TU WIEN Informatics

Exploration of Content-Based Cross-Domain Podcast Recommender Systems

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Matthias Hofmaier, BSc
Matrikelnummer 11944050

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Assistant Prof. Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Mitwirkung: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Wien, 7. Oktober 2024

Matthias Hofmaier	Julia Neidhardt

TU WIEN Informatics

Exploration of Content-Based Cross-Domain Podcast Recommender Systems

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Matthias Hofmaier, BSc
Registration Number 11944050

to the Faculty of Informatics

at the TU Wien

Advisor: Assistant Prof. Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Assistance: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Vienna, 7th October, 2024

Matthias Hofmaier	Julia Neidhardt

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Matthias Hofmaier, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 7. Oktober 2024

Matthias Hofmaier

Danksagung

Zuerst möchte ich mich ganz besonders bei Dr.in Julia Neidhardt und Thomas E. Kolb vom Christian Doppler Labor für Recommender System bedanken. Mir ist bewusst, dass die Betreuung meiner Diplomarbeit für Euch oft mit viel Arbeit verbunden war, die dann meist kurzfristig erledigt werden musste. Euer offenes Ohr und euer konstruktives Feedback zu meiner Arbeit haben wesentlich zur Fertigstellung beigetragen und ich weiß dies sehr zu schätzen.

Mein Dank gilt auch der FALTER Verlagsgesellschaft mbH für die Bereitstellung der Podcast- und News-Artikel-Daten sowie für die Teilnahme am qualitativen Experteninterview.

Mein Dank gilt auch den Teilnehmenden der Annotationsstudie, Sophie, Nina und Mara. Ich bin mir bewusst, dass die Durchführung eine anstrengende Aufgabe war und weiß Eure Unterstützung mehr als zu schätzen.

Ich danke auch meinen Eltern, die mich während meines Studiums immer unterstützt haben und mir schließlich die Möglichkeit gegeben haben, diese Arbeit abzuschließen.

Zuletzt möchte ich mich ausdrücklich bei Mara bedanken. Ich weiß, dass die Zeit der Erstellung dieser Arbeit auch für Dich nicht einfach war. Ohne Deine stetige Unterstützung und Deine aufbauenden Worte, bin ich mir sicher, dass diese Arbeit nicht in dieser Form fertiggestellt worden wäre.

Acknowledgements

First of all, I would like to thank Dr.in Julia Neidhardt and Thomas E. Kolb from the Christian Doppler Laboratory for Recommender Systems. I am aware that supervising my thesis often involved a lot of work for you, most of which had to be completed at short notice. Your open ear and constructive feedback on my thesis contributed significantly to its completion and I greatly appreciate this.

I would also like to thank FALTER Verlagsgesellschaft mbH for providing the podcast and news article data and for participating in the qualitative expert interview.

My thanks also go to the participants in the annotation study, Sophie, Nina and Mara. I am aware that conducting this was a difficult task and I greatly appreciate your support.

My thanks also go to my parents, who have always supported me during my studies and finally gave me the opportunity to complete this thesis.

Lastly, I would like to specifically thank Mara. I know that the time of writing this thesis was not easy for you either. Without your constant support and encouraging words, I am sure that this thesis would not have been completed in this form.

Kurzfassung

Podcasts haben sich im letzten Jahrzehnt zu einem beliebten Medium entwickelt. Die riesige Menge an verfügbaren Daten motiviert die Forschung zu Podcast-Empfehlungssystemen, um diese Daten den Nutzern zugänglich zu machen. Da interaktionsbasierte Datensätze für Podcasts nur den großen Streaming-Anbietern zur Verfügung stehen, werden inhaltsbasierte Methoden benötigt, um ein Empfehlungssystem aufzubauen. Die Entwicklung inhaltsbasierter Empfehlungssysteme ist eng mit dem Bereich des Information Retrieval verbunden, der im Podcast-Bereich gut untersucht ist. Die meisten dieser Forschungsarbeiten befassen sich jedoch mit dem Retrieval auf der Grundlage der Transkription der Audiodatei und vergleichen nicht die Effektivität anderer Darstellungen, was eine Forschungslücke darstellt. Podcasts werden oft als das auditive Gegenstück zu textuellen Medien wie Nachrichtenartikeln bezeichnet, und die Verwendung von Transkriptionen verbindet diese verschiedenen Medientypen auch in der Art und Weise, wie ihr Inhalt dargestellt wird. Die Ähnlichkeit der Medien motiviert die Forschung zu domänenübergreifenden Empfehlungssystemen, die darauf abzielen, Informationen aus einer Quelldomäne, wie z. B. Podcasts, zu nutzen, um Empfehlungen in anderen Domänen zu generieren. In der Forschung wurde jedoch noch kein derartiges System für den Bereich Podcasts veröffentlicht. Um diese Lücken zu schließen, untersuchen wir, wie ein inhaltsbasiertes, domänenübergreifendes Empfehlungssystem zwischen Podcasts und Nachrichtenartikeln aufgebaut und evaluiert werden kann, ohne dass interaktionsbasierte Daten verfügbar sind. Darüber hinaus untersuchen wir wie sich verschiedene Attribute, die zur Darstellung von Podcasts in einem Empfehlungssystem verwendet werden, auf die Leistung des Systems auswirken. Dies geschieht durch die Erstellung eines manuell annotierten Datensatzes zwischen Podcast-Segmenten und Nachrichtenartikeln. Unter Verwendung dieses Datensatzes berechnen wir mehrere Modelle, die jeweils unterschiedliche Podcast-Darstellungen verwenden, mit dem Ziel, Empfehlungen für Nachrichtenartikel anhand eines bestimmten Podcast-Segments zu generieren. Bei der Evaluierung der Ranking-Qualität und der thematischen Vielfalt stellen wir fest, dass unser Ansatz vier verschiedene Basismodelle in Bezug auf die Ranking-Qualität übertrifft. Wir stellen jedoch auch fest, dass diese Steigerung der Ranking-Qualität auf Kosten der Empfehlungsvielfalt geht. Außerdem beobachten wir entgegen unserer vorherigen Annahmen nicht eine bestimmte Gruppe von Podcast-Merkmalen, die alle anderen übertrifft, aber große Unterschiede zwischen verschiedenen Podcast-Shows. Dies motiviert zu einer tieferen Untersuchung der Eigenschaften von Shows, die diese Unterschiede erklären könnten.

Abstract

Podcasts have become a popular medium in the last decade. The huge amount of data available motivates research on podcast recommender systems to make this data accessible to users. Since interaction-based datasets for podcasts are only available to the large streaming providers, content-based methods are needed to build a recommender system. Building content-based recommender systems is closely related to the field of information retrieval, which is well studied in the podcast domain. However, most of this research examines retrieval based on the textual transcription of the audio file and does not compare the effectiveness of other representations such as metadata or audio features, which represents a gap in the research. Podcasts are often referred to as the auditory counterpart of textual media such as news articles, and using transcriptions also connects these different types of media in the way their content is represented. This similarity in media content motivates research on cross-domain recommender systems, which aim to use information from one source domain, such as podcasts, to generate recommendations in other domains. However, no such system in the podcast domain has been published in research. To address these research gaps, we investigate how to build and evaluate a content-based cross-domain recommender system between podcasts and news articles without the availability of interaction-based data. Furthermore, in this work, we investigate how different attributes used to represent podcasts in a recommender system scenario affect the performance of the system. This is done by creating a manually annotated cross-domain dataset between podcast segments and news articles. Using this dataset, we fit several models, each using a different set of podcast representations, with the goal of generating news article recommendations given a particular podcast segment. Performing an evaluation in terms of ranking quality measures and the beyond-accuracy measure of topical diversity, we find that our approach outperforms four different baseline models in terms of ranking quality. However, we also find that this increase in ranking quality comes at the expense of recommendation diversity. Moreover, contrary to our prior beliefs, we do not observe a particular set of podcast features that outperform all others, but rather a large difference in performance between different podcast shows, which motivates further investigation into the properties of shows that might explain these differences.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Over the past decade, podcasts have become increasingly popular. Podcasts are often referred to as the auditory counterpart to online blogs, and producing a podcast only requires a smartphone and a piece of software such as Anchor or Podbean that can record and edit the podcast [XPX+16, JZS+21]. With the advent of audio streaming platforms like Spotify, Amazon Music or Apple Music, it has also become easy for producers to distribute their content to a wide range of listeners.

Because of the ease of production, there is a huge number of podcasts available in various forms and genres. According to Listen Notes, as of May 2024 [lis], there are over three million podcast shows with over 190 million episodes available worldwide. By comparison, IMDB only lists less than 10 million total movie and TV show titles [imd]. This huge number of available podcasts and podcast listeners also translates into a profitable business. The Business Research Company reports that the podcast market will be worth $27.73 billion in 2023 and is expected to grow to a market size of $104.97 billion by 2028. This represents a compound annual growth rate of 30.1% [pod]. The outlook for the podcast business market and the sheer volume of data available is motivating the development of intelligent recommender systems to make this content accessible to users.

Because podcasts have only emerged in the last few years, not much research has been published in the area of podcast recommender systems. The three main approaches that can be found in the literature are based on *collaborative filtering* [GAR08, BFWC20], *sequential recommendations* [BFWC20], or *content-based solutions* [XPX+16, RA23]. The challenge with the first two approaches is the availability of the data needed to implement such systems. In collaborative filtering approaches, recommendations are made by finding similar *users* or *items* based on their historical tastes. To quantify the

similarity in taste between two entities, the interaction history between users and items is used [RRS22]. Sequential recommender systems, on the other hand, use short-term sequences of interactions between items, e.g. songs in a listening session that are played one after the other, to recommend new items to a user [RRS22]. Both methods require *interaction data* between users and items, and while open datasets exist for domains such as movies or books, no such dataset exists for the podcast domain. This fact makes it difficult to develop interaction data dependent systems outside of the major streaming providers.

In content-based approaches, on the other hand, the simplest form of recommender systems can be built completely without requiring interaction data between users and items. For example, in [XPX+16], the authors propose a two-step content-based podcast recommendation system in which they use a latent representation of the textual metadata associated with a podcast (title, description, tags, etc.). They classify the representation into a category and then use the predicted category along with numeric metadata (upload date, duration, plays) to train a model that predicts the *affinity* of a user profile and a podcast. Since they have no interaction data, but know which user uploaded the podcast, they use that information as a proxy for *user affinity*. This means that their model does not actually predict whether a user likes a particular item or not, but rather whether they uploaded it to the podcast platform [XPX+16].

It is also possible to exclude the personalization aspect of a content-based recommender system and view it as an information retrieval task, where a query item is used to retrieve relevant recommended items in response. This is done by Mizuno et al. in [MOG08]. Here, the authors extract relevant keywords from podcast titles using a modified version of the term frequency-inverse document frequency (TF-IDF). This statistic measures how important certain words are in a document. The TF-IDF statistic of a podcast transcription is then used to retrieve similar episodes by measuring the cosine similarity of encoded source and target items.

