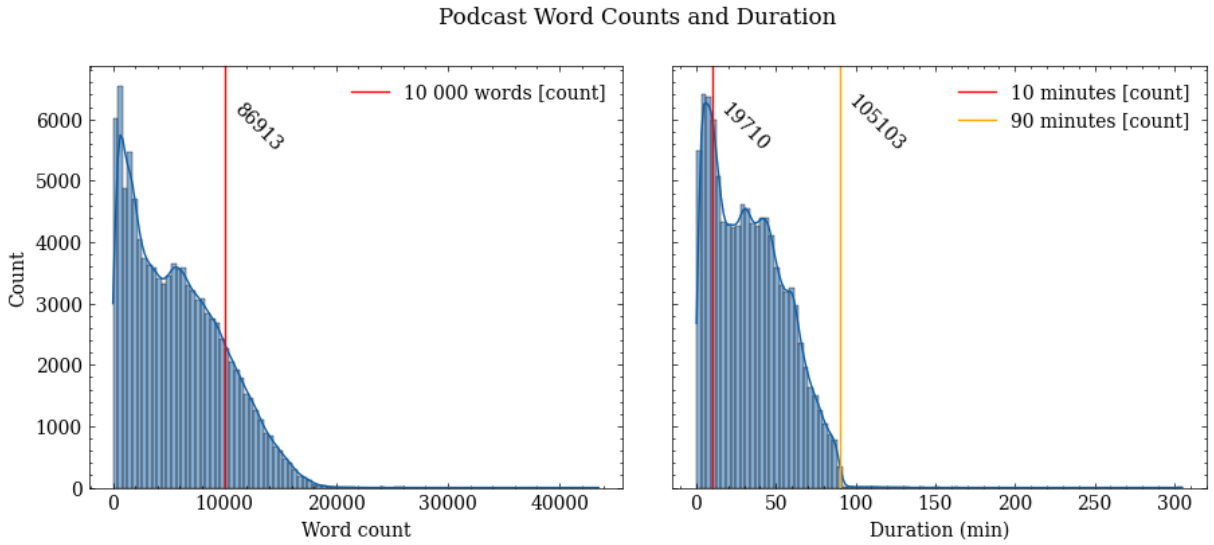

Figure 4.1: *Distribution of word count and duration in minutes.*

In addition to the metadata provided, we scraped the category labels set by the creators using the podcast episode RSS feeds. Clifton et. al. (2020) points to the fact that these labels might be unreliable as they are set in an arbitrary fashion by the creator and might not be the best representation of the content of an individual episode. Additionally, podcast episodes may be tagged with more than one category label. However, we decided that scraping the primary category of each episode should be done to gain an insight into what type of material is predominant in the dataset and what a downstream topic modelling task may output. In total, 129 categories were found. Figure 4.2 shows the distribution of the top 20 categories.

As can be observed in Figure 4.2, the top seven categories represent the largest part of the data. We found that 74 categories were represented by less than ten episodes and 33 categories were only tagged by a single episode. Some of the categories represented by a single episode were 'Entrepreneur', 'cashflow' and 'Spanish' pointing to the large ambiguity in how categories are created and assigned highlighting a potential need for a better categorisation of episodes depending on their content. One example is the category assignment of 'Entrepreneur' which could be argued should fall under the category 'Business'.

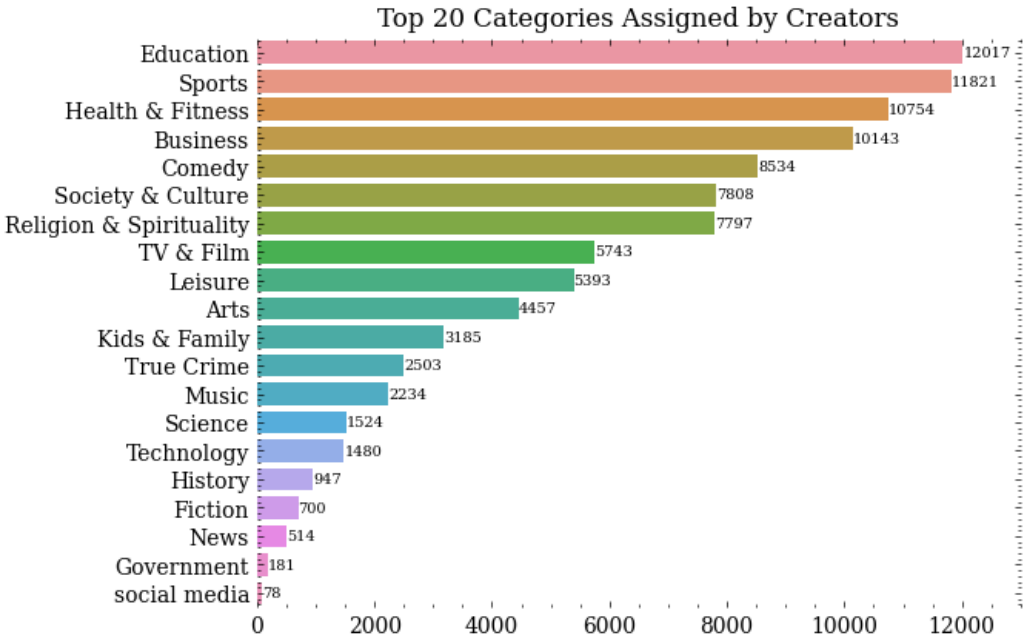

Figure 4.2: *Distribution of creator assigned categories.*

4.4.2 Episode Transcription using Google Speech-to-Text API

Transcription of the podcast episodes was performed by Clifton et. al. (2020) with the Google Cloud Platform Text-to-Speech API (GCP-ASR). In addition to the spoken words, it provides the start and end timestamps of each word and a confidence score for each sentence. Figure 4.3 shows an excerpt of the output of the Speech-to-Text API.


```
"transcript": "No words just lightning breaking darkness and crashing into the Earth with
brilliant presence. It seemed like the dream went on all night long. Okay, maybe it was
20 minutes. I wake up and the bedroom was filled with the tangible manifested presence of
Destiny and written out in front of me and great big Amber letters. I read Jo.",
"confidence": 0.8400493264198303,
"words": [
    {
        "startTime": "0.800s",
        "endTime": "1.100s",
        "word": "No"
    },
    {
        "startTime": "1.100s",
        "endTime": "1.900s",
        "word": "words"
    },
```


Figure 4.3: *Excerpt of raw data.*

Each sentence gets a confidence score which appreciates the word accuracy rate of the sentence (1 – [word error rate])(Clifton et. al., 2020). For each transcript, we averaged the confidence scores over all sentences and got the distribution for the full dataset depicted in Figure 4.4. The mean confidence is 83.2 percent with the extreme of some transcripts averaging below 60 percent accuracy. This score reflects the complex nature of the dataset discussed in section 4.4.1 and poses a challenge for downstream processing and modelling tasks.

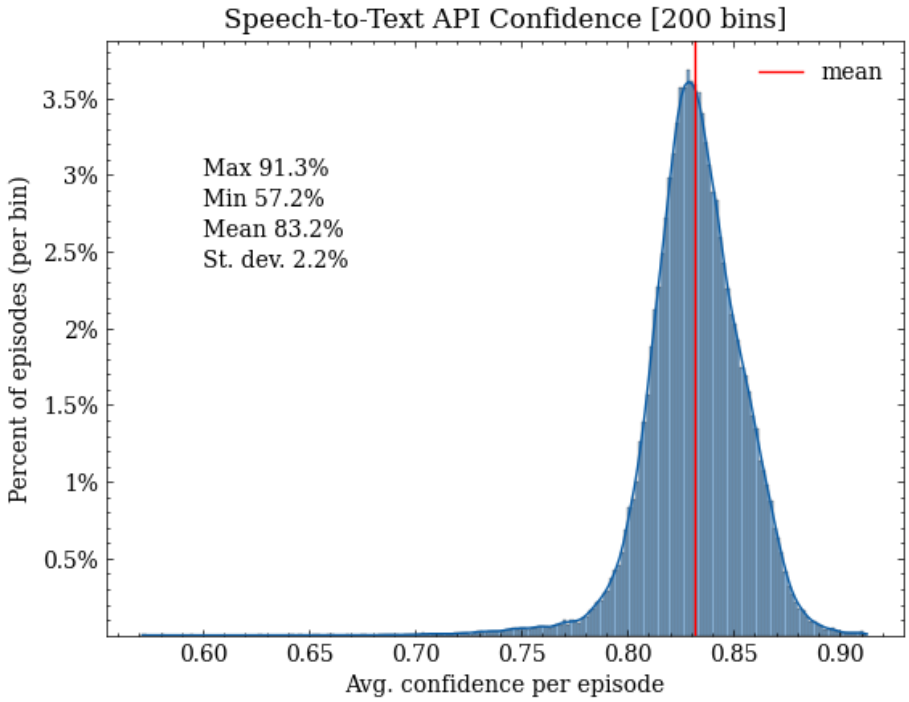

Figure 4.4: *Distribution of the average confidence score per transcript from the Google Speech-to-Text API.*

4.5 Data Preprocessing

In this section, we describe the data preparation stage of CRISP-DM, highlighting the techniques and steps taken to prepare, refine and annotate the dataset.

4.5.1 Data Wrangling

The first task in the data processing was to transform the output from the format delivered by the Google Speech-to-Text API. The data of each episode was delivered in a JSON file in the formatting displayed in Figure 4.3. Each episode JSON file was put in a directory representing its show. In that directory, all other episodes of the same show were put, thus each directory contained between one and 1,072 JSON files. Moreover, the show directories were put in directories representing the first character of their ID, thus ranging from 0 to Z. Each of these 37 directories was then distributed uniformly across three additional directories. Additionally, a separate file with the metadata of each episode was delivered. Extraction of the pure transcripts was done by walking the hierarchy of directories and opening each JSON file from which sentences and confidence scores were extracted and aggregated for each episode. This exercise resulted in a dataset with size 105,306 by three with columns '`episode_id`', '`transcript`' and '`confidence`'. This dataset could then be joined with the metadata on '`episode_id`' and represents the base on which the following processing and modelling have been done.

From this point, it was easy to analyse the condition of the transcribed transcripts. Figure 4.5 shows an excerpt from a transcript in which the complex nature of the data can be

well observed. Analysing the excerpt qualitatively, it is possible to understand the topic of the conversation on a high level and gather that there is more than one speaker partaking in the conversation. Typical traits of spoken language can also be observed such as in the second row where one of the speakers repeats the word '...but...' four times and the phrase '...the thing...' two times.

```
They're putting in the context of is Mike bad pack in trouble, which I
don't think he is, but but but but the thing the thing is it's like it
that's internally they're not saying that because this is a tough schedule
and and but but externally the fans I mean you look at the and you can
believe it you can not believe it but the commentary on social media the
commentary on Twitter. I know it's whatever but you know There is a lot of
criticism of Babcock right now. There's a lot of criticism of the star
players who again last night really did not show up. I mean Austin
```

Figure 4.5: *Excerpt of a transcript after preprocessing.*

The second phase of the data wrangling process demanded fetching the creator-fed categories for each podcast episode. In the metadata for each episode, a link to the show's RSS feed is available where additional metadata for both the show in general and each episode is made available. Rich Site Summary (RSS) is a standardized XML-based format for delivering regularly updated content from a website or a platform. In the context of podcasts, the RSS feed gets updated with the latest episode information and metadata and podcast directories, platforms, and applications use the feed to fetch new episodes and display them to subscribers. To further augment our data, the category for each show was fetched using the *feedparser* package in Python. The decision to further enhance the available metadata is part to improve our ability to perform insightful exploratory analysis but also illustrates the high-level and inconsistent categories made available on podcast platforms. The comprehensiveness of the extracted data is limited by inconsistencies in the RSS feed format, resulting in a 96 percent retrieval rate for categories

4.5.2 Data Filtering

From the original dataset, we have additionally filtered it in two ways. Firstly, after analysing the distribution of language tags we saw that only 132 episodes were tagged with non-English language tags, such as ['pt'], ['hi'] and ['es']. Therefore, we chose to remove those transcripts. Secondly, because of the sheer size of the dataset and the modelling task we wanted to perform, the number of transcripts to use in the modelling had to be brought down. This decision was made after trying to perform non-linear dimensionality reduction using various techniques on the full dataset causing a machine with an NVIDIA A100-SXM4-40 GB GPU and RAM of 83 GB to fail, effectively ruling out the possibility of modelling the full dataset.

Two approaches for cropping the dataset were considered. (1) Taking a random sample of 15 000 transcripts or, (2) subset the dataset based on a creator-assigned primary category. Since this study aims to identify topics which are meaningful for adverting, we chose the second method of sub-setting to highlight the inconsistency of the current categorisation. Naturally, choosing the largest category was our starting point. However, analysing the word count and duration distribution of the category, as can be seen in the top row of Figure 4.6, showed that most of the transcripts are rather short and for the purposes of the

segmentation task, this may not be optimal. Longer transcripts should intuitively contain a higher number of segments and therefore are better suited for the task. Therefore, we plotted the same distributions but for the second largest category, 'Sports', shown in the bottom row of Figure 4.6. The length of these transcripts was concluded to be better suited for the downstream tasks with 726 podcasts shorter than ten minutes in the Sports category as compared to 4,286 podcasts shorter than ten minutes in the Education category which can be observed by comparing the red vertical lines in the two-word count distribution plots respectively. The confidence score distribution resembles the one of the full dataset. The total number of episodes in the Sports category is 11,821 which is what comprises the final dataset used in this study. From here on, the term 'dataset' will refer to the filtered version of the original dataset.