Information retrieval techniques in the podcast domain are generally better studied than recommender systems. An example of this is the segment retrieval task in the podcast track that was part of the Text REtrieval Conference (TREC) in 2020 [JCC+20] and 2021 [JCC+21]. In both years, one of the main goals was to retrieve two-minute segments of podcast transcriptions that are relevant to a given query. Solutions to this task included generating candidate segments using *BM25* and *re-ranking* through more sophisticated models, or dense retrieval approaches that directly retrieved the relevant segments [JCC+20, JCC+21]. Both of these methods may be of particular interest to this research and will be explored further in this thesis. Some solutions for TREC 2021 [JCC+21] also include non-textual features such as audio features of podcasts in their content-based retrieval systems, which is also of interest to us. Due to the heterogeneity of podcast content and the vast amount of possibilities for representation, the investigation of podcast representation methods will be an essential part of this research work.

Due to the similarity of spoken word content in podcasts and text in news articles, *cross-domain* methods are of particular interest for research in podcast recommender systems.

In contrast to traditional recommender systems that recommend items within a domain (e.g., news article to news article or podcast to podcast), cross-domain recommender systems aim to use information from one source domain to make recommendations to other domains (e.g., news article to podcast or vice versa) [RRS22]. While a cross-domain recommender system has been successfully used to overcome the *cold-start problem* of new users by leveraging music taste in [NCP+20], to our knowledge no research has been done on how to leverage a cross-domain recommender system between podcasts and textual media. This is also suggested in [JZS+21]. Therefore, we want to investigate this question within this research.

1.2 Problem Statement

Due to the lack of an interaction dataset between *users* and *items* pairs with a specific relevance signal, collaborative filtering and sequence-based recommender system approaches cannot be applied to the podcast domain. Therefore, content-based approaches using information retrieval techniques as in [XPX+16, MOG08, JCC+20, JCC+21] are needed to build a podcast recommender system. Each of these approaches uses different sets of podcast representations to accomplish these tasks. To our knowledge, there has been no evaluation of these representations, making it difficult for researchers to choose appropriate representations when building their own systems. The lack of evaluation of representations therefore represents a clear research gap.

Although content-based recommender systems can be built without an interaction dataset, it is still necessary to quantitatively evaluate a solution after it has been implemented. Since no such dataset is available for the podcast domain, and especially for the cross-domain case between podcasts and news articles, this is also a clear research gap. Furthermore, all existing content-based podcast recommender systems focus on making recommendations within the podcast domain. While cross-domain approaches have been well studied in other recommendation scenarios, such as [ZCW+19, ZWC+20], a content-based cross-domain recommender system between podcasts and other textual media has not yet been investigated. This represents another gap in research.

1.3 Aim of the Work

The aim of this work is to explore how *content-based cross-domain podcast recommender systems* can be built and evaluated. The first issue we want to investigate is how to efficiently represent podcasts in recommender system scenarios. As motivated before, there are several possibilities, including metadata, transcriptions as well as audio features. However, no evaluation of these different representation methods has been published in the literature. The second topic we want to investigate is the recommendation quality of a cross-domain recommender system between podcasts and news articles compared to baseline recommender systems. Here, recommendation quality refers to ranking quality

metrics such as *Mean Reciprocal Rank (MRR)* and *Normalized Discounted Cumulative Gain (NDCG)*, as well as *beyond-accuracy measures* such as *diversity*.

From the aforementioned aims of this research work, we derive the following research questions:

- **RQ1:** What is the comparative effectiveness of episode titles, episode descriptions, transcriptions, and audio features as representations for podcasts in terms of recommendation performance?

  In existing solutions for podcast recommender and retrieval systems, a variety of representation methods are proposed [XPX+16, MOG08]. To our knowledge, no evaluation of these methods has yet been published. This constitutes the possibility of investigating this question.

- **RQ2a:** How can a cross-domain recommender system between podcast segments and news articles be effectively built?

  Effectively built means that the proposed cross-domain recommender system will outperform baseline models, using only the transcriptions of podcast segments and the text of news articles.

- **RQ2b:** How well does this system perform in terms of recommendation performance, and beyond-accuracy measures compared to a baseline?

  Cross-domain recommender systems can be an approach to increase the quality and variety of recommendations. Especially in content-based systems, filter bubbles can be a huge issue [RRS22]. Therefore, we want to evaluate the recommendation performance and beyond-accuracy measures of the cross-domain recommender systems compared to baselines.

1.4 FALTER Verlagsgesellschaft mbH

This research work is done in partnership with the Christian Doppler Laboratory for Advancing the State-of-the-Art of Recommender Systems[1] at the TU Wien. In order to advance the development of recommender systems, the lab is collaborating with a wide variety of industry partners. One of its partners is the Austrian news company *FALTER Verlagsgesellschaft mbH (FALTER)* in Vienna. Founded in 1977 by students and artists, *FALTER* publishes a weekly print magazine covering politics, economics, media, culture, urban and natural environment [fal]. In addition to its long-standing online offering of news articles, *FALTER* also began publishing podcasts in 2017. For this work, *FALTER* kindly provides us with podcast and news article data, which will be used for the investigation of our formulated research questions.

[1] https://recsys-lab.at/

1.5 Methodology

In this research we apply a three cycle approach to design science research as described by A. Hevner in [Hev07]. This approach includes a relevance cycle, a rigor cycle, and a design cycle. In the relevance cycle, the application environment is analyzed, requirements are gathered, and finally a field test of the proposed solution is conducted. The rigor cycle assesses the state-of-the-art and evaluates existing theories, artifacts, and processes. This cycle ensures that the solutions produced are research contributions and not just routinely built IT artifacts [Hev07]. The design cycle is at the heart of this research methodology. It derives design requirements from the relevance cycle and theories and processes from the rigor cycle. The goal of this cycle is to iteratively build solutions and evaluate them against the collected requirements until a satisfactory result is obtained. Finally, the results of the design cycle are fed back into the research knowledge base. The following sections briefly explain the steps performed in each cycle.

1.5.1 Relevance Cycle

Qualitative Interview for Requirements Gathering In order to gather the requirements for the recommender system to be designed within this research, a qualitative semi-structured interview as described by Kallio et al. [KPJK16] is conducted. The expert participating in the interview is a podcast creator from our industry partner *FALTER*. The interview consists of questions about podcast representations, the general functionality and expectations of a cross-domain recommender system between podcasts and news articles, and the planned annotation study.

1.5.2 Rigor Cycle

Literature Review First, in this thesis, a literature review is performed. To begin, a tentative starting set of publications will be identified by searching Google Scholar and topic-specific conferences and workshops such as the ACM RecSys conference[2] or the podcast track of the Text Retrieval Conference (TREC)[3]. This initial set is be used to find related publications by examining the references of each publication in the set. The review is focused on podcasts, content-based and cross-domain recommender systems, information retrieval methods, and methods for representing podcasts using natural language processing techniques and non-textual features such as audio. Furthermore, existing approaches for the creation of datasets are examined.

Thesis Document Creation At the end of the research work, the contributions must flow back into the research knowledge base. This is done by publishing this master thesis document.

[2] https://recsys.acm.org/
[3] https://trecpodcasts.github.io/

1.5.3 Design Cycle

Dataset Definition and Annotation Study A dataset is defined to evaluate the recommender system. Furthermore, an annotation study is conducted to retrieve relevance annotations between podcast and news article pairs to evaluate the proposed solutions. The study involves three participants with different academic backgrounds. Before the study starts, the participants are informed about the research work. Each participant will be asked to annotate the same dataset, and a measure of inter-annotator agreement will be used to ensure consistency in label assignment across study participants. The study involves three participants to allow for majority voting in case of disagreement between annotators. The candidates for podcast/news article pairs to be annotated are generated using a baseline retrieval model. The annotation study is conducted using the survey tool LimeSurvey[4].

Selection and Implementation of Representation Methods Suitable representation candidates for podcasts are selected and implemented. To address RQ1, the recommendations for particular representations will be compared in terms of quantitative evaluation metrics.

Design and Implementation of a Content-Based Cross-Domain Podcast Recommender System To address RQ2, a concept for content-based cross-domain recommender systems between podcasts and news articles is designed and implemented. In addition, baseline models are implemented for the generation and evaluation of annotation candidates.

Quantitative Evaluation For a quantitative evaluation of the designed cross-domain recommender systems in terms of recommendation performance, the annotated dataset obtained during the annotation study is used. The dataset is divided into training and test subsets. The training set will be used to train the machine learning models, while the testing set will only be used to evaluate the proposed methods. The metrics we consider for the quantitative evaluation are *Mean Reciprocal Rank (MRR)* and *Normalized Discounted Cumulative Gain (NDCG)* and a beyond-accuracy metric representing the topical recommendation diversity.

[4]https://www.limesurvey.org/

1.6 Structure of the Work

This thesis is structured as follows: In Chapter 2, we describe the state-of-the-art in the literature related to this thesis. In Chapter 3, we describe how we designed an expert interview and the results we obtained from conducting this interview. In Chapter 4 and 5, we analyze the podcast and news article datasets used in this thesis. In Chapter 6, we describe the study design and results of the first annotation study. Then, in Chapter 7, we show how we extracted named entities from textual attributes in the data. In Chapter 8, we report on how we developed and conducted a second annotation study and the results we obtained. In Chapter 9, we show how we approached building a content-based cross-domain podcast recommender system using a learning-to-re-rank approach. In Chapter 10, we discuss the results obtained during the entire research work, and finally in Chapter 11, we end this document with a final conclusion.

CHAPTER 2

Related Work

In this chapter, we summarize related work that was found during the literature review. First, we describe podcast recommender systems in the literature in Section 2.1. We then summarize content-based approaches to podcast retrieval (described in Section 2.2) and cross-domain recommender systems (see Section 2.3) in the literature. Finally, in Section 2.4, we discuss the Transformer architecture and derived models that form the basis of the methods used in this work.

2.1 Podcast Recommender Systems

Since podcasts are a relatively new medium, not much research has been published in the area of podcast recommender systems. Existing approaches found in the literature can be categorized into *collaborative filtering* [GAR08, BFWC20], *sequential recommendations* [BFWC20] or *content-based solutions* [XPX+16].

In [GAR08], Gratz et al. propose an ad hoc collaborative filtering podcast recommender system for mobile networks. Collaborative filtering (CF) is a method for building recommender systems by finding similar *users* or *items* based on their historical tastes. To quantify the similarity in taste between two entities, the interaction history between users and items is used [RRS22]. Typically, CF-based recommender systems require that users have interacted with similar items in the past, so that these similar interactions can be used to quantify the similarity between users and users or items and items. Since Gratz et al. aim to build a recommender system that finds users with similar tastes in a local network of mobile devices, this requirement is unlikely to be met. Therefore, instead of using the interactions between users and items directly for CF, the authors derive a content-based description of a user's listening history using keywords. These keywords are weighted by ratings given by a user, and then these weighted listening history descriptions are used to find similar users by computing the cosine similarity [LPS16]. Finally, a recommendation is made using the items of neighboring users with

high ratings. Since there is no publicly available podcast dataset with user ratings, the authors use the MovieLens dataset [HK15] for their experiments. The lack of availability of an interaction-based dataset for podcast recommendation represents a major research gap, and due to the use of the MovieLens dataset, the results of Gratz et al. are not transferable to the podcast recommendation scenario.

Currently, interaction-based datasets for the podcast domain are only available from large streaming providers. Without providing detailed information on the dataset used, Benton et al. from Spotify [BFWC20] compare the performance of a CF-based user-to-podcast-show recommendation system with a sequential approach. User-to-podcast-show means that they do not aim to recommend a specific podcast episode to a user, but a podcast show. For their experiments, they exclude the last podcast show a user listened to from his or her interaction history and then evaluate whether this show was predicted by both recommender system approaches.

The authors' CF-based recommender is built using a matrix factorization approach. They construct a matrix X, where each row in the matrix corresponds to a user and each column corresponds to a podcast show, and each entry in the matrix is one if a user has listened to a show and zero if not. Given this matrix X, the authors use the coordinate descent algorithm [Wri15] to find matrices W and H such that $X \approx WH$ [CP09]. After finding these matrices, the rows in W are embeddings of users and the columns in H are embeddings of shows. Using these matrices, recommendations for a user can be made by computing the cosine similarity of a user vector in W with all show vectors in H. The shows with the highest similarity are then recommended to a user [BFWC20].

For their sequential approach, the authors use the sequential listening history of a user, where each entity in this history is a podcast show that the user has listened to. Using a knowledge graph of the podcast library, which is matched with data from publicly available data sources such as Wikipedia, the authors represent each podcast show as an embedding of an entity in this knowledge graph [YYH+14]. Details of how exactly this knowledge graph was created are not further described by the authors. Given the sequences of knowledge graph embeddings representing podcast shows, Benton et al. train a recurrent neural network (RNN) that predicts the next show in the sequence based on the shows previously fed to the network. With this sequential knowledge graph embedding approach, the authors achieve a 450% increase in recommendation performance compared to their CF-based approach [BFWC20].

Unlike CF-based or sequence-based systems, content-based recommender systems can be built without interaction-based datasets. In [XPX+16], Xing et al. describe how they built a content-based podcast recommender system by combining embeddings of all textual attributes associated with a podcast with numerical features. The textual attributes include episode titles, episode descriptions, tags, and uploader names. The numerical attributes include durations, play counts, download counts, and timestamps. Although the authors call their approach a content-based recommender system, their approach predicts whether a particular user has uploaded a podcast episode to a podcast platform rather than making recommendations. Xing et al. use the embeddings derived from

textual attributes together with the numerical features to first build a vector containing the features of all podcast episodes that a user has uploaded to the Chinese podcast platform Ximalaya[1]. They then use pairs of these combined user and item features to fit a decision tree model that predicts whether or not that episode was uploaded to the platform. The authors explain that they use this approach because interaction data between users and items is not available, and so they model a user's affinity for a podcast through the proxy of uploading it to the platform.

In [RA23], Raharjo et al. also proposed a content-based recommender system in the context of educational podcasts. The authors used a semi-personalized approach in which users receive recommendations of new podcast episodes based on previously selected episodes. These previously selected episodes are referred to as *profile items* by the authors. To retrieve podcast episodes for recommendation, the authors use only the episode titles of the podcasts. Raharjo et al. represent these titles using the *term frequency-inverse document frequency (TF-IDF)* statistic. This statistic measures how important particular words are to a document and provides a more sophisticated way of representing text than simply using a vector of word counts (also called a bag of words). Using these TF-IDF representations of the episode titles of the selected profile items, their system generates recommendations by measuring the cosine similarity [LPS16] with the representations of other podcast episodes. Again, the authors of this paper do not really evaluate how relevant these recommendations are, but only compare whether the recommended podcasts are in the same category (an attribute included in their dataset) as the selected profile items.

The interaction data needed to perform collaborative filtering or sequential recommendations is not publicly available, and the content-based approaches of Xing et al. [XPX+16] and Raharjo et al. [RA23] are questionable, especially in terms of their evaluation. Moreover, the goal of this work is to build a cross-domain recommender system between podcasts and news articles, not a single-domain recommender system like the methods described here. Therefore, in this work we will follow an information retrieval-inspired approach that works with a small amount of annotated data. Since no annotated data is available at all, we will annotate it manually in an annotation study.

2.2 Content-Based Podcast Retrieval

In contrast to recommender systems, more research has been published on content-based podcast retrieval. One reason for this is the publication of "100,000 Podcasts: A spoken English document corpus" [CRY+20], a podcast information retrieval dataset from Spotify. This dataset formed the basis for the podcast track of the Text Retrieval Conference (TREC)[2] in 2020 [JCC+20] and 2021 [JCC+21].

The dataset by Clifton et al. [CRY+20] consists of over 100,000 randomly sampled podcast episodes that contain nearly 60,000 hours of audio. In addition to this massive

[1] https://m.ximalaya.com/
[2] https://trec.nist.gov/

amount of audio data, the authors also published metadata such as episode titles, episode descriptions, and transcriptions for two-minute segments of podcast episodes generated using automatic speech recognition. The podcast episodes in the dataset come from 18,376 podcast shows, and the average length of an episode is 33.8 minutes, resulting in 5,700 transcribed words. In addition, the dataset contains 1,669 labels for query-segment pairs and 303 labels for description-episode pairs on an Excellent/Good/Fair/Bad (EGFB) scale, which have been annotated by humans. Based on these two annotated pair types, a podcast search and podcast summarization task was initially launched as part of TREC 2020. The query and podcast segment pairs were used for podcast search, and the episode description and episode pairs were used for summarization within TREC. For search, the labels reflect how well a particular search result matches the query, while for summarization, the labels reflect how well the episode description summarizes the podcast episode [CRY+20]. As part of TREC 2021, the podcast search task was expanded to require not only a ranked list of podcast segments for each query, but also three additional ranked lists based on whether they are entertaining, subjective, or contain discussion [Sob21].

The task of podcast search is closely related to our work, since its goal is to retrieve an item based on the content of another item, as is also the case in non-personalized content-based recommender systems. In [CRY+20], Clifton et al. describe baseline models for search. These baselines are built using term frequency-based models such as BM25 [RW94] with segment transcriptions as the documents to be retrieved based on the queries. The best results for podcast search in TREC 2020 were achieved by Galušcáková et al. [GNO20]. In their best submission, they use seven different term frequency-based approaches, such as the TF-IDF model [MV15], to retrieve different lists of podcast segments that represent potentially relevant podcast segments. These lists are then combined into a single list. Using the combined list, four different re-ranking models based on the Transformer architecture [VSP+17] (see also Section 2.4) are applied, with each model producing a new ranked list. Finally, these lists are again combined into one list, which is then the final list of results for a given query. With this complex two-step method, involving a total of 11 different models, the authors are able to significantly outperform all other submissions.

In 2021, colleagues from the TU Wien [HSH21] achieved a top performance in the podcast search task. Hofstätter et al. use their TAS-B model [HLY+21] trained on the human-generated machine reading comprehension dataset (MS MARCO) [NRS+16] in combination with BM25 [RW94]. TAS-B is a Transformer-based text retrieval model that uses a topic-aware query and balanced margin sampling technique [HLY+21] for training and has achieved state-of-the-art results on other TREC tasks. The combination of the Transformer-based TAS-B model and the term-frequency-based BM25 model is referred to by the authors as a hybrid sparse-dense retrieval approach.

While all podcast search approaches in 2020 used only textual attributes, as originally provided by Clifton et al. [CRY+20], the submissions in 2021 also included three solutions that made use of features derived from the audio signal. Unfortunately, a paper was only

published for one of these submissions. In this paper [BFG+21] Bondarenko et al. describe that they extracted features using the pre-trained COLA embedding model [SGZ21], which they then used to re-rank the results based on whether they were entertaining, subjective, or contained discussion. However, the authors do not see any improvement in the evaluation results using these features. The short descriptions of the approaches in [JCC+21] suggest that the other two submissions also used audio features for re-ranking only. Neither an improvement in terms of evaluation results can be found for them.

As noted above, most of the submissions to the TREC podcast search tracks used textual attributes only. One publication that was not a submission to the TREC podcast track, but builds on the results of TREC podcast track 2020 [JCC+20], is "Podcast Metadata and Content: Episode Relevance and Attractiveness in Ad Hoc Search" by Carterette et al. [CJJ+21]. In this paper, the authors examine the effectiveness of the textual attributes contained in the Clifton et al. [CRY+20] when conducting a podcast search. To do this, the authors first annotated additional data for metadata and full episode transcriptions with respect to their *attractiveness* and *relevance* using three new scoring types. Since episode titles and episode descriptions are often not accurate enough to reflect the relevance of a podcast episode, the authors decided to annotate two of the three new assessment types based on attractiveness rather than relevance. The three additional assessment types used by the authors are:

- **Title attractiveness:** Given a title of an episode, would a user find this episode attractive to consume?
- **Title and description attractiveness:** Given a title and an episode description, would a user find this episode attractive to consume?
- **Full transcript relevance:** Given the transcript of a full podcast episode, does the user find this episode relevant to the query?

Using these assessments in conjunction with the original assessments (i.e., the annotated relevance of a podcast segment transcription to a query), the authors built indices using episode titles, episode descriptions, concatenated episode titles and descriptions, full transcripts, and concatenated transcripts, episode titles, and descriptions. To create these five different indices, the authors used the open source search library Apache Lucene [BMII12], without further specifying the settings used. Using each of these created indices, the authors evaluated their performance in terms of the four different types of assessments. The authors' findings are that using episode titles and descriptions yields the best results in terms of attractiveness but at the same time can lead to less relevant rankings. However, using transcripts results in more relevant but less attractive episodes. The combination of metadata and transcripts leads to the strongest relevance, with only a slight decrease in attractiveness. Overall, the authors conclude that metadata and transcriptions should be used when building search engines and that the attributes presented to the user should be carefully considered depending on the application and context.

Another paper on content-based podcast retrieval that is closely related to this work was published by Mizuno et al. [MOG08] before the existence of the Clifton et al. dataset in 2008. In their paper, the authors propose to retrieve podcasts with similar content using transcriptions generated by automatic speech recognition. Since the quality of automatic speech recognition was not as good as it is today, Mizuno et al. use a *confusion network* [MBS00] in their work. A confusion network is a network that condenses intermediate speech recognition results in such a way that their word error rate is minimized [MOG08]. From this confusion network, the authors extract keywords that describe the content of a podcast episode using TF-IDF. They then retrieve similar episodes by computing the cosine similarity [LPS16] between vectors of episode keywords. To evaluate their approach, the authors manually annotated pairs of podcast episodes on a four-point scale based on their similarity. The authors found that their confusion network approach improved episode retrieval compared to a regular TF-IDF approach, but only when the transcription quality was not too low. Although the methods described in this paper are no longer state of the art, they still show that earlier approaches measuring the similarity between podcast episode representations have been used to retrieve information in the podcast domain. Moreover, the manual annotation performed by the authors again highlights the lack of available datasets annotated based on similarity in the podcast domain.