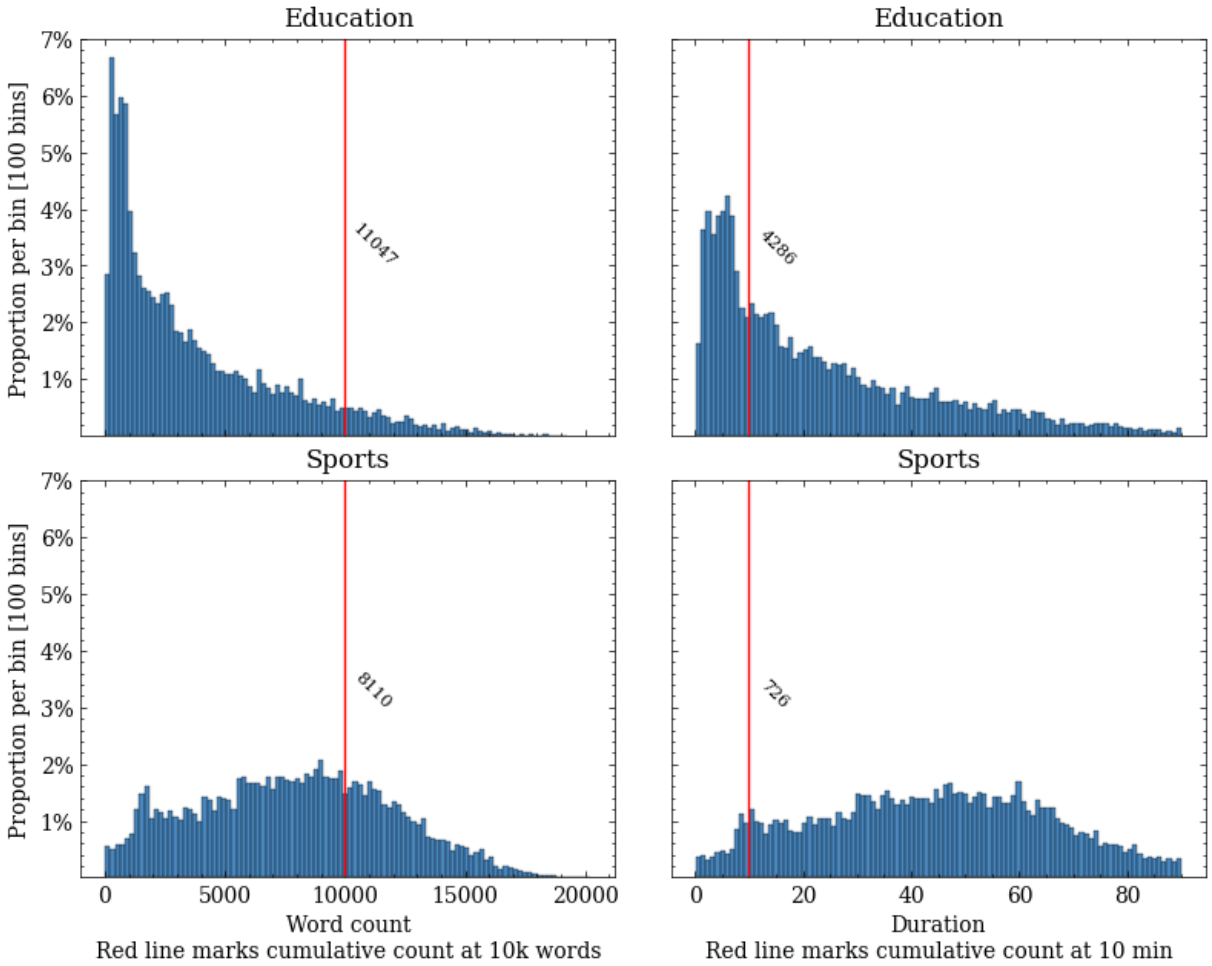

Figure 4.6: *Word count and duration distribution for the Education and Sports categories.*

4.5.3 Annotation

For the purposes of this study, a subset of the transcripts had to be manually annotated to evaluate the segmentation task. 20 transcripts were randomly sampled from the dataset for this purpose. Annotation was performed by the authors by reading the transcripts and inserting the symbol '@@' wherever a segment boundary was determined to be situated. Because of the imperfections caused by the uncertainty of the Speech-to-Text API and the nature of putting spoken language into writing without any post-transcription

formatting, gaining a detailed understanding of the content of each episode showed to be a difficult task. Therefore, the annotated segment boundaries could not be determined to be definite but rather hypothesised and their positioning might be off. Another approach to annotation is to listen to the podcasts on Spotify and insert segment boundaries in the written transcripts. However, this approach was not pursued due to time constraints.

Analysis of the annotated transcripts was performed partly to determine that the selected transcripts were a good representation of the full dataset. The average word count is 8,023, the minimum word count is 1,205 and the maximum word count is 12,860 reflecting the dataset distribution well. In total, 220 segments were annotated across all evaluation transcripts. The shortest segment is three sentences and the longest is 100 sentences long. The average segment size is 25 sentences.

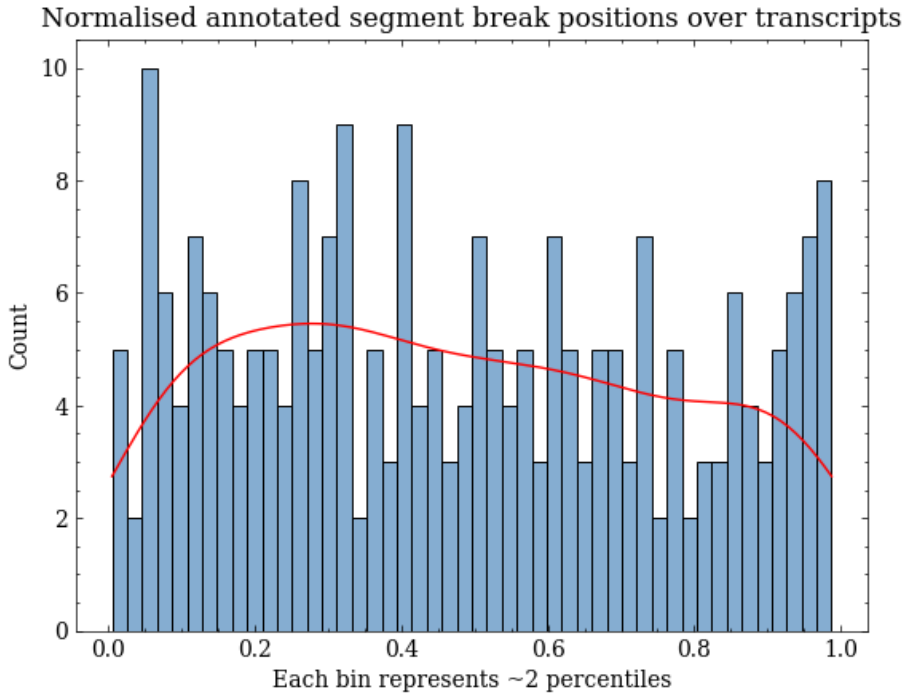

Figure 4.7: *Normalized annotated segment break positions over length of transcript across all transcripts.*

Additionally, the positioning of segment boundaries was analysed. In Figure 4.7 the normalised segment boundary positions are plotted with the fitted line representing the kernel density estimate (KDE) over the distribution. As evidenced by the KDE, the distribution of the positions is almost uniform meaning there is no positional bias in the annotation.

Automatic annotation by prompting GPT-3.5 Turbo was attempted. This proved to be an unsuccessful approach because of the generative nature of the model which made it return different segment boundary positions for the same transcript when prompted more than once. The segment boundary vectors could vary in length from 5 to 250 for the same transcript ranging 400 sentences in different answers. Therefore, this method was discarded.

4.6 Clustering Text Data

This section will go through the first part of the modelling pipeline which includes clustering the documents in the dataset and mention how we evaluate this step. This section together with sections 4.7, 4.8 and 4.9 will describe the data modelling and evaluation stages of CRISP-DM. Sections 4.6 to 4.9 describe together the full modelling framework proposed by this study which is illustrated in Figure 4.8.

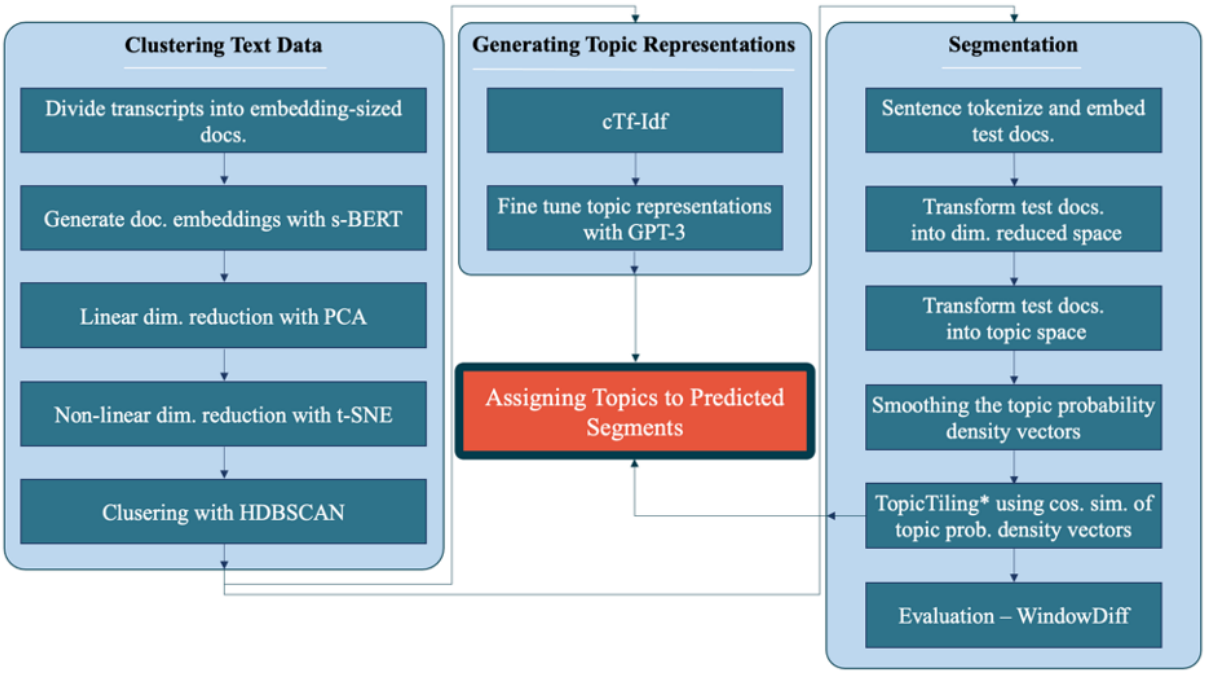

Figure 4.8: *Full modelling framework. Each step of this framework will be discussed and reflected upon in the upcoming sections of this chapter.*

4.6.1 Clustering and Topic Modelling with BERT

To perform the topic modelling, we chose to follow the pipeline suggested by Grootendorst (2022) called BERTopic. As discussed in the theory chapter, BERTopic leverages the most recent technologies within transformer neural networks, document embeddings and clustering to find dense regions of documents. In addition, Grootendorst suggests a modified version of the TF-IDF approach to finding topic representations by computing the TF-IDF based on classes (all documents in a cluster) instead of based on individual documents. Grootendorst has named this approach cTF-IDF. The BERTopic pipeline has four main stages: (1) generating document embeddings using a pre-trained transformer, (2) applying dimensionality reduction to the document embeddings, (3) clustering the documents and (4) generating representations for each topic. As suggested by Grootendorst, the topic representations generated in stage four by cTF-IDF can later be fine-tuned by for example prompting a generative language model. We decided to add this step to our pipeline by prompting GPT-3.5 Turbo for more interpretable topic representations. This will be further discussed in section 4.7.2. In addition to his paper, Grootendorst has created a Python library to facilitate research based on his paper. The library is modular

and allows for different algorithms in each step which facilitates model hyperparameter tuning across the pipeline. We started out using this library but found that the current implementation of the BERTopic library was unstable, proved by amplified runtimes for individual steps. Thus, we implemented the same structure but using other libraries for the purposes of this study. The following sections will undertake an in-depth examination of each step involved in the clustering pipeline and provide a discussion of the decision-making process underlying the choices made in each step. In section 4.7 we will discuss everything related to generating topic representations.

4.6.2 Embeddings

The first step of the modelling framework is to generate embeddings of the documents in the corpus. The approach which is suggested by Grootendorst (2022) and that we chose to pursue is to use a BERT transformer. BERT stands for Bidirectional Encoder Representation Transformer and maps documents into a high dimensional space where semantic and syntactic features are preserved. We used the s-BERT model 'all-MiniLM-L6-v2'[1] which is typically used for tasks such as clustering and semantic search. It is available via Huggingface and maps documents of maximum 256 tokens into a 384-dimensional embedding space. The model is a fine-tuned version of the 'nreimers/MiniLM-L6-H384-uncased' model. Fine-tuning was performed with a contrastive learning objective over one billion sentence pairs where the model was given a sentence from a pair and was tasked with predicting which of a random sample of sentences is the true counterpart. The training data is sampled from a multitude of sources and contexts giving the model strong resilience against different sources of input, making it appropriate for our dataset.

As our dataset consists of transcripts ranging up to above 20,000 words (see Figure 4.1), the transcripts had to be chunked before being fed to the BERT as they otherwise would have been truncated at 256 tokens. Two approaches were used to embed the documents. The first approach was to divide each transcript into lengths of 256 tokens, effectively dividing a transcript of e.g., 2,560 words into ten different documents which are considered separately by the model. The second approach was to use a sentence tokenizer from the Natural Language Toolkit[2] that divided the transcripts at each sentence. The resulting documents in the second approach therefore have varying lengths while in the first they have the exact same. Using the first approach we went from having 11,821 transcripts to having 368,835 documents of size 256 tokens. Using the second approach resulted in 5,460,190 documents. The number of document embeddings generated in the second approach became unfeasible to feed to the dimensionality reduction stage as computational complexity became too large for the machine to manage. Therefore, a random subsample of 500,000 documents was drawn from the tokenized sentences.

4.6.3 Dimensionality Reduction

Dimensionality reduction represents the process of finding an optimal representation of the data in an embedding space with lower dimensionality to make downstream tasks

[1] https://www.sbert.net/

[2] https://www.nltk.org/api/nltk.tokenize.html

less computationally expensive. It is often the case that some of the higher dimensions are poorly used by the data and can thus be removed with small amounts of information loss compared to the variance removed. Reducing dimensionality has been designed as a two-step process for the purposes of this study to leverage multiple characteristics of the different types of models presented in the theory chapter. The first step is to apply PCA to linearly project dimensionality down to a feasible number (Gisbrecht, Schultz & Hammer, 2014). In the second step, we reduce dimensionality non-linearly using t-SNE to a space where clustering can be performed efficiently. For the second step, UMAP and t-SNE were both evaluated which is discussed below.

Linear Dimensionality Reduction by PCA

As mentioned in the theory chapter, Principal Component Analysis (PCA) is a common method for reducing dimensionality by an orthogonal linear projection. It works by transforming the data $x \in \mathbb{R}^M$ to a new coordinate system $x \in \mathbb{R}^m$ where $M > m$ such that the largest variance by some scalar projection gets attributed to the first axis and subsequent scalar projections that reduces the variance gets ranked by the amount of variance reduced and attributed to the subsequent axes (Joliffe & Cadima, 2016). The axes of the new coordinate system are called principal components. From this projection, we can choose how many principal components to keep and from there also calculate how much variance is kept in the data.