Within this research, we will create a dataset that follows the approach of Clifton et al. [CRY+20] when they created their dataset. Furthermore, we will adopt approaches from the TREC podcast tracks by framing our content-based cross-domain podcast recommender system as a search problem for finding news articles based on podcast segments.

2.3 Cross-Domain Recommender Systems

A cross-domain recommender system (CDRS) is a recommender system that leverages knowledge in a source domain to generate recommendations in a target domain. Cross-domain recommenders do this with the goal of improving the overall quality of recommendations [RRS22]. To achieve this, the source and target domains must be linked. While there are different definitions of domain change in CDRS in the literature [RRS22, ZWC+21, ZZL+22], we use the notion described by Zhu et al. in [ZWC+21]. The authors define the following three categorizations on the domain change in cross-domain recommender systems:

- **Content-level relevance:** The source and target domains are linked by using content-based similarity (i.e., similarity of attributes) between *users* or *items* in both domains. Both domains do not need to have common *items* or *users*, but need entities that are content-related to some degree.
- **User-level relevance:** The source and target domains share common *users* and different *items* (e.g., books and movies), and the domains are linked through these

common *users*.

- **Item-level relevance:** The source and target domains share common *items* (e.g., movies) but have different *users* in each domain (e.g., Netflix and Amazon). The domain linkage in such systems is through the common items, and this type of domain change is often referred to as *cross-system* or *cross-platform recommendation*.

CDRS using data from the podcast or news domains [NCP+20, LSC+17, ESH15] can all be categorized as user-level relevance-based systems. In [NCP+20], Nazari et al. use users' music listening behavior and taste to solve the *user cold-start problem* in the podcast domain. The user cold-start problem describes the problem that there is no or not enough interaction history for the recommender system to draw on, and thus other approaches must be pursued [NCP+20]. In [LSC+17], the authors take a related approach to solving the *dual cold-start problem* in the news domain by leveraging information about app installations on a user's mobile device. The dual cold-start problem describes the situation where not only is a user new to the system, but also the item is newly added, and thus there is not enough interaction history for both entities.

State-of-the-art data-agnostic approaches to cross-domain recommender systems, such as [ZCW+19, ZWC+20, MXM+24, ZZH+23, LLZ+23], all require interaction data between users and items in different domains. As mentioned in the previous chapter when describing the problem statement of the problem of this work, such data is not available for the podcast domain, and especially not for the cross-domain case between podcasts and news articles. Thus, in this work we aim to build a *non-personalized content-based* cross-domain recommender system. Non-personalized means that the recommendations made by the system do not discriminate between users [KC16]. Content-based means that our goal is to generate recommendations based solely on the content of items. Therefore, the approach we will follow can be clearly categorized as a content-level relevance-based CDRS. Although the similarity of podcasts to other textual media such as news articles or web blogs provides an opportunity to exploit their content for cross-domain recommendation scenarios, no such system has been published in research. As also suggested by Jones et al. in [JZS+21], cross-domain recommendations should be explored for the podcast domain, since other textual domains are likely to contain rich information about users' topical preferences.

2.4 The Transformer Architecture

The publication of the paper "Attention Is All You Need" by Vaswani et al. [VSP+17] in 2017 marked a change in the era of natural language processing (NLP). Until then, NLP models were based on either bag-of-words (BoW) strategies, which represent text sequences as a set of words regardless of their order, or recurrent neural networks (RNNs), which incorporate each word as input sequentially to construct internal representations of text sequences. BoW-based models have the disadvantage of losing contextual information of the text by ignoring the order of the words. This drastically limits the performance of

such methods. While RNNs overcome this problem by processing words sequentially, this approach introduces a bottleneck in parallelization because the internal state of the models for a given token always depends on previous states. This becomes especially critical when processing longer text sequences in large corpora. The Transformer architecture introduced in [VSP+17] overcomes these problems by relying entirely on an attention mechanism that allows global relationships between inputs and outputs to be captured.

The Transformer architecture consists of an encoder and a decoder. The encoder maps an input sequence of symbol representations (i.e. learned vector representations of words, often referred to as word embeddings) $\boldsymbol{x} = \{x_1, ..., x_n\}$ to a sequence of continuous representations $\boldsymbol{z} = \{z_1, ..., z_n\}$. The decoder is then used to sequentially generate an output sequence of symbol representations $\boldsymbol{y} = \{y_1, ..., y_m\}$ from the sequence $\boldsymbol{z}$. This generation of symbol representations works in an auto-regressive way, i.e. each generated representation is used as an additional input for the generation of the next one. The components of the encoder and decoder of a Transformer model are shown in Figure 2.1 and are explained below.

The encoder component of the proposed architecture consists of $N = 6$ stacked layers, where each of these layers contains a multi-head attention sub-layer followed by a feed-forward sub-layer with fully connected neurons. Furthermore, the input of each sub-layer is added to the produced output and then normalized. This addition of the input to the output is called a residual connection and addresses the problems that arise when optimizing the model. All of the aforementioned sub-layers produce outputs of dimension $d_{model} = 512$. Before we elaborate on the concept of multi-head attention, we describe regular single-head attention:

Attention is a mechanism that weights the importance of input tokens for the corresponding output tokens. For a particular token, an attention function uses queries and keys of dimension d_k and values of dimension d_v as input. The result of the attention function is obtained by computing the dot products of the query with all keys, then dividing it by $\sqrt{d_k}$ and finally applying a softmax function to receive the weights for the values. This computation can be combined for multiple queries Q using the following formula:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

To compute attention for multiple parts of the input sequence in parallel, queries, keys, and values are linearly projected h times using different learned linear projections to dimensions d_k, d_k, and d_v. This joint computation of attention is referred to as mutli-head attention and is calculated as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{2.2}$$

where

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \tag{2.3}$$

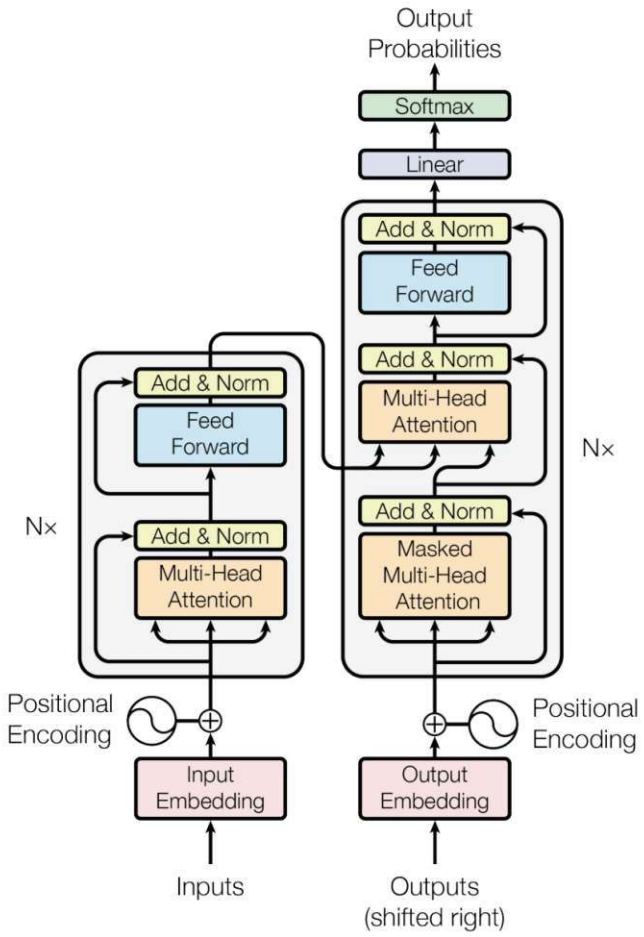

Figure 2.1: The Transformer architecture with its encoder (left) and decoder (right) components [VSP+17].

$W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ are the learned projection matrices of the Transformer model. The architecture proposed by Vaswani et al. contains $h = 8$ of the described attention heads.

Within the Transformer, two types of attention layers can be distinguished: self-attention and encoder-decoder attention layers. Self-attention measures the interaction and relationship between words in a sequence. This allows the model to weigh the importance of words in the sequence when encoding or decoding and captures important contextual information. Self-attention is used in the encoder and decoder layers. Encoder-decoder attention is primarily used in the decoder layers and allows the decoder to focus on relevant parts of the input sequence when decoding the output. Looking again at the sub-layers of the decoder in Figure 2.1, we can see a masked multi-head attention layer. This layer differs from regular multi-head attention layers in that it masks parts of the input so that future tokens that have not yet been generated are ignored. This ensures

that the generation of tokens is auto-regressive.

Since the Transformer architecture is based solely on attention and processes the entire input sequence at once, a mechanism must be implemented that provides the model with information about the order of the tokens in the sequence. Within [VSP+17], the authors propose *positional encodings* to solve this problem. Although the authors also evaluated learnable positional embeddings that yielded similar results, Vaswani et al. decided to use sine and cosine functions of different frequencies that are added to the input word embeddings. The two functions used by the authors are as follows

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{2.4}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{2.5}$$

where *pos* denotes the position and i the dimensionality.

The original Transformer described above gave rise to many publications proposing model training strategies and architectural variants, such as Bidirectional Encoder Representations from Transformers (BERT) [DCLT18], Robustly optimized BERT approach (RoBERTa) [LOG+19], or the recently very popular Generative Pre-trained Transformer (GPT) [RNS+18], which marked another breakthrough in natural language processing. BERT is a family of general-purpose Transformers that are pre-trained on unlabeled data by performing two unsupervised tasks: i) Masked Language Modeling (MLM) and ii) Next Sentence Prediction (NSP). In MLM, 15% of the input sequence tokens are randomly masked, and the model's task is to predict these masked tokens. This allows the model to learn the relationship between words. In NSP, the task of the model is to predict the sentence that follows a given input sequence, allowing the model to learn relationships between sentences. In order to use BERT for a specific downstream task, such as sentence classification or question answering (QA), BERT must be fine-tuned with task-specific data. This fine-tuning step is relatively inexpensive, as the model benefits from knowledge gained from general unsupervised pre-training. When BERT was introduced in 2018, it achieved new state-of-the-art results in 11 benchmark natural language modeling tasks [DCLT18].

In 2019, Liu et al. present a BERT pre-training replication study that examines the influence of hyperparameters and training data in [LOG+19]. Liu et al. found that the previously proposed BERT models were significantly undertrained and proved that the performance of BERT models can be significantly improved by longer training, using longer input sequences and larger data batches, removing NSP pre-training, and using a dynamic masking approach when performing MLM pre-training. The improved model proposed by the authors is called RoBERTa and outperforms BERT in several NLP benchmarking tasks [LOG+19].

In this research, we use several Transformer-based models fine-tuned for automatic speech recognition, language classification, and text representation, all of which are described in the following chapters.

CHAPTER 3

Expert Interview

To gather requirements and assess the relevance of a content-based cross-domain podcast recommendation system, we conduct an interview with a podcast producer from *FALTER*. In the following sections, we describe the method and design of the expert interview, followed by the results obtained.