The application of PCA to our dataset is primarily used as a step to reduce the computational complexity in the succeeding step. This decision was taken after trying to reduce dimensionality to $\mathbb{R}^5$ with PCA and evaluating the clustering model which resulted in nearly every document getting assigned to the same cluster. The complexity of PCA is $\mathcal{O}(p^2n + p^3)$ where p is the number of features and n is the number of instances (Géron, 2019). t-SNE used in the successive step has a complexity of $\mathcal{O}(n^2)$ (Pezzotti et. al., 2016) but is dependent on the computation of pairwise distances of all instances which is faster in smaller vector spaces. Therefore, reducing the number of features with PCA reduces the complexity of the overall dimensionality reduction step. PCA is a parametric model meaning that it preserves the transformation from the former to the latter coordinate system, making the algorithm appropriate for mapping unseen data into the same space later.

The number of components kept is set to either 50 or 100 in different configurations of our modelling framework. For the embeddings with a document size of 256 tokens, $PCA(components = 50)$ reduced variance by 41.98 percent and $PCA(components = 50)$ by 24.61 percent. For the embeddings with single sentences representing each document, $PCA(components = 50)$ reduced variance by 51.05 percent and $PCA(components = 100)$ by 33.64 percent.

Manifold Learning using t-SNE

For the second step, both UMAP and t-SNE were evaluated after reviewing their properties presented in the theory chapter. UMAP is generally the preferred technique as it

attempts to preserve properties of both local and global structures in the data (McInnes et al., 2020). However, the choice fell on t-SNE because of problems with memory complexity caused by the large number of instances in our data making UMAP fail on a machine running an NVIDIA A100-SXM4-40 GB GPU and RAM of 83 GB.

t-SNE is short for t-Distributed Stochastic Neighbour Embedding and is a two-step algorithm that learns the local structure of the data in a higher dimension and finds the optimal embedding of it in a lower dimension. It was first proposed by (van den Maarten & Hinton, 2008). The technique is particularly efficient for reducing the dimensionality of data that lie on different but related low-dimensional manifolds which often is the case of high-dimensional data. It does this by focusing on keeping data points that are close to each other in a high dimension near each other in the low dimensional embedding. This is done by treating similarities of all data points to each other as joint probabilities that x_i would pick x_j as its neighbour (van den Maarten & Hinton, 2008). This joint probability is proportional to the probability density function under a Gaussian centred at x_i in the high dimension. The mapping in the low dimension tries to replicate the joint probability distribution of the high dimension but uses the student's t-distribution instead of a Gaussian to create more space in the low dimension (van den Maarten & Hinton, 2008). This alleviates the crowding problem which occurs when many points from a high dimensional space have the same distance to each other but that are mapped in different dimensions are mapped into the smaller space of the lower dimensionality based on distance.

The algorithm needs one input from the user, called perplexity, that controls the variance of the distributions. More practically this can be thought of as the number of data points similarities are computed against for each data point. Low perplexity makes the algorithm focus on local structure. Therefore, we have set the perplexity to 30 since we want to uncover granular clusters which hypothetically would generate meaningful topics for advertisement. Run-time increase linearly with perplexity, so a low value also allowed for faster computation. In our study, t-SNE has been used to reduce the dimensionality $\mathbb{R}^{100} \hookrightarrow \mathbb{R}^2$ and $\mathbb{R}^{50} \hookrightarrow \mathbb{R}^2$ in different configurations of the modelling framework.

Since the focus of t-SNE lies on local structure, one of the major critiques of t-SNE is that the clusters of the local structure are arbitrarily scattered in the lower dimensional space (Policar, Stražar & Zupan, 2019). Additionally, t-SNE is traditionally non-parametric and ill-suited to use for transformations of complementary data not seen in training. For the purposes of this study we, therefore, used a variation of t-SNE that can transform data points not seen in training into the lower dimensional space. The Python library OpenTSNE put forth by (Policar, Stražar & Zupan 2019) offers an optimised implementation of t-SNE which can embed new data. OpenTSNE embeds new data independently of each other and without changing the reference embedding. Therefore, the authors stress that large additions of unseen data are discouraged since the reference embedding will be void after a while.

After embedding and reducing the dimensionality of the dataset such that all data points were mapped in $\mathbb{R}^2$, the documents could be visualised as can be seen in Figure 4.9. In all four graphs, dense regions can be observed. Additionally, two things can be noticed. First, in the blue plots representing embeddings of sentence-size documents, there are several points spread out on the far left which are distant from other points. This could be a result of the random subsample of 500,000 documents that were drawn from the initial

dataset of over five million data points. Some subtopics within the sports category might therefore be ill-represented which results in this mapping in the embedding space. In the yellow plots, dense areas can be observed throughout the space. Secondly, when comparing the graphs representing $PCA(components = 50)$ and $PCA(components = 100)$ we can observe that the separation between dense regions is better when PCA components are set to 100. Therefore, we will proceed with the embeddings of documents with 256 tokens and $PCA(components = 100)$ visualised on the right in Figure 4.9 on account of the two observations presented above.

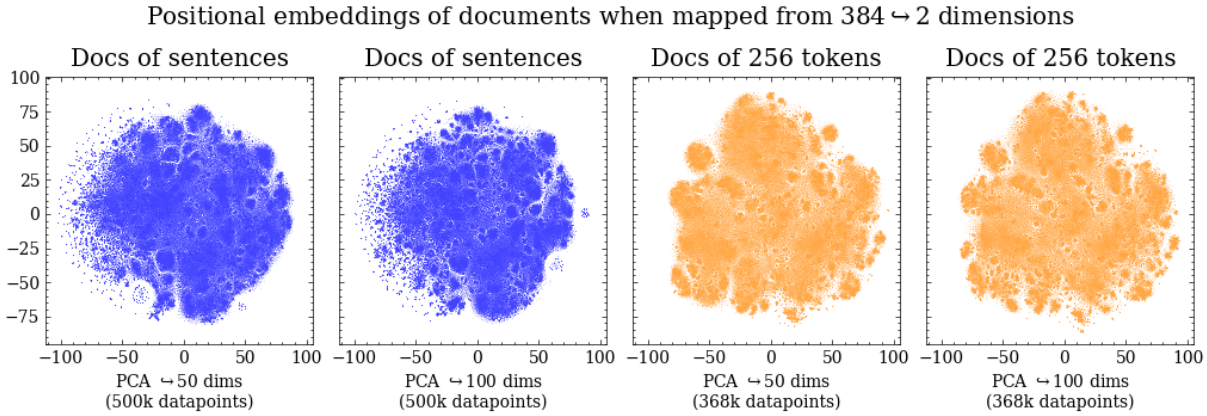

Figure 4.9: *Visualization of document embeddings after dimensionality reduction.*

4.6.4 Clustering

The next step in the modelling framework is to cluster the dimensionality-reduced embeddings such that topics can be found. For this task, we have used Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) proposed by (Campello et al., 2013). As our data contain a lot of noise as described in section 4.4.1 which has made the embeddings less accurate, we need a robust clustering algorithm for varying densities and non-spherical data patterns, ruling out approaches such as K-means and agglomerative hierarchical clustering as reviewed in the theory chapter.

To understand HDBSCAN, a basic understanding of DBSCAN is helpful. DBSCAN has two important hyperparameters, ε and m_{pts} where ε defines the global density threshold and m_{pts} controls the minimum amount of data points a cluster must have (Campello et al., 2013). It considers the data as an undirected graph with edge weights defined by an $n \times n$ symmetric matrix with pairwise distances defined by some metric $d(\cdot\,,\cdot)$, e.g., Euclidean distance, between data points x_p and x_q where $x_p, x_q \in X$. Core objects are defined as data points in the graph which have at least m_{pts} data points in its $\varepsilon - neighbourhood$ defined by $d(\cdot\,,\cdot)$. Clusters are defined as non-empty maximal subsets of X with respect to m_{pts} and $\varepsilon - distance$ such that every data point is density connected. Two data points are density connected if they are both defined as core objects and are directly or transitively $\varepsilon - reachable$. Points that are not core objects are considered noise (Campello et al., 2013).

HDBSCAN removes the need of specifying ε by building a density-based cluster hierarchy with varying levels of ε. It does this by introducing the concept of Core-Distance

(Campello et al., 2013). The Core-Distance for a datapoint x_p is equal to the distance of its $m_{pts} - 1$ neighbour (effectively including itself) and is denoted $d_{core}(x_p)$. The Core-Distance is the foundation for the next step called Mutual Reachability Distance which is defined between two data points as:

$$d_{mreach}\left(x_p, x_q\right) = \max\left\{d_{core}\left(x_p\right),\ d_{core}\left(x_q\right), d\left(x_p, x_q\right)\right\}$$

From the $d_{mreach}\left(\cdot,\cdot\right)$ the complete graph $G_{m_{pts}}$ is defined in which all data points that belong to X are vertices and the $d_{mreach}\left(\cdot,\cdot\right)$ are set as edge weights (Campello, Moulavi & Sander, 2013). In this graph, clusters can be found if we let $G_{m_{pts},\varepsilon} \subseteq G_{m_{pts}}$ and remove edges with weights above ε. By this definition, we can find partitions of the complete graph by varying $\varepsilon \in [0, \infty)$. At this point, HDBSCAN has accomplished the same things as DBSCAN if ε is defined by the researcher. If we instead measure the graph partition over all values of ε a hierarchy of nested clusters emerges. In the second step of HDBSCAN, the stability of the hierarchical clusters is measured over all values of ε, which is what allows for a variable density threshold. The stability of a cluster is defined by the following equation:

$$S\left(C_i\right) = \sum_{x_j \in C_i} \left(\frac{1}{\varepsilon_{min}\left(x_j, C_i\right)} - \frac{1}{\varepsilon_{max}\left(C_i\right)} \right)$$

Where $S\left(C_i\right)$ is the stability of cluster i, $\varepsilon_{min}\left(x_j, C_i\right)$ represents the ε value beyond which data point x_j no longer belong to cluster C_i, and $\varepsilon_{max}\left(C_i\right)$ is the minimum density level at which cluster C_i exist (Campello, Moulavi & Sander, 2013).

When clustering the embedded documents, three values for m_{pts} have been explored $\{15, 50, 200\}$ to find a cluster model suitable for the downstream tasks. Since the main purpose of the cluster model is to serve as a foundation for the text segmentation task, we have chosen to evaluate the hyperparameter tuning based on the evaluation metric of that task. Therefore, we refer the reader to section 4.8.3 for evaluation methodology. However, to get a better understanding of how the models perform we will also report their silhouette scores in the Results chapter.

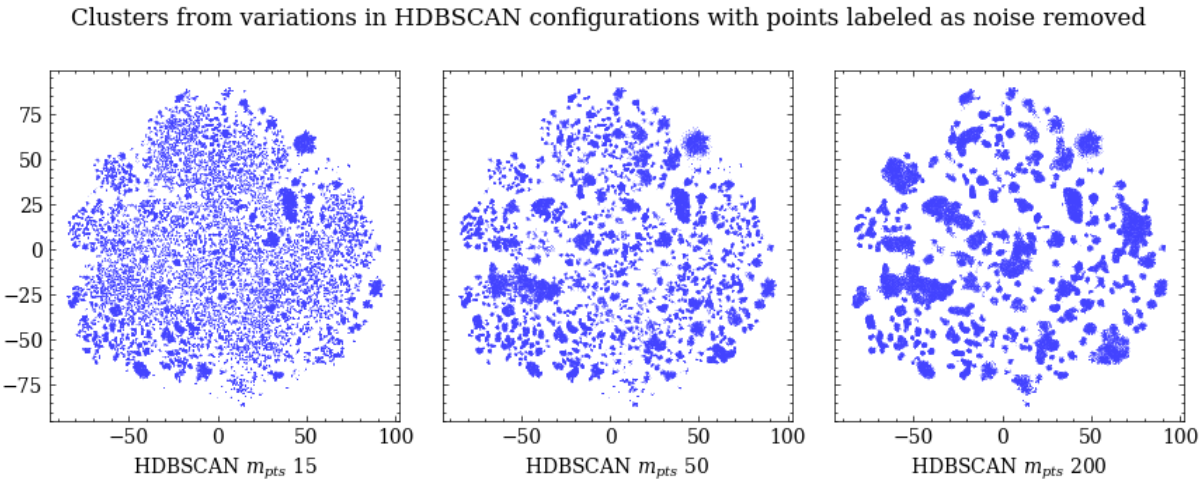

Figure 4.10: *Visualization of clusters across configurations after removing all points classified as noise.*

After clustering the data points, we plotted the clusters without outliers as seen in Figure 4.10. It can be observed from Table 4.1 and Figure 4.10 that when adjusting the m_{pts} variable from 15 to 200, the clusters increase in size and reduce in number while the number of data points classified as noise remains relatively equal.

$\mathbf{m_{pts}}$	**No. clusters**	**Noise**	**Largest cluster**
15	2,881	179,935	4,521
50	556	185,500	11,970
200	156	181,389	13,255

Table 4.1: *Descriptive statistics across cluster model configurations.*

4.6.5 Silhouette Score

The silhouette score is a common evaluation technique for cluster models proposed first by (Rosseeuw, 1987). It combines the measures of cluster separation and cluster cohesion into one metric which ranges between $[-1, 1]$. Cohesion is the measure of how similar an object is to other objects in its own cluster and separation measures how similar the object is to those of other clusters. A high silhouette score describes an object which is matched well to its own cluster and poorly to the neighbouring clusters. The similarity between objects can be measured with any similarity metric, in our case we have used Euclidean similarity as the data is normalised across dimensions. In short, the score is derived by computing $a(i)$ which is the average distance to all other points in the same cluster where $i \in \textit{cluster } A$ and $b(i)$ which is the average distance to all points in the closest neighbouring cluster (Rosseeuw, 1987). The metric assumes more than one cluster. Rosseeuw describes that by this definition we get the equation:

$$s\left(i\right) = \frac{b\left(i\right) - a(i)}{\max a\left(i\right),\ b\left(i\right)}$$

for each data point. The score then takes on values in the range depending on $a\left(i\right)$ and $b(i)$ when:

$$s(i) = \begin{cases} > 0 \text{ if } a(i) > b(i) \\ 0 \text{ if } a(i) = b(i) \\ < 0 \text{ if } a(i) < b(i) \end{cases}$$

We get a combined silhouette score for all classified data points by calculating the average $s(i)$ of all objects.