3.1 Method and Design

Interviewing a subject matter expert to collect data is a qualitative research method. The most widely used method is the semi-structured qualitative interview [KPJK16, Bla13]. Since semi-structured interviews are versatile and flexible and are particularly suitable for collecting expert opinions, we also use this method in this thesis.

Semi-structured qualitative interviews use a so-called *interview guide*, which contains the main topics to be discussed during the interview. The term *semi* in semi-structured qualitative interviews comes from the fact that this interview guide only provides an interview framework, but the interview is conducted dynamically by asking follow-up questions or new questions that are not contained in the guide [KPJK16, Bla13].

In [KPJK16], Kallio et al. propose a framework to develop a semi-structured interview guide, which we follow to conduct the expert interview within this thesis. After ensuring that the expert interview meets the criteria of being conducted as a semi-structured interview, as it is flexible enough for the expert to give their full perspective and opinion on the topic, we design an initial interview guide. The initial guide includes questions to gather information about the role of the interviewee, *FALTER*'s podcast offering, expectations for podcast recommender systems, podcast representation, and content-based relevance between podcasts and news articles.

In parallel with the creation of the initial interview guide, we create a slide deck that introduces the thesis and the corresponding research questions. As recommended in

Dimension	Question
Background	What is your position at FALTER? What is your relationship with the FALTER podcast offering? How would you describe the FALTER podcast offering in your own words (Contents, characteristics, formats, target audience, etc.)? What is your experience with recommender systems? What do you expect from a (cross-domain) podcast recommendation system?
Podcast representation	Do you think there are parts of podcasts (beginning, middle, end) that are more or less important? Which representation of podcasts do you think is most valuable for making recommendations for news articles? How many topics are covered in a podcast? How long is each topic covered in a podcast? How long should podcast segments be to recommend news articles based on them?
Annotation study	How many gradations do you think are necessary to assess the relevance between podcast segments and news articles? Would you say that if a news article is relevant to a podcast segment, the podcast segment is also relevant to the news article?

Table 3.1: Semi-structured qualitative interview guide used in the expert interview.

[KPJK16], we conduct a pilot interview test in which we test the created slide deck and the initial interview guide. We find that we need to create additional slides for the introductory slide deck and reorder the questions in the interview guide. After these modifications, we finally have the interview guide shown in Table 3.1.

In a one-hour face-to-face meeting at *FALTER*'s offices, we first present the slide deck and then conduct the interview with a person responsible for the production of the podcast program. We record the audio of the interview and later transcribe the interview using automatic speech recognition. The results of this expert interview are described in the following sections.

3.2 Results

In the following sections, we describe the results of the expert interview. To extract key findings from the expert interview, we use the transcription created using automatic speech recognition. We briefly summarize the information obtained about the interviewee's background, the representation of podcasts, and the annotation study.

3.2.1 Background

The interviewee works in the audiovisual department of *FALTER* and spends most of the time producing podcasts and videos. Their responsibilities include the entire production process from start to finish, including content planning, recording, editing, and post-production. In addition, they research content for specific podcasts and direct the recording process.

The interviewee describes *FALTER*'s podcast offerings as very diverse, with shows ranging from topics such as politics or true crime to literature, with some shows approaching topics in a humorous way. The structure of the shows varies from structured interviews to casual conversations, with all shows including intro and outro sections. The most popular podcasts are of a political nature, which can be attributed to the generally political readership of *FALTER*'s news articles. Podcasts dealing with literature are less popular and consumed by only a few people, as this marks a niche interest.

The expert says that they enjoy using recommender systems, especially on music streaming platforms such as Spotify, where playlists are created based on users' listening habits. They say that such features help them discover items that they would not have discovered otherwise. However, they also describe how some recommender systems have given them recommendations that are overly personalized to a particular item they have viewed, and therefore fail to find relevant recommendations overall.

Their main expectation of a recommender system within *FALTER* is to motivate users to stay on *FALTER*'s website as long as possible and to encourage interaction with items in order to place advertising effectively. In addition, the expert expects the system to recommend items that are contextually relevant, including older content that has semantic similarities to current publications. The interviewee emphasizes that a recommender system should go beyond the use of attribute similarities and be able to recognize thematic overlaps between podcasts and news articles in order to present appropriate recommendations to the user.

3.2.2 Podcast Representation

Asking about parts of podcasts that are more or less important to creating recommendations, the interviewee replies that this is strongly dependent on the format of the particular podcast show. They say that the structure of each podcast show is different and can even differ between single episodes. Although the expert says that there is no applicable rule as to which part of a podcast is most important, most episodes contain

intro and outro sections that can range from a few seconds to three minutes. In addition, several seconds of musical elements can be included in the podcast.

According to the interviewee, podcast episode descriptions are the most valuable representations to make recommendations, as they provide a concise overview of the topics discussed. The expert says that titles are probably also useful but are often too concise to pique listeners' interest. They also say that podcast transcripts can be interesting, but topics of conversation tend to vary throughout an episode. According to the interviewee, the number of topics and the length of the topics discussed during an episode are highly dependent on the podcast show. They say that there are shows where many small topics are discussed densely one after the other and shows that usually cover one broad and complex topic. Regarding the length of podcast segments to make recommendations, the expert does not have a clear advice. They say that segment length is a trade-off between capturing the right context and isolating specific topics. The interviewee expects that it may also be beneficial to use different segment lengths for different formats because they treat topics at different levels of abstraction. For episodes where topics change frequently, the expert suggests using small podcast segments under 10 minutes.

3.2.3 Annotation Study

With respect to the gradations in relevance annotation between podcast segment and news article pairs, the interviewee states that binary annotations (e.g., relevant, non-relevant) are probably sufficient, since it should not make a difference to the user how much they like an article. Furthermore, they emphasize that relevance annotation can be challenging, especially in cases where transcriptions are inaccurate, which should be addressed by data cleaning methods. They also emphasize that it can be important to give clear instructions to annotators to ensure consistent data.

Concerning the directionality of relevance between podcast segments and news articles, the expert expects it to be bidirectional. In other words, it seems reasonable to assume that if a podcast segment is relevant to a news article, then the news article could also be considered relevant to the podcast segment. This would imply that recommendations could be made in both directions similarly.

CHAPTER 4

Podcast Dataset

The podcast dataset is a key component of this research and is derived from the podcast offering of our industry partner *FALTER*. In this chapter, we first describe the *FALTER* podcast offering, followed by how we created an initial dataset by crawling their RSS feeds. We then explore the attributes of the dataset, generate transcriptions from the audio files, and finally perform an in-depth analysis of the transcriptions using statistical measures, topic modeling, and classification methods.

4.1 The *FALTER* Podcasts

In September 2017, *FALTER* expands its news offering with the publication of podcasts. The first podcast show published by *FALTER* is "FALTER Radio". From September 2017 until May 2023, *FALTER* has grown its podcast offering to four different podcast shows that are very different in style and content. Each of these shows is briefly described below:

- ***Besser lesen mit dem FALTER***[1] is a show in which the bookseller Petra Hartlieb invites authors to talk about their newest books and reading in general.
- ***FALTER Radio***[2] is a podcast hosted by the journalist Raimund Löw, where he invites *FALTER* editors to speak about interesting stories from the current newspaper publications.
- ***Klenk + Reiter***[3] is a podcast by the forensic pathologist and university professor Christian Reiter and the chief editor of *FALTER*, Florian Klenk, where they talk about the most spectacular criminal cases in the Austrian history.

[1] https://www.falter.at/buchpodcast
[2] https://www.falter.at/falter/radio
[3] https://www.falter.at/gerichtsmedizin

- ***Scheuba fragt nach***[4] is a show that is hosted by the cabaret artist and author Florian Scheuba and satirically discusses current national topics with different guests in each episode.

4.2 Dataset Creation

The podcast dataset is created by collecting all episodes of the shows mentioned in Section 4.1 from the *FALTER* podcast RSS feeds[5,6,7,8] through April 15, 2023. This results in a total of 1164 different episodes. Due to the different ages and release frequencies of the shows, we get a highly unbalanced number of episodes per show, as shown in Figure 4.1.

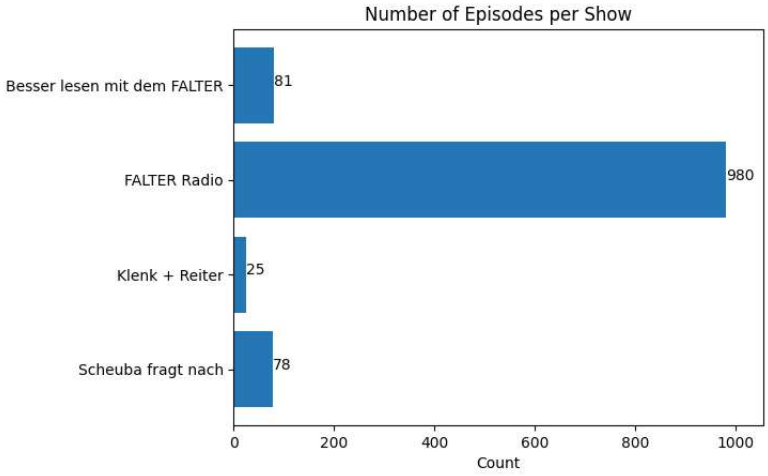

Figure 4.1: Number of episodes per show in the *FALTER* podcast dataset.

We can see that "FALTER Radio" has the highest number of episodes with a total of 980. Followed by "Besser lesen mit dem FALTER" with 81 episodes and "Scheuba fragt nach" with 78 episodes. The show with the fewest episodes is "Klenk + Reiter" with a total of 25 episodes.

4.3 Metadata

For each of the 1164 episodes, eight metadata attributes are retrieved from the RSS feeds. These metadata attributes include the episode title, the episode ID, a brief description of the episode, the publication date, the duration, the title of the show, the filename, and a URL pointing to the corresponding audio file in MP3 format. All attributes except the

[4]https://www.falter.at/scheuba
[5]https://feeds.acast.com/public/shows/1974c592-0a34-5e8c-a961-7ff704ac4476
[6]https://feeds.acast.com/public/shows/a869c471-dcd6-5bb9-aa22-0d45def50b1c
[7]https://feeds.acast.com/public/shows/6332f5b140179e001274bb37
[8]https://feeds.acast.com/public/shows/620e76fac97dbc00135a3d68

publication date, duration, description, and title of the shows are unique for each record in the dataset. We now describe the evolution of episode publications over time, followed by duration and the textual attributes contained in the metadata.

4.3.1 Publication Date

As mentioned above, *FALTER* started publishing its first podcast show "FALTER Radio" in 2017 and expanded its podcast offering to four shows in May 2023. In Figure 4.2, the development over time of the episodes of podcasts published per show is shown.