4.7 Generating Topic Representations from Clusters

4.7.1 c-TF-IDF

The topic generation of this study adopts the suggested modification to the Term Frequency – Inverse Document Frequency (TF-IDF) suggested by (Grootendorst, 2022) called class-TF-IDF. TF-IDF is a common method for evaluating how representative a word is to its document relative to a corpus of documents and is comprised of two statistical methods, (1) Term Frequency, TF, and (2) Inverse Document Frequency, IDF. The importance of a word increases with TF which measures how many times a word appears in the document and decreases with IDF which measures how prevalent the word is among other documents in the corpus. In this way, the importance of words that are frequently used across documents and subjects is offset. While a multitude of variations have been suggested (Jalilifard et. al., 2001; Hiemstra, 2000) the standard definition of TF-IDF (Robertson, 2004) where:

$$tf\,(t,d) = \ f_{t,d}$$

Where $f_{t,d}$ is the raw count of appearances of term t in document d. IDF is formalized as follows:

$$idf\,(t_i) = \ log\frac{N}{n_i}$$

where N is equal to the number of documents in the equation and n_i is the count of documents that term t_i appears in. cTF-IDF representations are found by calculating TF-IDF on a corpus of 'class documents' D_i constructed by merging all documents assigned to the same class C_i across the dataset such that $D_i = \sum_{x_j \in C_i} \ x_j$ for all $x_j \in C_i$ (Grootendorst, 2022). Furthermore, the altered IDF equation becomes:

$$cidf\,(f_i) = \log\left(1 + \frac{A}{f_i}\right)$$

where A is the average number of words per class and f_i is the frequency of word i across all classes (Grootendorst, 2022). From this representation, the top n words representative of each topic can be extracted. The final equation becomes:

$$ctfidf(t_i, D) = \ tf(t_i, D) \cdot cidf(f_i).$$

4.7.2 Enhancing Topic Representation using LLMs

Considering recent developments within the field of NLP with releases of Large Language Models such as GPT-3 from OpenAI and PaLM from Google, we wanted to explore how such a model could aid in understanding topics. For a small amount of money, it is possible to call the GPT-3.5 Turbo API offered by OpenAI stating a prompt which returns an answer to that prompt.

When calling the API, we must state what 'role' we as callers of the API have and the 'content' which is the prompt. Additionally, it is possible to tune a handful of parameters to control how the model behaves. Out of the tuneable parameters, we set the 'presence penalty' and 'frequency penalty' parameters to non-default values. Parameter space is

defined in the range [-2, 2] for both. The presence penalty increases the probability of receiving an answer about new topics when set to positive values. Frequency penalty decreases the probability of repeating the same verbatim. Both parameters were set to -0.5 to disincentivise deviant answers from the topic representations that were sent.

Prompting can be made in different ways depending on the purpose. We set the 'role' parameter to 'user' and 'content' to the following prompt:

> "I have a topic that is described by the following keywords: {topic_rep}.
>
> Based on the information above, describe the topic with approximately four words that would be descriptive for companies that want to advertise products or services related to the topic."

{topic_rep} represents the top n words representative of each topic extracted from the cTF-IDF.

4.8 Segmentation

This section will go through the second stage in the modelling framework and the evaluation techniques used.

4.8.1 TopicTiling

TopicTiling has been introduced in the literature review, however, a more comprehensive understanding is helpful to understand the modifications proposed in this study.

TopicTiling is an algorithm used for segmenting texts by changes in topics and was put forth by Riedl and Biemann in 2012. The algorithm starts by extracting topics from the training corpus by LDA. LDA assumes that documents are created by randomly sampling a distribution of topics that in turn consist of a pool of words. The Bayesian inference method of LDA assigns to each word a probability of belonging to each topic. TopicTiling takes each word and assigns it the topic ID with the highest probability, effectively representing each word by a single topic. In this way, documents are reduced in complexity from word space to topic space. The choice of not representing words by the full topic probability vector but instead incorporating this winner-takes-all approach reduces noise and random fluctuations (Riedl & Biemann, 2012). Additionally, it stabilises the topics at a low computational cost. The authors refer to this process as the mode of a topic assignment. For the mode of a topic assignment to be meaningful, the training data should be similar to the testing data.

Sentences are considered the smallest text unit in TopicTiling (Riedl & Biemann, 2012). At this point, the window parameter w is introduced which specifies the number of sentences to the left and right of each position p that a coherence score c_p is calculated. The left window w_L and the right window w_R are defined as $w_L = S_{p-w}, \ldots, S_p$ and

$w_R = S_{p+1}, \ldots, S_{p+w}$ where S_p denotes the sentence at position p. If T topics are extracted from LDA, each window gets assigned a T-dimensional topic vector that represents the count of words belonging to each topic in the window made possible by the mode of a topic assignment. The authors make no recommendation for the size of w and state that optimal values are different depending on the text. After this, topic tiling slides the window of size w over the document and calculates the c_p for all p. The coherence score used is cosine similarity. Plotting the c_p sequentially for a text makes it possible to discover the local minima. Each local minimum is treated as a candidate segment boundary.

For each candidate segment boundary, a depth score d_p is calculated. The purpose of the d_p is to measure the deepness of the local minima by finding the maxima to the left and right of it and applying the following formula:

$$d_p = \frac{1}{2}(hl\,(p) - c_p + hr\,(p) - c_p)$$

Where the functions $hl\,(p)$ and $hr\,(p)$ iterate to the left and right respectively until the coherence score stop increasing, which is then returned. Which of the candidate segment boundaries should be returned as predicted segment boundaries are then calculated by returning the positions with a d_p over a threshold set by $\mu - \frac{\sigma}{2}$ where μ denote the mean and σ the standard deviation of the depth scores of all local minima. An illustration of the concept applied to synthetic data can be found in Appendix A. The algorithm has a linear runtime proportional to the number of possible segmentation points, i.e., the number of sentences ($\mathcal{O}(N)$).

4.8.2 Updating TopicTiling using Topic Probability Distribution Vectors from HDBSCAN

In order to handle the noise in the dataset, we have developed and proposed a new approach to text segmentation and the TopicTiling algorithm using transformer-based document clusters instead of LDA as the foundation of vector representations of sentences. An HDBSCAN cluster model is first trained on a corpus of documents that have been embedded using a transformer. Like TopicTiling, the cluster model must be trained on a corpus similar to that of the test documents (Riedl & Biemann, 2012). Sentences S of the testing documents are then embedded and transformed into the topic space of the cluster model. From the positions of each sentence, we extract the probability that the sentence belongs to each topic. After doing this for all sentences in a transcript we get a matrix with dimension $T \times S$ of topic probability distribution vectors of a complete transcript. We denote this matrix as Y and the individual vector of sentence S_i is denoted y_i. To reduce noise and random fluctuations caused by transforming unseen data into a non-parametric cluster model we smooth all components of the vectors y_i that are below a threshold τ by setting them to 0. This smoothing step was introduced after noticing that the y_i vectors were dense and had many topic probabilities that were extremely small causing problems in the coherence score calculation. To replicate the sparse vector representations of the windows in TopicTiling (Riedl & Biemann, 2012) we set τ to $1e-20$ which increased the sparsity of Y significantly.

At this point, we return to the window parameter w which is defined identically as in TopicTiling. The topic probability distribution vectors for each sentence y_i in the windows are added, such that the window topic probability distribution vector becomes:

$$\sum_{i=1}^{w} y_i$$

$$\forall y_i \in w \text{ where } w \subsetneq Y$$

For each position p the coherence scores are calculated based on the window topic probability distribution vectors. Local minima representing candidate segment boundaries are found in the same way as in TopicTiling. We use cosine similarity as coherence score c_p which is calculated as follows:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

An additional smoothing mechanism is introduced in the calculation of depth scores. We chose to implement this step when discovering that the curve of coherence scores over transcripts exhibits a fluctuating pattern in local ranges. Hence, in the functions $hl(p)$ and $hr(p)$ we have added a stopping criterion where c_p must consistently decrease over n values from the first decreasing c_p for the function to return that value. In our implementation, n is set to 3. The depth score threshold is calculated in the same way as in TopicTiling and from that, candidate segment boundaries are converted to predicted segment boundaries if they meet the condition of the threshold. The algorithm with the modifications to TopicTiling reviewed in this section will be denoted as TopicTiling* from this point on.

Hyperparameter overview in TopicTiling*	
w	Window size
τ	y_i vector smoothing threshold
n	Depth score smoothing distance

Table 4.2: *Hyperparameter overview in TopicTiling**

4.8.3 Evaluation - WindowDiff

As an evaluation metric serving both HDBSCAN and TopicTiling* we have chosen WindowDiff which is commonly used in Text Segmentation. WindowDiff was suggested by (Pevzner & Hearst, 2002) as an improvement of the commonly used Pk score put forth by (Beeferman et al., 1999). As explained by Pevzner and Hearst, the evaluation of text segmentation has several challenges tied to it. First, human annotators do not always agree on where a segment boundary is located. Secondly, in different types of texts, near misses may or may not be acceptable to different degrees. A near miss is defined as a predicted segment boundary located in the near proximity of the actual boundary. Precision

and Recall are two metrics used for Text Segmentation where Precision is the percentage of boundaries predicted by the algorithm that are actual boundaries and Recall describes the percentage of all actual boundaries that are identified by the algorithm. However, the inherent trade-off between the two makes them difficult to optimise for (Pevzner & Hearst, 2002). Additionally, they are insensitive to near misses (Pevzner & Hearst, 2002).

The Pk score tried to improve on Precision and Recall by sliding a window with a size equal to half of the average true segment size over the transcript with both actual and predicted boundaries inserted. At each position, the algorithm increases a penalty counter if the ends of the window are in different segments when compared between the actual and predicted boundaries. The result is then scaled to the range $[0, 1]$ where 0 describes a perfect score. The problems with this approach, as described by (Pevzner & Hearst, 2002), are that it penalises False Negatives more than False Positives, ignores the number of boundaries, is sensitive to variations in segment size, near-misses are penalised too much, and the numbers are difficult to interpret because of the unclarity in how they are scaled. Therefore, they suggest the WindowDiff metric as a remedy for the problems and is the evaluation metric used in this study.

WindowDiff works in a similar fashion as the Pk score by employing a sliding window of size k which for each position compares the number of predicted and actual segmentation boundaries (Pevzner & Hearst, 2002). A penalty is calculated for each position where the actual (a_i) and predicted (p_i) segment boundaries are not equal, $|a_i - p_i| > 0$. Formally, WindowDiff is calculated by:

$$WindowDiff\left(ref, hyp\right) = \frac{1}{N-k} \sum_{i=1}^{N-k} \left(\left| b\left(ref_i, ref_{i+k}\right) - \ b\left(hyp_i, hyp_{i+k}\right) \right| > 0 \right)$$

Where $b\left(ref_i, ref_{i+k}\right)$ signify the number of actual boundaries between position i and $i+k$ and $b\left(hyp_i, hyp_{i+k}\right)$ signify the number of predicted boundaries between position i and $i+k$. N represent the number of sentences in the document and k is the window size. $WindowDiff\left(ref, hyp\right) \in [0, 1]$ where 0 is a perfect score and 1 describes a state where the algorithm has predicted boundaries for all possible positions in the document except where the actual boundaries are found

4.9 Topics of Predicted Segments

To monetise podcasts in an efficient way, in addition to segment breaks serving as advertisement spots, we want to understand what topics are discussed in the segments surrounding a boundary. Intuitively, once predicted, each segment could have gone through the topic modelling pipeline, meaning that we embed, reduce dimensionality, and predict clusters for each segment. However, this approach has two main drawbacks: (1) Segments longer than 256 tokens would have been truncated by the embedding model and (2) it would incur a lot of additional processing. Therefore, since the topic probability density vectors of each sentence have already been extracted to find advertisement spots, we chose to add the corresponding vectors of each segment and find the position of the largest topic for each segment vector. This approach circumvents the problem of truncated segments

and yields a less computationally heavy process. We do this by adding the topic probability distribution vectors for each segment $seg_{i,k}$ with length k and find the index of the largest component in the vector.

$$\sum_{i=1}^{k} y_i$$

$$\forall\ y_i \in seg_{i,k}\ where\ seg_i \subseteq Y$$

Matching the resulting index with the index of the topic representations we infer the topic of seg_i.

Furthermore, from each topic assignment, we calculate a normalised certainty score that serves as a proxy for how certain the model is that the assigned topic is the true topic. Since any topic for any sentence can have a maximum value (probability) of 1, the maximum value that the segment can have for the assigned topic is equal to the number of sentences (k) in that segment. Therefore, we divide the scalar of the added probabilities corresponding to the winning topic of each segment by k to get a normalised score. The score takes on values in the range $[0, 1]$ where the score increases as the confidence of the model increase.

5 Results

5.1 Topical Segmentation of Podcast Transcripts

In Chapter 4 we formulated the methodology applied in this study to address the problems presented by the research questions. This chapter will present the performance of the modelling framework considering the research objectives both qualitatively and quantitatively. To achieve this, all three configurations of HDBSCAN presented in section 4.8.3 have been tested as part of TopicTiling* and evaluated on a range of TopicTiling* window sizes (w) in the range [2, 40] with a step size of two, effectively evaluating 60 modelling framework configurations in total.

5.1.1 Evaluating the Segments using WindowDiff

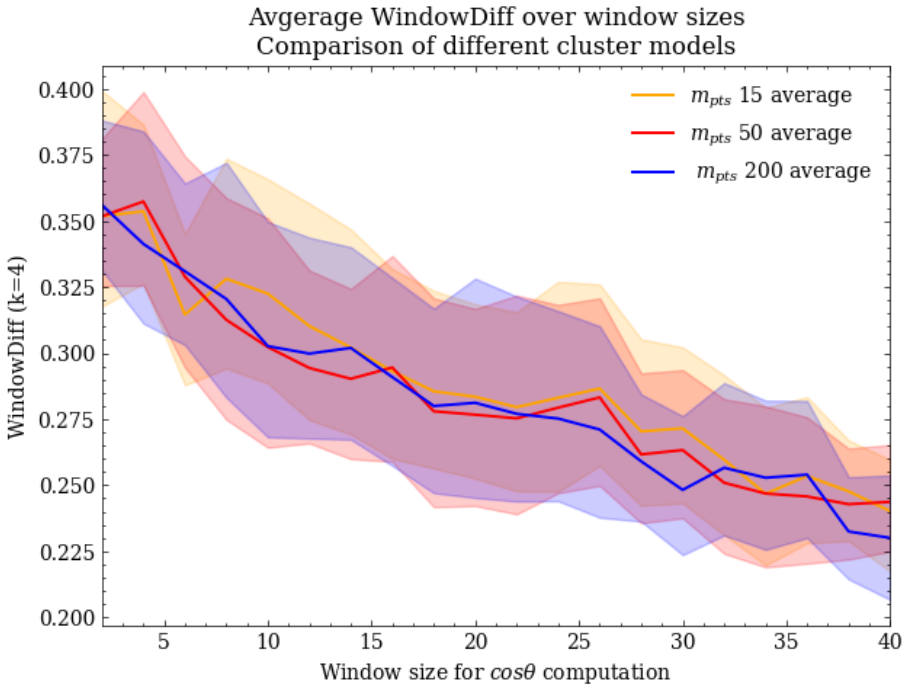

Figure 5.1: *Average WindowDiff over w for all cluster models*

The selected method of evaluation is WindowDiff whose functionality is described in section 4.8.3. As the evaluation metric is designed to allow for close matches to be accounted for in the final score, we had to set the window (k) for what is considered a

close match. We selected $k = 4$, which means that a predicted segment boundary with a maximum sentence distance of 4 will improve the score and anything beyond that will reduce it.

In Figure 5.1, the WindowDiff scores for all 60 models are plotted. Each line represents an HDBSCAN configuration, and the X-axis describes the TopicTiling* w parameter. The Y-axis represents the WindowDiff of each model. The shaded areas corresponding to each line visualize the confidence interval with $\alpha = 0.05$. The scores are downward sloping for all models as w increases, indicating a better match between actual and predicted segments. The dimensionality of the topic probability distribution vectors (different m_{pts} settings) is shown to not affect the WindowDiff score as the three lines follow each other closely across w. Overall, the average score across w for the three models is around 0.29, as displayed in Table 5.1, with HDBSCAN $m_{pts} = 200$ having the best results. This score is in line with what the original TopicTiling algorithm based on LDA scored in the evaluation (Riedl & Biemann, 2012). This suggests the robustness of the methodology proposed in this study considering the noisy input data, which made clustering increasingly difficult

m_{pts}	**Avg. WD for w in range [2, 40]**
15	0.2900
50	0.2848
200	0.2840

Table 5.1: *Average WindowDiff score for all three configurations.*

5.1.2 Determining the Window Hyperparameter in TopicTiling*

To conclude the optimal setting of w we investigated the performance of the three models on the individual transcripts in the annotation set. In Figure 5.2 we can observe the performance of the three cluster models on all transcripts. The two truncated lines at the top of the plots represent two transcripts with less than 100 sentences. For these two, measurements were stopped when $w \times 2 \geq$ *number of sentences*. We can observe a negative exponential relationship where the change in WindowDiff between two positions becomes smaller as w increases. The largest change is observed in the three curves when w takes on values below 20. Hence, for the forthcoming results, we will discuss models with $w = 20$.

5.1.3 Effect of Cluster Model Configuration on TopicTiling*

The cosine similarity range is defined as the difference between the maximum and minimum cosine similarity that is calculated for each transcript. Taking the average of the differences of all transcripts in the annotation set gives the size of the range for a model. A pipeline configuration which makes use of a larger range of cosine similarities would potentially increase the threshold of the depth scores which impacts the number of boundaries that are converted from candidate to predicted boundaries. A model which has a higher depth score threshold could be more robust against randomness and noise in the topic probability distribution vectors and therefore potentially yield more reliable bound-

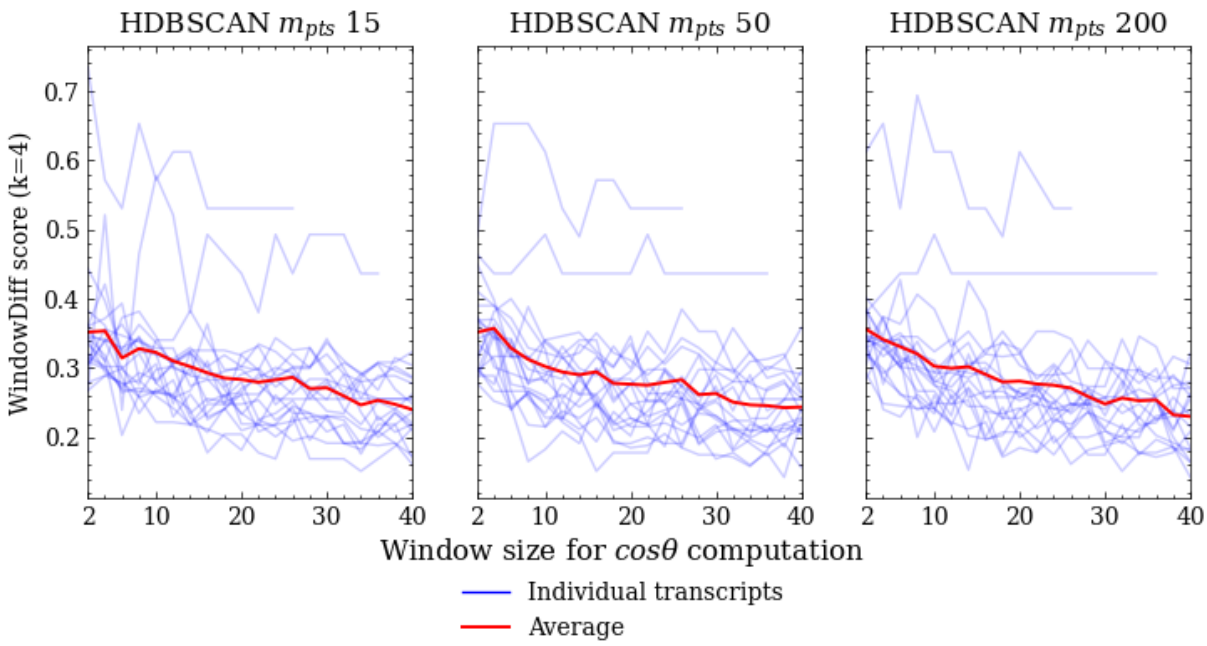

Figure 5.2: *WindowDiff scores for each transcript in the annotation set for each cluster model.*

aries. Therefore, investigating the effect of different configurations of the cluster model on the range of cosine similarities and depth scores is relevant.

$\mathbf{m_{pts}}$	**Avg. cos(θ) range size**	**Avg. depth score**
15	0.714	0.456
50	0.720	0.551
200	0.664	0.448

Table 5.2: *Average cosine similarity range and average depth scores across models.*

In Table 5.2 the impact of cluster model configuration on cosine similarity range size and average depth scores can be observed across the full annotation set for the three models. The average depth score for the three models increases as m_{pts} shifts from 15 to 50 and then decreases again as m_{pts} go to 200. The cosine similarity range is approximately the same for the two models with m_{pts} set to 15 and 50 and then drops down for the m_{pts} 200 model. A high average depth score in relation to the average cosine similarity range means that the model produces high local maxima and low local minima relative to the cosine similarity range used. This could implicate that the model is more robust against randomness in local variability of the sequential cosine similarities of a transcript.

$\mathbf{m_{pts}}$	**Avg. cos(θ) range**	**Avg. depth score**	**WD**
15	0.811	0.339	0.26
50	0.862	0.487	0.22
200	0.888	0.431	0.22

Table 5.3: *Average cosine similarity range, depth scores and WindowDiff score for example transcripts.*

To illustrate an example of this, we have in Figure 5.3 plotted the cosine similarity between windows over all positions sequentially for one of the annotated transcripts as the blue line. The red crosses mark where boundaries are predicted to be situated and the actual

boundaries are shown as yellow vertical lines. As seen in the accompanying Table 5.3 which displays the descriptive statistics, as m_{pts} increases from 15 to 200 the range the cosine similarity takes on increases from 0.811 to 0.888, and the depth score average increase from 0.339 to 0.431. Clustering with $m_{pts} = 50$ gets a range of 0.862 and an average depth score of 0.487. As can be observed in the rightmost column is that for the two models where the range is smaller and the average depth score is higher ($m_{pts} = \{50, 200\}$), the models are better at placing segmentation boundaries as evidenced by the WindowDiff scores. However, this can only serve as a demonstration of how the cluster model may impact TopicTiling*. To make any conclusions about this finding, an evaluation of a larger set of annotated data would be required than what is possible in this study.

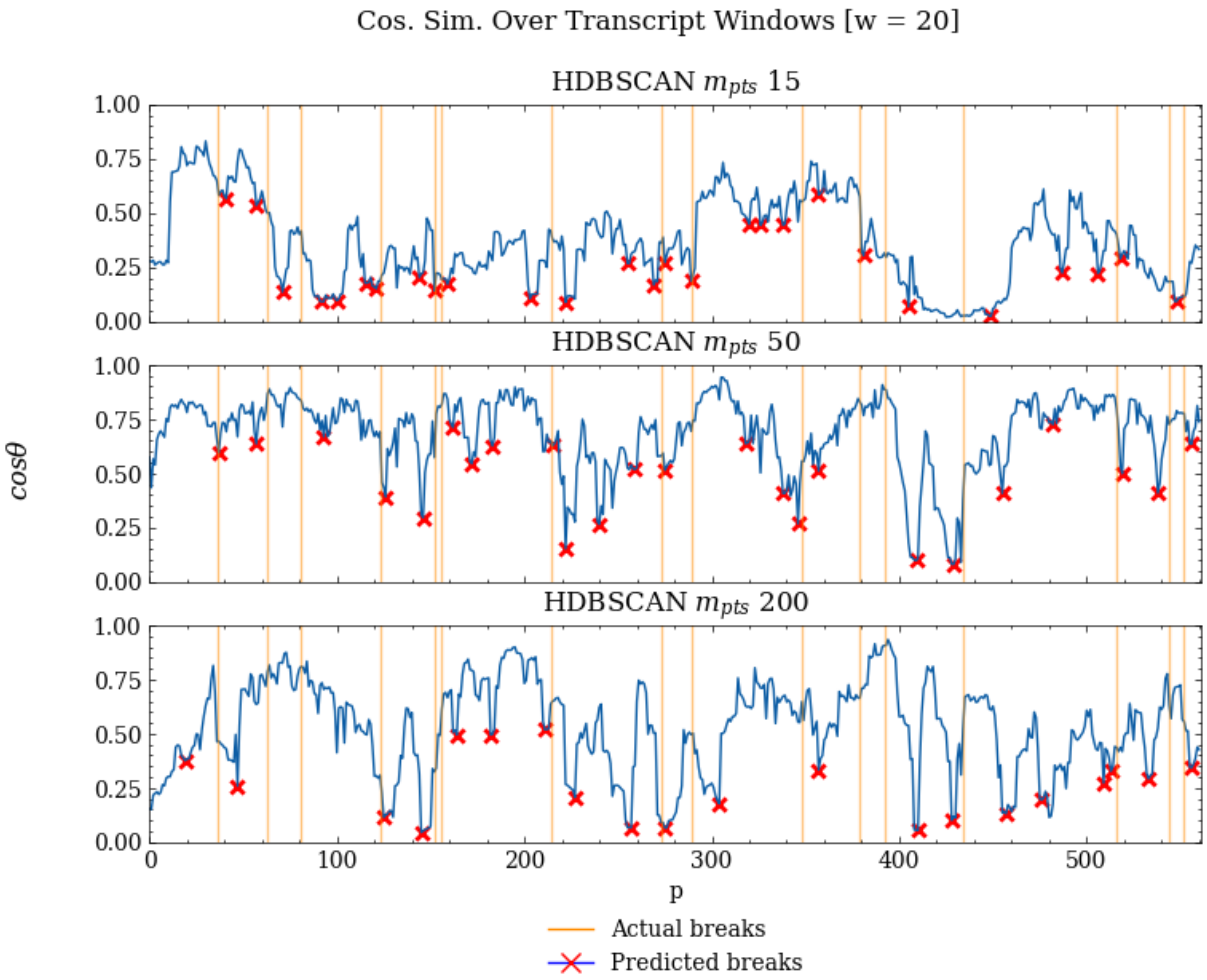

Figure 5.3: *Cosine similarity, and actual and predicted segment breaks for one example transcript.*

5.2 Enabling Native Advertisement in Podcasts

To enable native advertisement in podcasts where the content of the advertisement matches the content in the adjacent segments, we need to be able to understand which topic is most prevalent in each predicted segment. We start by constructing descriptive topic representations of each cluster and follow by assigning topics to each predicted segment.

5.2.1 Topic Representations from cTF-IDF

After removing English stop words (see Appendix B) and using cTF-IDF to extract the top ten representative words for all topics, we got the lists of topic representations as presented in Tables 5.4, 5.5, and 5.6.

The top ten words for each cluster give a description of the contents of its documents. When evaluated qualitatively, the topic representations from c-TF-IDF are separated well and provide descriptive representations of their topics. However, certain topics include words that may make sense to listeners but might be unclear to advertisers who may not be well-versed or familiar with individual subjects (e.g., 'wwe', 'leafs' and 'rory').

Largest topic	Smackdown	Ring	Wcw	Rumble	Wrestlemania	Nxt	Wrestlers	Wrestle	Wrestler	Wwe
2nd largest	Sri	Batting	Indian	Indies	Wickets	Bowling	Batsman	Pakistan	India	Cricket
3rd largest	Tee	Courses	Tour	Swing	Rory	Golfers	Golfer	Pga	Golf	Tiger
4th largest	Gordon	Pats	Julian	Afc	Gronk	Patriot	Patriot	Edelman	Belichick	Brady
5th largest	Exercise	Trainer	Halter	Exercise	Horsemanship	Clicker	Saddle	Behaviour	Reinforcement	Horses

Table 5.4: *Topic representations for top five largest topics of* $m_{pts} = 15$

Largest topic	Offensive	Backs	Tight	Touchdown	Tackle	Linebacker	Receivers	Yards	Wide	Receiver
2nd largest	Wrestlemania	Nxt	Matches	Wrestlers	Wrestle	Ring	Match	Wrestler	Wwe	Wrestling
3rd largest	Australia	Batting	Indian	Batsman	Bowling	Pakistan	England	Test	India	Cricket
4th largest	Catch	Rod	Kayak	Boat	Lake	Water	Bass	Bait	Fishing	Fish
5th largest	Stanley	Hockey	Toronto	Defenseman	Bruins	Maple	Matthews	Babcock	Nhl	Leafs

Table 5.5: *Topic representations for top five largest topics of* $m_{pts} = 50$

Largest topic	Guy	Line	Ball	Quarterback	Defence	Offense	Running	Wide	Yards	Receiver
2nd largest	Ran	Pace	Races	Mile	Miles	Run	Training	Running	Marathon	Race
3rd largest	Boston	Stanley	Puck	Ice	Playoffs	Bruins	Blues	Nhl	Leafs	Hockey
4th largest	Innings	Sox	Astros	Series	Runs	Bullpen	Brewers	Yankees	Baseball	Pitching
5th largest	Program	Basketball	Parents	High	College	Coaching	Coaches	Kids	School	Coach

Table 5.6: *Topic representations for top five largest topics of* $m_{pts} = 200$

5.2.2 Fine-tuned Topic Representations by LLM

When fine-tuning the topic representations by prompting GPT3.5 Turbo from OpenAI as described in section 4.7.2 we get the topic descriptions shown in Table 5.7. This mitigates the problem with topic-specific words described in the previous section as GPT can leverage additional information sources to generate meaningful descriptions.