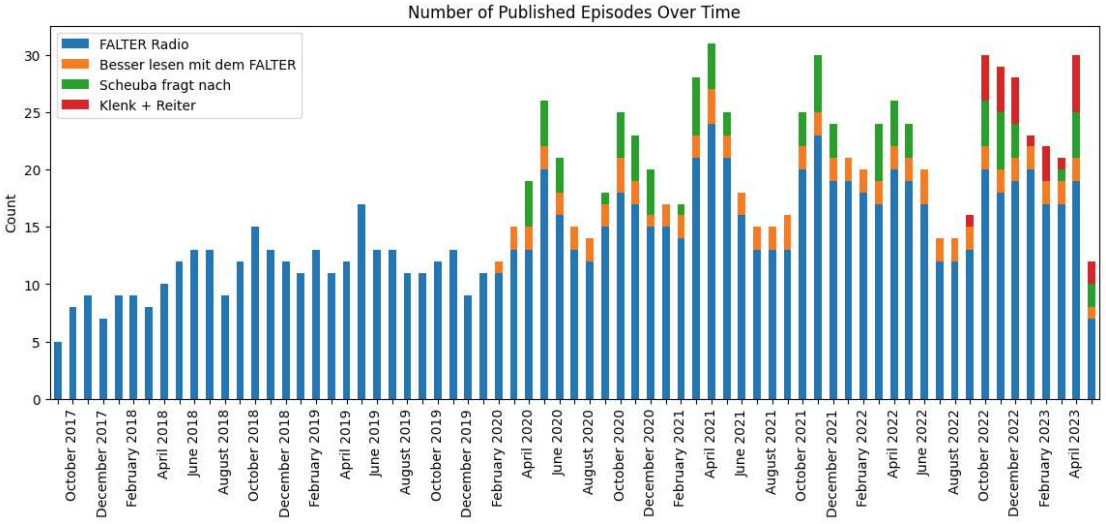

Figure 4.2: Development over time of episode publications per show in the *FALTER* podcast dataset.

One can see that until January 2020 only episodes of "FALTER Radio" were published. In February 2020, *FALTER* published the first episode of their second show "Besser lesen mit dem FALTER". Two months later, in April 2020 they added the third show "Scheuba fragt nach" to their repertoire. Despite a few months where no episodes of "Scheuba fragt nach" were published, the offering included three shows for over two years. In September 2020, *FALTER* decides to launch its fourth podcast show "Klenk + Reiter". Looking at the distribution of published episodes for a particular month, we can see that even with the introduction of new shows, "FALTER Radio" has the highest publication frequency of all podcasts. One reason for this may be that the show discusses content that has already been published in the *FALTER* magazines and is therefore easier to produce than shows that include new content.

4.3.2 Duration

Looking at the duration of the episodes of the podcast shows, we can calculate an arithmetic mean of 41.606 minutes with a standard deviation of 17.146 minutes and a

median of 36.383 minutes for all four shows. More statistical measures are shown in Table 4.1.

If we look at the boxplot of duration per show shown in Figure 4.3, we can see some outlier points representing podcast episodes that are overly short or long. The short episodes are most likely cases where some sort of announcement was made, such as an episode being skipped due to illness. Since these episodes do not reflect the content of a regular episode, they should be removed from the dataset. The overly long episodes are probably caused by topics that contain a lot of content, so we see no need to remove them from the dataset.

Show	Mean	Std.	Q 25%	Q 50%	Q 75%	Min	Max
Besser lesen mit dem F.	34.087	3.855	31.633	34.017	36.350	25.683	46.350
FALTER Radio	42.389	18.261	30.679	36.517	51.054	4.733	132.850
Klenk + Reiter	36.826	13.470	35.017	37.967	41.400	1.950	73.733
Scheuba fragt nach	41.112	7.319	34.229	41.700	46.229	25.167	56.417
All shows	41.606	17.146	31.062	36.383	48.871	1.950	132.850

Table 4.1: Statistics of the podcast duration attribute for each show separately and all shows combined.

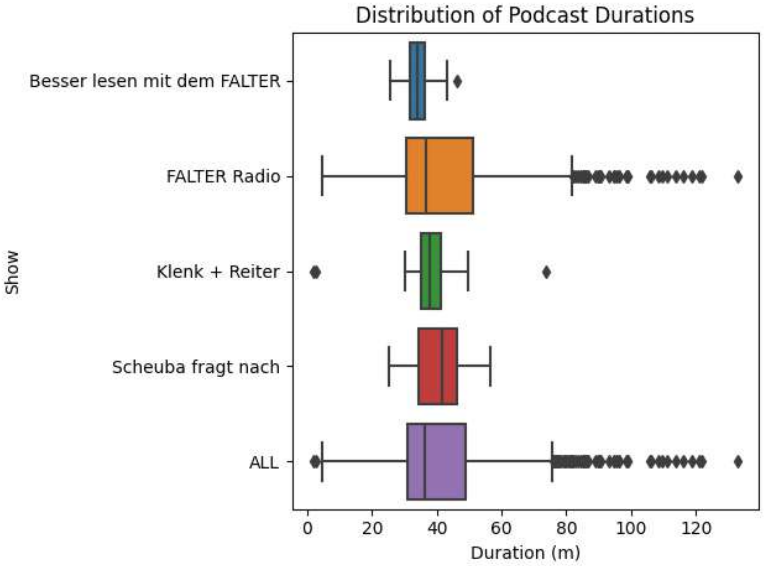

Figure 4.3: Boxplot of the duration per show in the *FALTER* podcast dataset.

If we look at the spread of the duration, we see a very similar picture for "Besser lesen mit dem FALTER", "Klenk + Reiter", and "Scheuba fragt nach". Only "FALTER Radio" has a wider spread. Reasons for this could be that there was no desired length of the

episodes when *FALTER* started podcast production, or that the topics of the magazines differ in content that is discussed in the podcast.

4.3.3 Episode Title and Description

Finally, we examine the textual attributes contained in the metadata: the episode title and the episode description. We apply word tokenization to both attributes using the Natural Language Toolkit[9] (NLTK) in Python and describe the statistics of the number of tokens obtained. In Table 4.2 and Figure 4.2, we can see that the length of podcast episode titles ranges from four to 17 tokens and has a mean of 8.411 with a standard deviation of 2.438 for all the podcast shows.

Show	Mean	Std	Q 25%	Q 50%	Q 75%	Min	Max
Besser lesen mit dem F.	5.160	0.782	5.0	5.0	5.0	5.0	10.0
FALTER Radio	8.596	2.401	7.0	8.0	10.0	4.0	17.0
Klenk + Reiter	6.640	1.221	6.0	7.0	7.0	4.0	10.0
Scheuba fragt nach	10.026	0.359	10.0	10.0	10.0	9.0	13.0
All shows	8.411	2.438	7.0	8.0	10.0	4.0	17.0

Table 4.2: Statistics of the number of tokens of the tokenized podcast episode titles for each show.

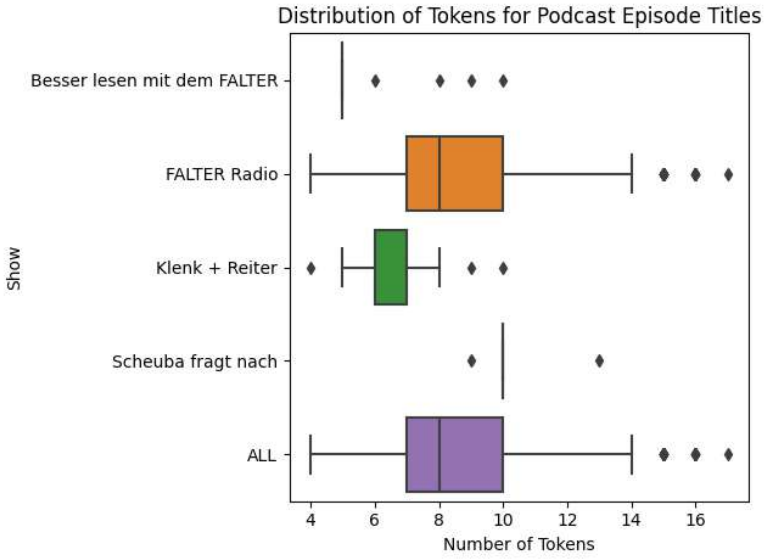

Figure 4.4: Boxplot of number of tokens from the tokenized podcast episode titles per show.

[9]https://www.nltk.org/

Show	Mean	Std	Q 25%	Q 50%	Q 75%	Min	Max
Besser lesen mit dem F.	257.000	62.888	210.00	246.0	292.0	119.0	424.0
FALTER Radio	166.236	69.515	126.00	153.0	190.0	68.0	514.0
Klenk + Reiter	266.240	60.174	213.00	277.0	301.0	169.0	424.0
Scheuba fragt nach	106.603	20.693	92.25	100.0	113.5	74.0	174.0
All shows	170.704	73.825	124.00	154.0	198.0	68.0	514.0

Table 4.3: Statistics of the number of tokens of the tokenized podcast episode descriptions for each show.

"Besser lesen mit dem FALTER" and "Scheuba fragt nach" have a smaller standard deviation than "FALTER Radio" and "Klenk + Reiter" (0.782 and 0.359 vs. 2.401 and 1.221). The reason for this is that the first two mentioned shows have fixed title naming schemes where only the guests name is different for each episode (e.g. "Besser lesen mit *<Guests Name>*"), whereas "FALTER Radio" and "Klenk + Reiter" use titles, that describe the content of the episode (e.g. "Wie funktioniert Österreichs Medienwelt?"). This fact makes the titles of the first two mentioned shows less relevant to be used as podcast representation.

Table 4.3 and Figure 4.5 show similar information for the podcast episode descriptions. The number of tokens ranges from 68 to 514 for all shows. The calculated arithmetic mean is 170.704 with a standard deviation of 73.825 tokens.

The show with the longest average episode description is "Klenk + Reiter" with an arithmetic mean of 266.240 tokens, followed by "Besser lesen mit dem FALTER" with a mean of 257 tokens. The episode descriptions of "FALTER Radio" contain 166.236 tokens on average. The show with the smallest average episode description length is "Scheuba fragt nach" with only 106.603 tokens. In contrast to the episode titles, no fixed naming scheme is used for any of the episode descriptions of all podcast shows. This qualifies this attribute as a valuable podcast representation.

4.4 Transcriptions

Since the main source of content in podcasts is spoken words contained in the audio file, textual transcriptions of these audio files are generated. We follow the same approach as Clifton et al. [CRY+20], the authors of "100,000 Podcasts: A Spoken English Document Corpus", to generate transcriptions. We use a two-minute sliding window with a one-minute overlap to create audio segments that will be transcribed. This results in 48138 segments created from the audio files of 1164 podcast episodes. To transcribe the segments, Clifton et al. use Google's cloud Speech-to-Text API[10] and achieve a word error rate (WER) of 18.1%.