A drawback of this method is that different results are returned if prompted with the same input multiple times. In the context of matching relevant advertising to podcast segments, we conclude that the GPT representations are easier to comprehend than the c-TF-IDF representations and are therefore more useful for the purposes of this study.

GPT Topic Descriptions	
HDBSCAN $\mathbf{m_{pts} = 15}$	
Largest topic	Professional Wrestling Events
2nd largest topic	Cricket Batting and Bowling
3rd largest topic	Golf Courses and Equipment
4th largest topic	New England Patriots Team
5th largest topic	Horse Training Techniques
HDBSCAN $\mathbf{m_{pts} = 50}$	
Largest topic	Football Game Advertising
2nd largest topic	Professional Wrestling Matches
3rd largest topic	Cricket Batting and Bowling
4th largest topic	Fishing equipment and services
5th largest topic	Hockey in Toronto
HDBSCAN $\mathbf{m_{pts} = 200}$	
Largest topic	Football Game Advertising
2nd largest topic	Running and Marathon Training
3rd largest topic	NHL Playoffs Advertising Opportunities
4th largest topic	Baseball Playoff Advertising
5th largest topic	Youth Basketball Coaching Services

Table 5.7: *GPT topic representations for largest topics across models.*

5.2.3 Segment Topic Assignment on Full Transcript

The last stage in the modelling framework is to find the topics of the predicted segments. For each of the 20 transcripts in the annotation set, each predicted segment's topic assignment receives a certainty score. The certainty score for the full transcript is reported as a weighted average with weights representing the size of each segment. The average, maximum and minimum of all weighted averages for each of the three models are presented in Table 5.8. We observe that model $m_{pts} = 50$ has the highest average, whereas the other two models are relatively similar to model $m_{pts} = 200$ performing slightly better of the two. The values are still relatively low, with the highest weighted average for any of the podcasts in the annotation set scoring 0.2943.

Model (m_{pts})	**Mean**	**Max**	**Min**
15	0.1196	0.1898	0.0640
50	0.1960	0.2943	0.0772
200	0.1198	0.2067	0.0612

Table 5.8: *Weighted average certainty scores across models.*

In Table 5.9 we present an example of one of the annotated transcripts. Each row corresponds to a segment with its location within the transcript, corresponding topic number, certainty score and the topic representation from GPT. Across the three models, we observe a wide range of topics that are not always coherent. Additionally, the weighted average certainty scores are low. Neither t-SNE nor HDBSCAN are meant to predict unseen data which could be the reason to the observed results. Another cause might be that the documents used for training are 256 tokens long and the testing documents are one sentence long. The inconsistency of topic representations among the models hints

that the topics may not be granular enough to represent the actual segments. However, it seems like the consistency between adjacent topic assignments increases as the number of topics increase (as m_{pts} decreases).

Segment	Topic no.	Certainty	GPT Topic Description
Model: HDBSCAN $\mathbf{m_{pts} = 15}$			
Sentence 1: 35	243	0.180	Farewell and Celebration Products
Sentence 36: 65	243	0.264	Farewell and Celebration Products
Sentence 66: 80	1884	0.066	Gaming and Snacks
Sentence 81: 84	1580	0.240	Football Players and Coaches
Sentence 85: 96	243	0.091	Farewell and Celebration Products
Sentence 97: 114	243	0.225	Farewell and Celebration Products
Sentence 115: 134	243	0.082	Farewell and Celebration Products
Sentence 135: 152	243	0.088	Farewell and Celebration Products
Sentence 153: 177	243	0.160	Farewell and Celebration Products
Sentence 178: 188	1744	0.091	Child Abuse Prevention Awareness
Sentence 189: End	243	0.111	Farewell and Celebration Products
Weighted average certainty for full transcript:		0.146	
Model: HDBSCAN $\mathbf{m_{pts} = 50}$			
Sentence 1: 64	64	0.268	NFL rivalry and players
Sentence 65: 84	84	0.122	Soccer Team Players
Sentence 85: 130	130	0.296	Baseball Pitching Performance
Sentence 131: 150	150	0.118	Sports Training and Tournaments
Sentence 151: 168	168	0.330	Sky and Nature tourism
Sentence 169: 189	189	0.095	Green Bay Packers roster
Sentence 190: End	189	0.202	Green Bay Packers roster
Weighted average certainty for full transcript:		0.217	
Model: HDBSCAN $\mathbf{m_{pts} = 200}$			
Sentence 1: 31	125	0.098	Football player merchandise
Sentence 32: 44	133	0.080	Rap Music Listening Preferences
Sentence 45: 56	120	0.250	Youth basketball coaching services
Sentence 57: 74	120	0.064	Youth basketball coaching services
Sentence 75: 84	125	0.163	Football player merchandise
Sentence 85: 114	38	0.059	Sports Betting Promotions
Sentence 115: 146	125	0.089	Football player merchandise
Sentence 147: 158	33	0.084	Sports stadium atmosphere
Sentence 159: 187	72	0.058	College Football Conference Teams
Sentence 188: End	132	0.043	Controversial language and religion
Weighted average certainty for full transcript:		0.084	

Table 5.9: *Segments, topic no, certainty score and GPT topic representation for an example transcript across all models.*

5.2.4 A Deeper Look at the Topic Models

Seeing the output of the topic models, a deeper look at how they function might give some additional insights into the results we observed earlier in this chapter. In Figure 5.4 we can observe the distribution of the top 25 topics of each model. In Table 5.10 we have summarized some descriptive statistics about each model. If we consider Figure 5.4 in conjunction with the insight that all models classified a similar number of documents as noise from Table 5.10, we can conclude that the distributions look relatively the same given the number of topics. What stands out is the relatively large top topic in HDBSCAN $m_{pts} = 50$ which seems to be proportionally larger than the rest when compared to HDBSCAN $m_{pts} = 15$ and HDBSCAN $m_{pts} = 200$. This might result in a smaller utility that can be derived from the model when it is used to assign topics to predicted segments.

However, this is mere speculation and would have to be confirmed on a supervised dataset with segment topics.

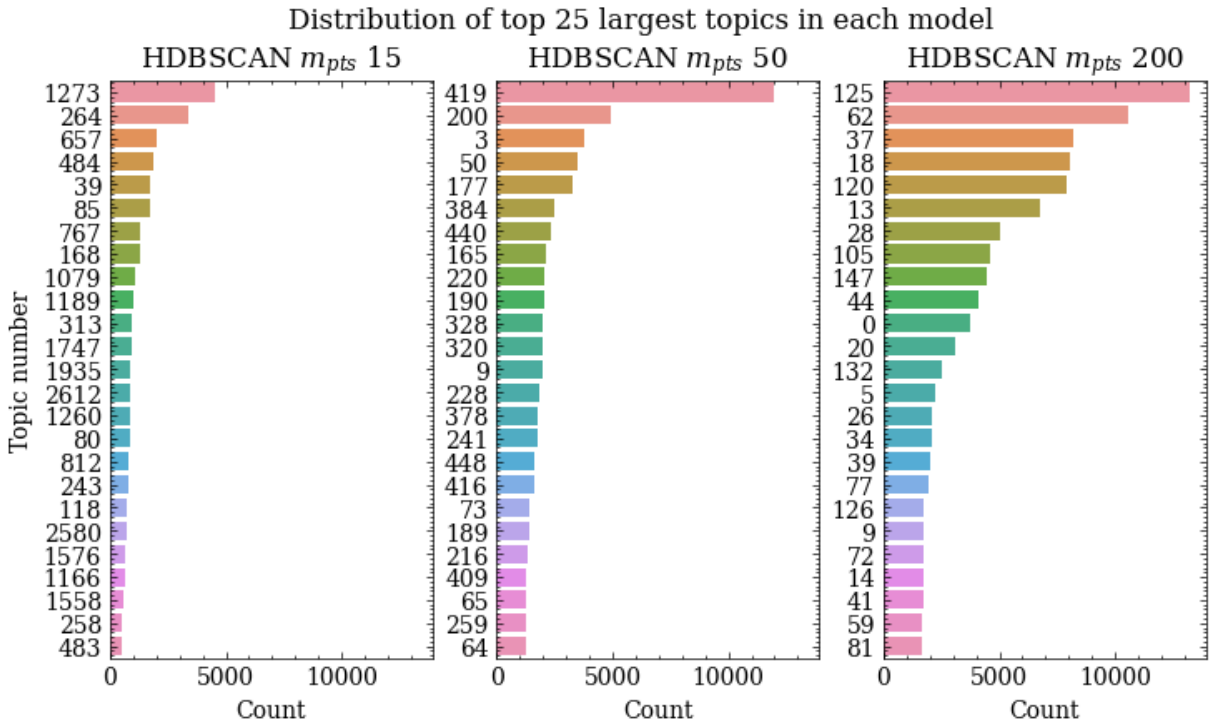

Figure 5.4: *Size of the top 25 largest topics of each model.*

Furthermore, we see that the HDBSCAN $m_{pts} = 15$ has the best silhouette score which measures the cohesion and separation of the clusters. This means that the clusters are better defined than in the other models and that there is less probability that a document assigned to a cluster belongs to another. However, as can be observed in Table 5.10, the margin is small.

$\mathbf{m_{pts}}$	**No. clusters**	**Noise**	**Largest cluster**	**Silhouette score**
15	2,881	179,935	4,521	0.4315
50	556	185,500	11,970	0.3564
200	156	181,389	13,255	0.3625

Table 5.10: *Descriptive statistics across cluster model configurations.*

6 Discussion

In this chapter, we will answer the research questions, analyse and assess our findings and discuss their implications. We will also discuss the study's limitations and provide a reference for future work.

6.1 Interpretation of Results

Approaching the problem of placing relevant advertisements in podcasts was done in two stages. First, advertisement spots (referred to as segment boundaries in this study) were allocated within the podcast transcripts. Secondly, the segments of the podcast between the advertisement spots were analysed to understand what topics are being discussed such that advertisement can be matched natively to the topic of the segment. To handle these two sub-problems, we developed the methodology presented in Figure 4.8 (modelling framework). Below, the results of each of the two sub-questions will be discussed.

The first research question this study set out to answer is how data science methods can locate topical shifts in podcast transcripts. TopicTiling was determined from the literature review to align best with the research objective. Considering recent headway within the NLP field, we decided to attempt shifting the methodology of how TopicTiling derives its topics from LDA to transformer-based clustering using HDBSCAN. Attempting such a feat was necessary given the non-uniformity of token representations created by the intrinsic complexities of the dataset, making a BoW methodology likely to fail.

The modified version of TopicTiling (TopicTiling*) suggested in this study demonstrates that it is possible to segment podcast transcripts in a meaningful way according to topical shifts. TopicTiling* has performed comparative results to those of TopicTiling when evaluated on WindowDiff ($k = 4$) with scores averaging 0.286 across the 60 model framework configurations. The data used in this study presented more noise compared to the data TopicTiling was evaluated on, suggesting the robustness of the methodology. However, as this study takes on an exploratory rather than a comparative research approach, we will not draw any definite conclusions about the comparative performances of TopicTiling* and TopicTiling. The demonstration of the performance of TopicTiling* progresses the technological development one step closer to enabling the segmentation of podcast transcripts at topical shift autonomously.

The second research question this study set out to answer is how data science methods can find topics that are meaningful for podcast advertisement. This problem was approached

from a topic modelling perspective. The methodology outputs topic representations for each segment which are derived from the topic probabilities of each sentence that make up the segment. To answer this question, we have evaluated the modelling framework by qualitative analysis of topic representations, topic assignments to segments over an example transcript and by computing their certainty scores. Beyond that, we have also presented the silhouette score of the cluster models to understand the cohesion and separation of the topics.

The topic representations presented in sections 5.2.1 and 5.2.2 are concluded to yield descriptive words and phrases of their underlying topics which could be leveraged for matching segments with native advertisements. However, when analysing an example transcript with assigned topics to predicted segments in section 5.2.3 we conclude that the topics are incoherent and hence unreliable in describing the underlying topics of identified segments. The assignments produced by the three model configurations get relatively low weighted average certainty scores which further supports this notion. However, the evaluation of the modelling framework with the certainty score as a proxy for a match between the actual and assigned topic is to be considered a heuristic and not a scientific method since it is not a well-researched metric. In this study, the certainty score is used to make the results comparable and set a benchmark for future research. We therefore conclude that the modelling framework serves as a proof-of-concept for this task as well and further efforts remain to find an optimal configuration of the hyperparameters for the use case of this study.

6.2 Implications

In this section, the implications of this study will be discussed. We will touch upon the implications from a technical and practical perspective and the potential implications of the methodology outside the confines of this study.

Implications of the results include the possibility of further scaling and enhancing advertisement capabilities on podcast platforms. If future efforts can produce a configuration of the modelling framework where the WindowDiff score is close to 0 for low values of k and where the topic assignments to segments are reliable, a feature could be built according to the business framework illustrated in Figure 6.1. A podcast is uploaded to the platform, and automatically transcribed with a Speech-to-Text API, segment boundaries are found with TopicTiling*, and topics are assigned to each segment. From this point the platform lets advertisers choose from the list of topics which topics they want to advertise next to and the advertiser provides a pre-recorded advertisement. The platform then distributes the advertisement across podcasts with segments matching the selected topics. This would increase the value proposition of the podcast platform for advertisers and creators.

As podcast platforms continue to expand their advertising capabilities via algorithms, data, and AI, they strengthen their position to benefit from network effects and other aspects of platform theory (Iansiti & Lakhani, 2020). The more advertisers, content creators, and listeners the platform attracts, the more data it gathers, and the more efficient TopicTiling* becomes at advertisement placement and targeting. This, in turn,

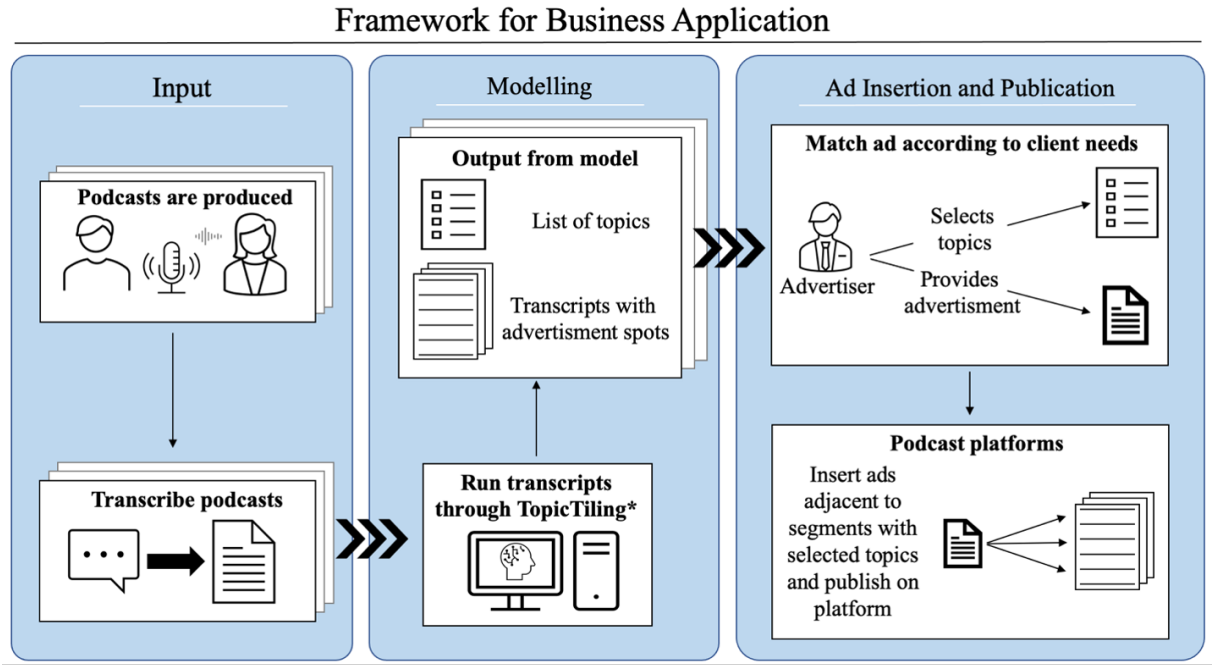

Figure 6.1: *The business framework describes how the modelling framework proposed in this study (called TopicTiling*) could be used by a podcast platform.*

attracts more advertisers, content creators, and listeners which creates a virtuous cycle of network effects which contributes to the platform's growth and improves its value proposition (Katz & Shapiro, 1985; Van Alstyne et al., 2016, Bresnahan, 1998). Additionally, the use of AI supercharges these effects by rapidly refining ad placement and targeting, accelerating growth, and reinforcing competitive advantages (Iansiti & Lakhani, 2020).

Furthermore, the implications of the study translate into the concepts of native advertisement (Campbell & Marks, 2015) and advertising as a monetization strategy of platforms (Goldfarb & Tucker, 2011). The podcasts are segmented at topical shifts which minimise the disruption of the content (Ritter & Cho, 2009) and match advertisements to segments with related content making it blend into the listening experience and enhance the effectiveness of the advertisement itself (van Reijmersdal et al., 2009). This technology can in the future make native advertisement in podcasts scalable since human annotation of its positioning is reduced. Additionally, brokering advertisements for podcasts can be priced by the platform, effectively adding a stream of revenue.

The result of this study suggests another approach to Dynamic Ad Insertion than the one proposed by (Leung, 2020; Spotify, 2022). The main difference is that TopicTiling* not only enables native ad placement but can also find an optimal location for the advertisement within the platform's catalogue of podcasts. This removes the need for creators to supply advertisement spots within their content, which is the proposed positioning methodology of (Leung, 2020; Spotify, 2022). The positioning methodology of TopicTiling* can also be adopted by (Leung, 2020; Spotify, 2022) as a way for positioning an advertisement which is tailored to the user rather than the content surrounding it. Additionally, TopicTiling* can be used by new platform companies to enable DAI since the collection of user data is not required. This lowers the barriers of entry into the market.

This study has implications for research as it expands on one of the more applied ap-

proaches to text segmentation and develops a new framework that leverages recent development in the NLP field. Using a transformer-based clustering model as the foundation for topic extraction enables text segmentation on data compiled from sources with a lot of noise and non-uniform language where a BoW approach may fail. This study has showcased the effectiveness of these techniques by applying them to the Spotify Podcast Dataset which includes common traits of spoken language from multi-speaker contexts such as repetition, slang, dialects, interruptions etc.

As the podcast medium presents its own set of complexities, including the need of transcribing spoken language, the possible implications of this study can be translated beyond the medium and into other forms of digital media. For example, segmentation of news articles on online platforms where extraneous content needs to be inserted in optimal locations.

6.3 Limitations

In this section, the limitations of this study will be raised and elaborated on. The following subsections will detail the observed limitations pertaining to the approach, dataset and preprocessing, modelling, and evaluation steps.

6.3.1 Two-step Architecture of the Modelling Framework

The two-step approach employed in this study, involving topic modelling and text segmentation, presents an inherent trade-off and conflict between the two components and is further amplified by the complexities in transcribed conversational data.

It can be argued that topic modelling in this study's context would benefit from fully leveraging the attention mechanisms of transformers and especially long-sequence transformers (Beltagy et al., 2020; Zaheer et al., 2020). They could have been advantageous as they could capture the semantic relationships in the data despite the noise and quality issues arising from the transcription process. On the other hand, text segmentation requires a more granular unit of analysis, i.e., sentences (Riedl & Biemann, 2012). As the objective is to accurately identify topic boundaries in the text and segment them accordingly, the most suitable unit of analysis is sentences and requires each sentence to have a topic probability density vector.

This trade-off between the two components highlights the challenges in designing a two-step architecture that can simultaneously cater to both objectives without forfeiting the performance of either component. The chosen methodology has proved advantageous for text segmentation but may have limited the potential of the topic modelling component to fully capture the topics present. Possible avenues for future research could explore alternative approaches to address this trade-off, such as training the topic model with long-sequence transformers or with a different segmentation method which could cater to the objectives of both components.

6.3.2 Dataset and Preprocessing

As access to the Spotify Podcast Dataset was the starting point of the formation of this study the inherent opportunities and challenges in the dataset have translated into some of the limitations seen in this study. The challenges, outside of the innate nature of the medium, are seen in the transcription of the podcasts. The Google Speech-to-Text API provides a word-for-word transcription of the data with varying confidence and with no regard to semantic context or grammatical structure. This introduced additional noise to the data and complexity to the choices in subsequent steps. The study could have benefited from a cleaner and curated dataset which would have freed the approach from the mentioned limitations.

As described in section 4.5.2, the dataset was filtered to only include transcripts from the sports category. The choice to limit the dataset stems from the computational constraints observed with the significant increase in data points when segmenting the transcripts into documents of smaller size. The filtering process may have introduced challenges for the downstream tasks, as including data from all categories may have improved the framework's ability to identify a more diverse set of topics which would help it make accurate predictions on the sentence level.

6.3.3 Modelling

The modelling used in this study is to a large extent based on non-parametric models such as t-SNE and HDBSCAN which have the inherent property that they cannot be used for large-scale predictions of unseen data as the reference embeddings and clusters are dependent on all data points (van den Maarten & Hinton, 2008; Campello, Moulavi & Sander, 2013). Therefore, the modelling approach presented is dependent on re-training after some time, meaning that the cost of computation in the long term is elevated. This also proposes a limitation to the business framework as new topic representations will evolve after re-training which implies that old advertisers must choose new topics to target their advertising.

Furthermore, the search for an optimal modelling framework configuration has been limited. The results presented in section 5.1.1 suggest a negligible impact on the WindowDiff metric across the framework configurations which were tested. The choice of embeddings has formed the subsequent steps of the modelling framework. First, as only one embedding model was used for this study, it has limited its ability to explore potential advantages across different transformers as discussed in section 6.3.1. Secondly, the dimensionality reduction algorithm suggested to be most suitable for the purposes of this study were UMAP, as seen in both the theory section and in the implementation by Grootendorst (2022). Due to challenges with memory complexity when attempting to apply UMAP, we resorted to using t-SNE instead. Third, the parameter space of m_{pts} in the cluster model would ideally have been assessed exhaustively. Future efforts can be put towards exploring these modelling limitations. The results of section 5.1.3 should guide such an effort.

6.3.4 Evaluation

The evaluation methodology of this study is based on a standard evaluation metric which is used for text segmentation research. Since the data was unlabelled, an effort was made to label some of the transcripts with segment breaks. However, due to time limitations, we were unable to complete the labelling of more than 20 transcripts which served as our evaluation set. Sample noise is an inherent risk with having a small evaluation dataset which refers to the data not being representative of the population which affects the generalizability of the results (Géron, 2019). This way, the results may reflect performance which is only generalizable to a small subset of the population. Conversely, the results may also display a much worse performance than what the approach would score if the evaluation data were representative of the underlying population. Selection of which transcripts to annotate was performed completely at random using the random function from the NumPy[1] Python library to mitigate this risk.

The result of this study serves as a proof-of-concept of the modelling framework. However, the configuration of the modelling framework as explored in this study may not be precise enough to enable complete autonomy of such a system since that would require an even lower WindowDiff score. An argument can be made that there is a mismatch between the evaluation metric WindowDiff and the demands of a podcast listener. The WindowDiff metric gives a better score for close matches than for complete misses by scoring all boundaries within the distance k of an actual boundary proportional to the miss. Even though this is a valid argument theoretically, it may not be in practice. Placing an advertisement in a podcast even one sentence off from the actual boundary may result in a distorted context of the conversation, making the listener miss the end of a story or the finishing sentence of an argument which results in a poor listening experience. To mitigate this limitation, we have selected a small k in this study, but efforts remain to limit k to 1.

6.4 Future Research

The modularity and interconnectedness of the components in our suggested modelling framework introduce a large hyperparameter space. Due to time and computational constraints, a limited scope of configurations is evaluated in this study. Thus, future research could evaluate a larger space and assess the proposed methodology in a more rigorous manner. The results of section 5.1.3 suggest that the hyperparameters should be optimized for a model which utilizes a large range of cosine similarities and a high average depth score proportional to the cosine similarity range. This may result in a model configuration that can enable the automatic segmentation of podcast transcripts. Additionally, more experimentation of the inner components of TopicTiling* should be explored in future iterations. This could include a closer examination of the depth score smoothing applied in this study and interactions between hyperparameters.

Additionally, future research can explore the comparative performance of TopicTiling* to other text segmentation techniques reviewed in this study and beyond. In such a study,

[1] https://numpy.org/doc/stable/

a suggestion is to compare the performance of TopicTiling* against TopicTiling, LcSeg and the SECTOR frameworks on standardised and annotated datasets such as the Choi Dataset (Choi, 2000) and Galley's WSJ Dataset (Galley et. al., 2003). The models should also be scored on more evaluation metrics such as the Pk score and on WindowDiff with k set to a range of values. From such a study, the generalizability of the performance of TopicTiling* could be concluded.

7 Conclusion

This study set out to investigate how data science can be leveraged to place relevant advertisements in podcasts. This objective was broken down into two sub-questions aiming to investigate (1) how data science methods can locate topical shifts in podcast transcripts, and (2) how data science methods can find topics that are meaningful for podcast advertisement. Building on previous research within topic modelling and text segmentation and by using the Spotify Podcast Dataset, we have suggested a modelling framework for solving the problems formulated by the research questions. The framework incorporates state-of-the-art NLP methodologies to combat the inherent challenges stemming from transcribed spoken language data and the nature of the medium as examined in the literature review. The proposed business framework enables native advertisement placement in podcasts in a scalable and data-driven manner. This reinforces aspects of network effects, platform dynamics and monetization strategies by leveraging algorithms and data to enhance platforms' value proposition to listeners, creators, and advertisers. Building data-driven features which are adaptable to a changing environment (by re-training the cluster model periodically) is a way for platforms to develop digital capabilities and strengthen competitive advantages.

After evaluating the proposed modelling framework, we can conclude that it has served as a proof-of-concept for placing relevant advertisements in podcasts. The results showed that applying TopicTiling* to podcast transcripts locates meaningful advertisement spots. Additionally, assigning meaningful topics to the predicted segments can be done, but the methodology would benefit from supplementary exploration. Efforts remain to find an optimal configuration of the modelling framework. Future research should attempt to optimise it to enable the business framework to a degree where it can be part of the offering of podcast platforms.

8 Bibliography

Abuzayed, A., & Al-Khalifa, H. (2021). BERT for Arabic topic modeling: An experimental study on BERTopic technique. Procedia computer science, 189, 191-194.

Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record, 28(2), 49-60.

Arnold, S., Schneider, R., Cudré-Mauroux, P., Gers, F. A., & Löser, A. (2019). Sector: A neural model for coherent topic segmentation and classification. Transactions of the Association for Computational Linguistics, 7, 169-184.

Athey, S., Calvano, E., & Gans, J. S. (2018). The impact of consumer multi-homing on advertising markets and media competition. Management science, 64(4), 1574-1590.

Beeferman, D., Berger, A., & Lafferty, J. (1999). Statistical models for text segmentation. Machine learning, 34, 177-210.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., & Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. Computational statistics & data analysis, 52(1), 155-173.

Berry, R. (2006). Will the iPod kill the radio star? Profiling podcasting as radio. Convergence, 12(2), 143-162.

Berry, R. (2016). Part of the establishment: Reflecting on 10 years of podcasting as an audio medium. Convergence, 22(6), 661-671.

Bezbaruah, S., & Brahmbhatt, K. (2023). Are podcast advertisements effective? An emerging economy perspective. Journal of International Consumer Marketing, 35(2), 215-233.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

Bonini Baldini, T. (2015). The 'second age'of podcasting: Reframing podcasting as a new digital mass medium. Quaderns del CAC, 41, 23-33.

Boudreau, K. (2010). Open platform strategies and innovation: Granting access vs.

devolving control. Management science, 56(10), 1849-1872.

Bresnahan, T. F. (1999). New modes of competition: Implications for the future structure of the computer industry. In Competition, Innovation and the Microsoft Monopoly: Antitrust in the Digital Marketplace: Proceedings of a conference held by The Progress & Freedom Foundation in Washington, DC February 5, 1998 (pp. 155-208). Springer Netherlands.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

Caillaud, B., & Jullien, B. (2003). Chicken & egg: Competition among intermediation service providers. RAND journal of Economics, 309-328.

Campbell, C., & Marks, L. J. (2015). Good native advertising isn'ta secret. Business horizons, 58(6), 599-606.

Campello, R. J., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17 (pp. 160-172). Springer Berlin Heidelberg.

Canavan, A., Graff, D., and Zipperlen, G., (1997) CALLHOME American English Speech LDC97S42. Web Download. Philadelphia: Linguistic Data Consortium. Available at: https://catalog.ldc.upenn.edu/LDC97S42

Cennamo, C., & Santalo, J. (2013). Platform competition: Strategic trade-offs in platform markets. Strategic management journal, 34(11), 1331-1350.

Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. Computer Speech & Language, 13(4), 359-394.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Choi, F. Y. (2000). Advances in domain independent linear text segmentation. arXiv preprint cs/0003083.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Fiedel, N. (2022). Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311.