[10] https://cloud.google.com/speech-to-text/docs/transcribe-audio-from-video-speech-to-text

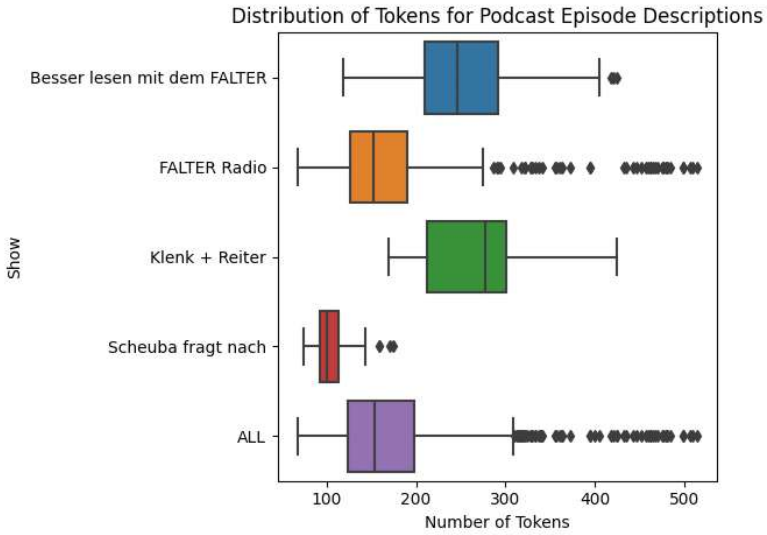

Figure 4.5: Boxplot of number of tokens from the tokenized podcast episode descriptions per show.

WER is a commonly used metric to evaluate *automated speech recognition (ASR)* systems. It is based on the *Levenshtein distance* and gauges how close the machine-generated transcription is to a reference transcription. We define the number of words substituted in the automatic transcription as S, the number of words in the reference transcription that were deleted in the automated transcription as D, and the number of words that were not present in the reference transcription but were inserted into the automated transcription as I. The WER is then calculated as

$$WER = \frac{S + D + I}{N}$$

where N is the total number of words in the reference transcription [MMD+04].

Since the WER of 18.1% is comparably high and the Google Speech-to-Text API charges costs for each transcription request, we decide to employ another *ASR* system for our purposes. In [RKX+22] Radford et al. present OpenAI Whisper, a Transformer-based multilingual open source *ASR* model that achieved state-of-the-art performance in transcribing popular speech recognition datasets such as Common Voice [ABD+19] or Multilingual LibriSpeech (MLS) [PXS+20]. The largest variant of the model *Whisper large-v2* achieved a WER of 6.4% and 5.5% when transcribing the German language from the Common Voice and Multilingual LibriSpeech datasets. Given the extensive computing time required to use the largest version of the model, we opt instead for the smaller model, *Whisper medium*, which still yields WERs of 8.5% and 7.4% for both datasets, while it only requires half the time for transcription. The application of the model to all podcast segments in our dataset sequentially results in a total computation

time of 255.075 hours on a NVIDIA GTX 1080 *graphics processing unit (GPU)*. A word cloud of the 200 most frequent words in the generated transcriptions is shown in Figure 4.6. The size of a given word corresponds to its relative frequency of occurrence.

Figure 4.6: Word cloud of the 200 most frequent words in the generated podcast segment transcriptions.

To evaluate the quality of the transcriptions, we randomly select five podcast segments from each show, making a total of 20 segments. We then manually create reference transcriptions for these segments and compare them with the transcriptions generated by the *ASR* system. This yields a *WER* of 4.7% which is 13.4% lower than it was reported for the dataset in [CRY+20]. The *WERs* for each podcast show and all combined shows are presented in Table 4.4.

Show	WER (%)
Besser lesen mit dem F.	2.726
FALTER Radio	4.516
Klenk + Reiter	9.372
Scheuba fragt nach	3.206
All shows	4.670

Table 4.4: Word error rates (WER) of five randomly sampled transcriptions for each show that were transcribed using the *Whisper medium automatic speech recognition* model.

During the creation of the reference transcriptions, we identify that most of the transcription errors are due to dialect (i.e. Austrian and Viennese), noisy pronunciation (e.g. *Background* is transcribed as *Backwand*), repeated words (i.e., if a speaker repeats a word multiple times, it is only transcribed one time), names of people and objects (e.g. *Blauensteiner* is transcribed as *Blondsteiner*) or speakers who are speaking at the same time. Filler sounds like “uh”, “ah”, or “um” are not transcribed by the model and were also not transcribed for the reference transcriptions. Additionally, we identify that some of the podcast segments contain music. Music does not reflect the topics discussed on the podcast and should not be used to represent an episode. If the music contains lyrics,

the ASR model partially transcribes them. If the music is only instrumental, the ASR model does not produce a transcription.

After the generation of segment transcriptions, we apply word tokenization to the transcription texts, as we also did for the textual attributes of the metadata. Table 4.5 presents the statistics and Figure 4.7 illustrates a boxplot of tokens from the transcriptions for each show and all shows combined.

Show	Mean	Std.	Q 25%	Q 50%	Q 75%	Min	Max
Besser lesen mit dem F.	370.278	58.910	338.0	378.0	409.0	123.0	539.0
FALTER Radio	319.733	59.672	282.0	321.0	360.0	1.0	542.0
Klenk + Reiter	309.405	50.892	291.0	316.0	340.0	39.0	430.0
Scheuba fragt nach	330.209	68.834	273.0	332.0	385.0	112.0	528.0
All Shows	323.170	61.331	284.0	324.0	365.0	1.0	542.0

Table 4.5: Statistics of the number of words in the podcast segment transcriptions for each show.

As shown above, the average number of words in the transcriptions of all podcast shows is 323.170, with a standard deviation of 61.331. This is comparable to the transcriptions generated in [CRY+20], where the authors report an average word count of 340 with a standard deviation of 70. One possible explanation for the slightly lower average word count in our dataset is the longer word length in German, as discovered in [Smi12]. Another reason may be the difference in speaking speed between English and German, as described in [CODP19]. The number of words per minute in the podcast segment transcription dataset is reported in Table 4.6.

4.5 Topic Modeling

To gain a deeper understanding of the dataset, we utilize topic modeling methods. The objective of these methods is to extract clusters of topics from the text corpus and assign individual documents (i.e. transcriptions of podcast snippets) to these clusters in an unsupervised manner. As there are various topic modeling methods available, we utilized three different techniques and compared their results. In the following sections, we explain

Show	Words per Minute
Besser lesen mit dem F.	185.139
FALTER Radio	159.866
Klenk + Reiter	154.703
Scheuba fragt nach	165.105
All shows	161.585

Table 4.6: Number of words per minute for the podcast segment transcript dataset.

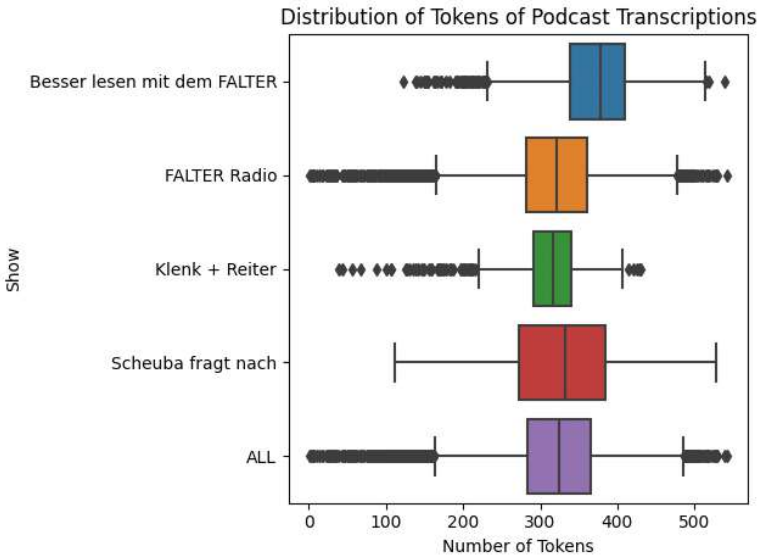

Figure 4.7: Boxplot of number of words in podcast segment transcriptions per show.

two conventional topic modeling methods that employ term frequencies to extract topics, namely Latent Dirichlet Allocation (LDA) [BNJ03] and Nonnegative Matrix Factorization (NMF) [PT94], and one method that utilizes text semantics by employing Transformer models (BERTopic) [Gro22]. The following topic modeling techniques are only applied to the generated podcast segment transcriptions, as it was also done in [CRY+20].

4.5.1 Latent Dirichlet Allocation (LDA)

LDA is a popular probabilistic generative model that was first introduced by Blei et al. in 2003 [BNJ03]. We select LDA as the first topic modeling method, as this was also the method that was used in [CRY+20]. Before delving deeper into the concept of LDA, we will first define the following terms:

- A *word* w (often also called *token* in the area of text processing) is the smallest data unit of text and defined as an item from a *vocabulary* V. The word at position v in the *vocabulary* is denoted as w^v.
- A *document* $\boldsymbol{w}$ is defined as a sequence of words of length N, where w_n is the word at position n in $\boldsymbol{w}$.
- A *corpus* D is a collection of M *documents* $\boldsymbol{w}$. The length of the d-th document $\boldsymbol{w}$ in the corpus D is denoted by N_d.

The LDA model is based on two assumptions: i) Documents in a text corpus are composed of a distribution of K topics, where K is assumed to be known and fixed, and ii) Topics

are represented by a distribution of different words. LDA models a corpus D using the following generative process for each document $\boldsymbol{w}$:

1. Choose a topic distribution $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})$, where $\text{Dir}(\cdot)$ is a Dirichlet distribution with scaling parameters $\boldsymbol{\alpha}$.
2. For each word w_n ($n \in \{1, ..., N_d\}$) in the document $\boldsymbol{w}$:
   a) Choose a specific topic $z_n \sim \text{Multi}(\boldsymbol{\theta})$, where $\text{Multi}(\cdot)$ is a multinomial distribution.
   b) Choose a specific word w_n from $p(w_n|z_n, \boldsymbol{\beta})$, a multinomial probability conditioned on the topic z_n.

In other words, this means that for each word w_n in each document $\boldsymbol{w} \in D$, a particular topic is chosen based on the document-topic distribution $\text{Multi}(\boldsymbol{\theta})$, where the probability distributions are parameterized by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. This process generates a mixture of topics with different probabilities for a document. With parameters given for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the joint multivariate distribution of a mixture of topics $\boldsymbol{\theta}$, a set of K topics $\boldsymbol{z}$, and a document $\boldsymbol{w}$ is computed as follows:

$$p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{n=1}^{N} p(z_n|\boldsymbol{\theta}) p(w_n|z_n, \boldsymbol{\beta}) \tag{4.1}$$

By integrating $\boldsymbol{\theta}$ and summing over $\boldsymbol{z}$, the marginal distribution of a document can be obtained. Taking the product of the marginal probabilities of each document in D, one obtains the following formula for the entire corpus:

$$p(D|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{d=1}^{M} \int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\boldsymbol{\theta_d}) p(w_{dn}|z_{dn}, \boldsymbol{\beta}) \right) d\boldsymbol{\theta}_d \tag{4.2}$$

[BNJ03, JWYF17, Ree12].