Clifton, A., Reddy, S., Yu, Y., Pappu, A., Rezapour, R., Bonab, H., ... & Jones, R. (2020, December). 100,000 podcasts: A spoken English document corpus. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 5903-5917).

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (2017, August). Scatter/gather: A cluster-based approach to browsing large document collections. In ACM SIGIR Forum (Vol. 51, No. 2, pp. 148-159). New York, NY, USA: ACM.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860.

Davenport, T. H., Barth, P., & Bean, R. (2012). How'big data'is different.

de Groot, M., Aliannejadi, M., & Haas, M. R. (2022). Experiments on Generalizability of BERTopic on Multi-Domain Short Text. arXiv preprint arXiv:2212.08459.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science, 41(6), 391-407.

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

Du Bois, J. W., and Englebretson, R. (2005) Santa Barbara Corpus of Spoken American English Part IV LDC2005S25. Web Download. Philadelphia: Linguistic Data Consortium. Available at: https://catalog.ldc.upenn.edu/LDC2005S25

Egger, R., & Yu, J. (2022). A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. Frontiers in sociology, 7.

Eisenmann, T., Parker, G., & Van Alstyne, M. W. (2006). Strategies for two-sided markets. Harvard business review, 84(10), 92.

Eisenstein, J. (2013, June). What to do about bad language on the internet. In Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies (pp. 359-369).

Eisenstein, J., & Barzilay, R. (2008, October). Bayesian unsupervised topic segmentation. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (pp. 334-343).

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

Evans, D. S., & Schmalensee, R. (2016). Matchmakers: The new economics of multisided platforms. Harvard Business Review Press.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. Annals of eugenics, 7(2), 179-188.

Galley, M., McKeown, K., Fosler-Lussier, E., & Jing, H. (2003, July). Discourse segmentation of multi-party conversation. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (pp. 562-569).

Geron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, ISBN: 978-1492032649.

Gisbrecht, A., Schulz, A., & Hammer, B. (2015). Parametric nonlinear dimensionality reduction using kernel t-SNE. Neurocomputing, 147, 71-82.

Goldfarb, A., & Tucker, C. (2011). Online display advertising: Targeting and obtrusiveness. Marketing Science, 30(3), 389-404.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. Computational linguistics, 27(2), 153-198.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:2203.05794.

Hagiu, A. (2009). Two-sided platforms: Product variety and pricing structures. Journal of Economics & Management Strategy, 18(4), 1011-1043.

Hagiu, A., & Wright, J. (2015). Multi-sided platforms. International Journal of Industrial Organization, 43, 162-174.

Hasebe, Y. (2015) Design and implementation of an online corpus of presentation transcripts of ted talks. Procedia-Social and Behavioral Sciences, 198:174–182. Available at: `https://www.researchgate.net/publication/282554079_Design_and_Implementation_of_an_Online_Corpus_of_Presentation_Transcripts_of_TED_Talks`

Hearst, M. A. (1994). Multi-paragraph segmentation of expository text. arXiv preprint cmp-lg/9406037.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780. Hocking, J.G. & Young, G. S. (1988) Topology. Dover. ISBN 0-486-65676-4. Iansiti, M., & Lakhani, K. R. (2020). Competing in the age of AI: Strategy and leadership when algorithms and networks run the world. Harvard Business Press.

Jalilifard, A., Caridá, V. F., Mansano, A. F., Cristo, R. S., & da Fonseca, F. P. C. (2021). Semantic sensitive TF-IDF to determine word relevance in documents. In Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 2 (pp. 327-337). Singapore: Springer Singapore.

Joliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Phil. Trans. R. Soc. A, 374, 20150202.

Katz, M. L., & Shapiro, C. (1985). Network externalities, competition, and compatibility. The American economic review, 75(3), 424-440.

Katz, M. L., & Shapiro, C. (1985). Network externalities, competition, and compatibility. The American economic review, 75(3), 424-440.

Lambrecht, A., & Tucker, C. (2013). When does retargeting work? Information specificity in online advertising. Journal of Marketing research, 50(5), 561-576.

LaValle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2010). Big data,

analytics and the path from insights to value. MIT Sloan management review.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788-791.

Leung, A. (2020). Rewriting the playbook for podcast advertising. https://ads.spotify.com/en-US/news-and-insights/streaming-ad-insertion-podcast-advertising/

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

MacQueen, J. (1967, June). Classification and analysis of multivariate observations. In 5th Berkeley Symp. Math. Statist. Probability (pp. 281-297). Los Angeles LA USA: University of California.

Manning, C. D. (2009). An introduction to information retrieval. Cambridge university press.

Manning, C., & Schutze, H. (1999). Foundations of statistical natural language processing. MIT press.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Hung Byers, A. (2011). Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute.

McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data: the management revolution. Harvard business review, 90(10), 60-68

McHugh, S. (2016). How podcasting is changing the audio storytelling genre. The radio journal–international studies in broadcast & audio media, 14(1), 65-82.

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

McIntyre, D. P., & Chintakananda, A. (2014). Competing in network markets: Can the winner take all?. Business Horizons, 57(1), 117-125.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(1), 86-97. Napoli, P. M. (2011). Audience evolution: New technologies and the transformation of media audiences. Columbia University Press.

Parker, G. G., Van Alstyne, M. W., & Choudary, S. P. (2016). Platform revolution: How networked markets are transforming the economy and how to make them work for you. WW Norton & Company.

Pearson, K. (1901). Principal components analysis. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 6(2), 559.

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (pp. 2227-2237).

Pevzner, L., & Hearst, M. A. (2002). A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics, 28(1), 19-36.

Pezzotti, N., Lelieveldt, B. P., Van Der Maaten, L., Höllt, T., Eisemann, E., & Vilanova, A. (2016). Approximated and user steerable tSNE for progressive visual analytics. IEEE transactions on visualization and computer graphics, 23(7), 1739-1752.

Poličar, P. G., Stražar, M., & Zupan, B. (2019). openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. BioRxiv, 731877. Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. Big data, 1(1), 51-59.

Pustejovsky, J. (1998). The generative lexicon. MIT press.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Riedl, M., & Biemann, C. (2012, July). TopicTiling: a text segmentation algorithm based on LDA. In Proceedings of ACL 2012 student research workshop (pp. 37-42).

Ritter, E. A., & Cho, C. H. (2009). Effects of ad placement and type on consumer responses to podcast ads. Cyberpsychology & Behavior, 12(5), 533-537.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. Journal of documentation.

Rochet, J. C., & Tirole, J. (2003). Platform competition in two-sided markets. Journal of the european economic association, 1(4), 990-1029.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20, 53-65.

Saunders, M., Lewis, P., & Thornhill, A. (2009). Research methods for business students. Pearson education.

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words

with subword units. arXiv preprint arXiv:1508.07909.

Shearer, C. (2000). The CRISP-DM model: the new blueprint for data mining. Journal of data warehousing, 5(4), 13-22.

Sia, S., Dalmia, A., & Mielke, S. J. (2020). Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too!. arXiv preprint arXiv:2004.14914.

sklearn (2023) Using stop words, Available at: https://scikit-learn.org/stable/modules/feature_extraction.html#stop-words

Spotify (2021). Taking podcast advertising to the next level. https://ads.spotify.com/en-GB/news-and-insights/2021-podcast-ads-announcements/ Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques.

Subakti, A., Murfi, H., & Hariadi, N. (2022). The performance of BERT as data representation of text clustering. Journal of big Data, 9(1), 1-21.

Sullivan, J. L. (2019). The platforms of podcasting: Past and present. Social media+ society, 5(4), 2056305119880002.

Van Alstyne, M. W., Parker, G. G., & Choudary, S. P. (2016). Pipelines, platforms, and the new rules of strategy. Harvard business review, 94(4), 54-62.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).

Van Reijmersdal, E., Neijens, P., & Smit, E. G. (2009). A new branch of advertising: Reviewing factors that influence reactions to product placement. Journal of advertising research, 49(4), 429-449.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. C. J. (2019). Evaluating word embedding models: Methods and experimental results. APSIPA transactions on signal and information processing, 8, e19.

Whitner, G. (2023). The meteoric rise of podcasting. Retrieved April 2023, from https://musicoomph.com/podcast-statistics

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on neural networks, 16(3), 645-678.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., ... & Ahmed, A. (2020). Big bird: Transformers for longer sequences. Advances in neural information processing systems, 33, 17283-17297

Appendix

A TopicTiling* on Synthetic Data

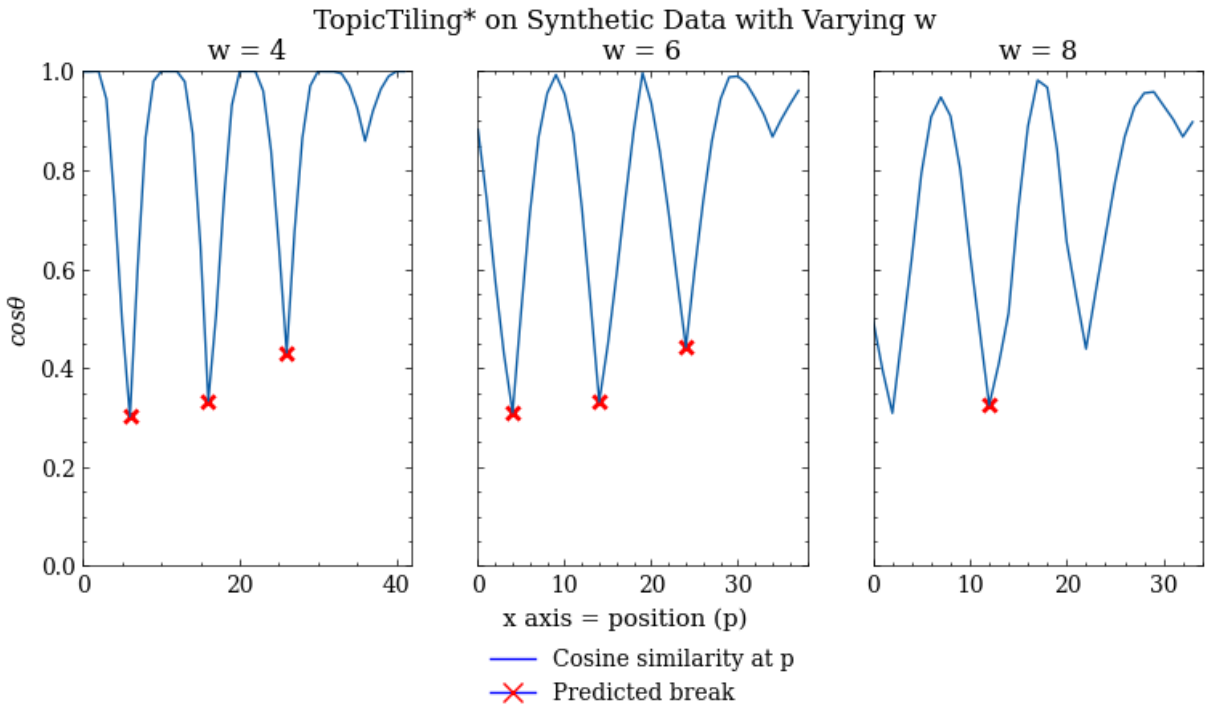

Illustration of segmentation using our implementation of TopicTiling with synthetic data generated randomly from a Dirichlet distribution with uniform noise using the Python library NumPy. Hypothetical segment boundaries are inserted at position sentences 10, 20, 30 and 40 in the synthetic data seen as the sharp decreases in the plots. Hyperparameters of TopicTiling* are set as follows: w set to $\{4, 6, 8\}$, τ set to $1e - 20$, and n set to 3. We observe the different shapes of the cosine similarity score in blue as w changes. For the first two plots, we see that the first three boundaries are found while the fourth is not considered a boundary by the algorithm since the decrease is too shallow. This is an effect of the depth score calculation as described in section 4.8.2. For the third plot where w is set to 8, we observe that the first decrease is not labelled a boundary. This is a result of the shallow drop on the left-hand side caused by the large size window relative to the position of the first boundary. Thus, the cosine similarity is already impacted by the boundary at position 1 which is not true when $w = 4$. The threshold for boundaries is thereby impacted and affects the third drop as well which does not get predicted as a boundary.

B Stop Words

a	been	eight	haven	made	one	someone	top
about	before	either	having	many	only	something	toward
above	beforehand	eleven	he	may	onto	sometime	towards
across	behind	else	hence	me	or	sometimes	twelve
after	being	elsewhere	her	meanwhile	other	somewhere	twenty
afterwards	below	empty	here	might	others	still	two
again	beside	enough	hereafter	mightn	otherwise	such	un
against	besides	etc	hereby	mill	our	system	under
ain	between	even	herein	mine	ours	t	until
all	beyond	ever	hereupon	more	ourselves	take	up
almost	bill	every	hers	moreover	out	ten	upon
alone	both	everyone	herself	most	over	than	us
along	bottom	everything	him	mostly	own	that	ve
already	but	everywhere	himself	move	part	the	very
also	by	except	his	much	per	their	via
although	call	few	how	must	perhaps	theirs	was
always	can	fifteen	however	mustn	please	them	wasn
am	cannot	fify	hundred	my	put	themselves	we
among	cant	fill	i	myself	rather	then	well
amongst	co	find	ie	name	re	thence	were
amoungst	con	fire	if	namely	s	there	weren
amount	could	first	in	needn	same	thereafter	what
an	couldn	five	inc	neither	see	thereby	whatever
and	couldnt	for	indeed	never	seem	therefore	when
another	cry	former	interest	nevertheless	seemed	therein	whence
any	d	formerly	into	next	seeming	thereupon	whenever
anyhow	de	forty	is	nine	seems	these	where
anyone	describe	found	isn	no	serious	they	whereafter
anything	detail	four	it	nobody	several	thick	whereas
anyway	did	from	its	none	shan	thin	whereby
anywhere	didn	front	itself	noone	she	third	wherein
are	do	full	just	nor	should	this	whereupon
aren	does	further	keep	not	shouldn	those	wherever
around	doesn	get	last	nothing	show	though	whether
as	doing	give	latter	now	side	three	which
at	don	go	latterly	nowhere	since	through	while
back	done	had	least	o	sincere	throughout	whither
be	down	hadn	less	of	six	thru	who
became	due	has	ll	off	sixty	thus	whoever
because	during	hasn	ltd	often	so	to	whole
become	each	hasnt	m	on	some	together	whom
becomes	eg	have	ma	once	somehow	too	whose

Table 1: Stop Words from sklearn

C Code

GitHub repository link for all code that has been used in this thesis.
`https://github.com/OskarMunck/Thesis`