Within this research work, the LDA implementation[11] of the Python topic modeling framework Gensim [ŘS10] is utilized. Before feeding the tokenized podcast snippet transcriptions to the model, the following pre-processing steps are applied:

- Conversion of tokens to lowercase.
- Removal of tokens that contain only one character.
- Removal of German stopwords (e.g. “aber”, “bei”, “ein”, etc.) contained in the NLTK package.

[11] https://radimrehurek.com/gensim/models/ldamodel.html

- Stemming of tokens, i.e. reducing tokens to their root form to capture their core meaning, by using the Porter stemming algorithm [Wil06].

The next sections describe the selection of LDA parameters and the results obtained by executing the model.

Selection of Parameters

As already mentioned, the LDA model is parameterized by $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and the number of topics K. As the purpose of applying the model is only to get a better understanding of the present data, we will use Gensim's default settings for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, where those parameters are automatically estimated. Our focus will be on selecting an appropriate number of topics K.

Determining the appropriate number of topics K in topic modeling is a research area in its own way. While Blei et al. employ *perplexity* as a measure to select K in [BNJ03], Zhao et al. [ZCP+15] propose a different method for selecting K utilizing the *rate of perplexity change (RPC)* which achieves better results for various datasets. *Perplexity* is a frequently used measure in information theory to assess the effectiveness of a statistical model in describing a dataset [ZCP+15]. The *perplexity* of a model given a test set of documents D_{test}, can be calculated as follows:

$$\text{perplexity}(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(\boldsymbol{w}_d)}{\sum_{d=1}^{M} N_d}\right\} \text{[BNJ03].} \tag{4.3}$$

Using the definition of *perplexity* in 4.3, the *RPC* can now be calculated with the following formula:

$$\text{RPC}(i) = \left|\frac{P_i - P_{i-1}}{t_i - t_{i-1}}\right| \tag{4.4}$$

where t repesents a sequence of numbers of topics sorted in ascending order, and P represents a sequence of calculated perplexities, where P_i is the perplexity for a particular number of topics t_i [ZCP+15]. Since [ZCP+15] suggests better results when using *RPC* for the determination of K, we utlize this method for a sequence of $\{5, 6, ..., 100\}$ different numbers of topics. The respective results of this method are shown in Figure 4.8. The authors state that an appropriate number of topics can be found at the point where the RPC-curve first changes its slope sign. In Figure 4.8, we can see that this occurs at the $K = 6$ topics, which is a much smaller number than we expected.

Due to numerous subsequent changes in the slope sign observed in the RPC-curve, it is difficult to identify the appropriate number of topics for the given dataset. Thus, we employ a different measure to determine K which is not based on perplexity. Therefore, we use an alternative measure to determine K that is not based on perplexity.

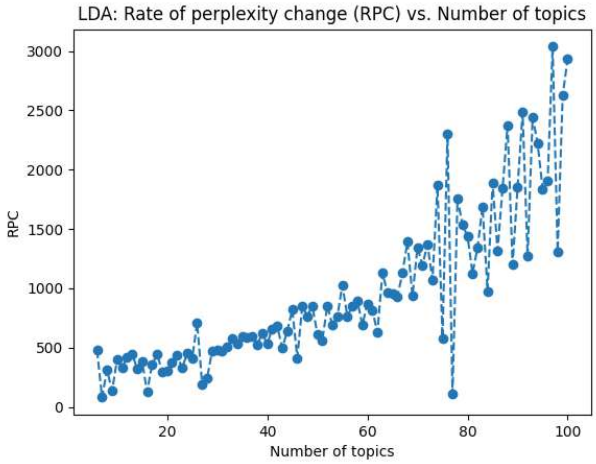

Figure 4.8: Rate of perplexity change (RPC) vs. Number of topics for the podcast segment LDA model.

Although perplexity is the method recommended by LDA creators to determine K, it does not accurately reflect the semantic relatedness of documents within a topic. This can lead to the introduction of irrelevant or unrelated topics [GMMCQ21]. Studies such as [NNT+10, GMMCQ21] that compare the use of perplexity-based and coherence-based methods to determine the number of topics have found that perplexity is sometimes even contrary to human judgment. The coherence of a topic model is often synonymous with human comprehension and interpretability of the structure of a topic. The coherence metric rates the semantic relatedness of words within a topic based on the co-occurrence statistics of words in the corpus [GMMCQ21]. There are multiple approaches available to calculate the coherence of a fitted topic model. In this work, we select the so-called C_v-coherence, as it showed the most promising results in [RBH15]. The calculation of C_v is based on Normalized Pointwise Mutual Information (NPMI) [Bou09], which provides a way to calculate the probability that two words occur together in a corpus. The core idea of C_v is to compute a word vector for each word in a topic by computing the probability of co-occurrence in the corpus with all other words in the topic as follows:

$$\vec{w}_{n,k} = NPMI(w^T_{n,k}, w^T_{m,k}) \forall m \in 1, 2, ..., N \tag{4.5}$$

where $w^T_{n,k}$ is the n-th word in topic k and N is the selected number of words in the topic used for the calculation. Based on these word vectors, topic vectors can be calculated by summing up all word vectors in a topic:

$$\vec{w}^*_k = \sum_{n=1}^{N} \vec{w}_{n,k}. \tag{4.6}$$

Now the C_v-coherence score of a topic model can be computed as

$$C_v = \frac{1}{NK} \sum_{k=1}^{K} \sum_{n=1}^{N} s_{cos}(\vec{w}_k^*, \vec{w}_{n,k}) \tag{4.7}$$

where $s_{cos}(\vec{a}, \vec{b})$ is the cosine similarity defined as

$$s_{cos}(\vec{a}, \vec{b}) = \frac{\vec{a} * \vec{b}}{||\vec{a}|| * ||\vec{b}||}. \tag{4.8}$$

An appropriate number of topics can be selected as the number of topics where a model yields the highest score for C_v [RBH15]. In this work, we fit LDA topic models for a sequence of $\{5, 6, ..., 100\}$ numbers of topics and calculate their respective coherence scores. Due to the resources required to fit a new LDA model for each value of K, we include an early stopping mechanism that stops fitting new models if the coherence did not increase for 30 iterations.

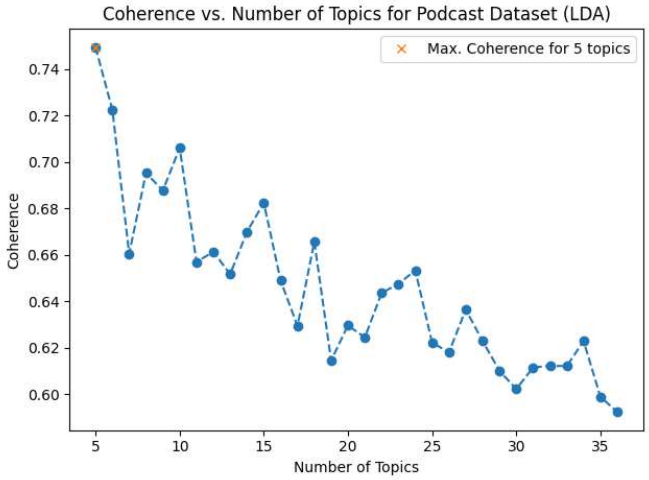

Figure 4.9: Coherence (C_v) vs. Number of topics for the podcast segment LDA model.

In Figure 4.9, we can see that the determination of the number of topics based on coherence leads to $K = 5$ topics, which is even lower than the number of topics we determined when using RPC. Although this is already an indicator that LDA is not well suited for modeling the topics of the podcast dataset, the identified topics will be investigated in the following section.

Identified Topics

As coherence is a measurement that reflects the human interpretability of a topic model, $K = 5$ is the value of choice for the investigation of the identified topics.

Figure 4.10: Word clouds of the $K = 5$ identified topics using LDA on the podcast segment dataset.

Looking at the word clouds of the five identified topics shown in Figure 4.10, we can see that topic 0 contains tokens in English. This is an unexpected result, since the recording language of the podcasts was assumed to be German only, and should be remembered for data cleaning in subsequent steps of this work.

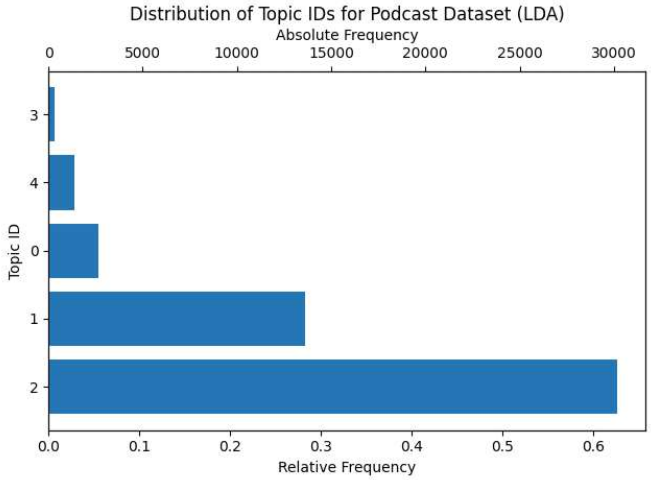

Figure 4.11: Distribution of documents between identified topics for the LDA model fitted on the podcast segment dataset.

As expected, the other topics do not show relevant information and seem to contain random words. Only the fourth topic shows some information, possibly from the intro and outro sections of the podcasts, but looking at how the documents are distributed among the different topics, as shown in Figure 4.11, one can see that most of the records are identified as the first and second topics, and therefore this information is negligible.

4.5.2 Nonnegative Matrix Factorization (NMF)

Nonnegative Matrix Factorization (NMF) is a popular dimensionality reduction technique that was first introduced by Paatero and Tapper [PT94] in 1994. Since dimensionality

reduction is closely related to topic modeling and text clustering, NMF is often applied to natural language processing problems [KCP15].

Given a nonnegative term document matrix $A \in \mathbb{R}_+^{n \times m}$ where the columns represent the documents and the rows represent the terms in a vocabulary, the goal of NMF is to find two lower rank matrices $W \in \mathbb{R}_+^{n \times k}$ and $H \in \mathbb{R}_+^{k \times m}$ such that A is approximated by

$$A \approx WH \tag{4.9}$$

where W represents a term-topic matrix, H represents a topic-document matrix and k (which was denoted as K in the context of LDA) is the number of topics assumed to satisfy $k < min\{n, m\}$. The matrices W and H can be found by solving the following optimization problem based on the Frobenius norm, a distance measure between matrices (other measures are also possible but less common):

$$\min_{W \geq 0, H \geq 0} ||A - WH||_F^2. \tag{4.10}$$

A column in W now represents a topic as a weighted combination of n words in the vocabulary. A column in H represents a document as a weighted combination of topics [KCP15, CLRP13]. This makes it similar to the structure of the output of the LDA model, except that the output of the NMF model is not column normalized [CLRP13].

Similarly to the LDA topic model, the Gensim topic modeling framework [ŘS10] is used for implementation[12]. Again, lowercasing, single token removal, stop word removal, and stemming are performed before applying the model.

The following sections will now describe the selection of parameters and the topics identified by the NMF topic model.

Selection of Parameters

Analogous to the LDA model, we do not modify any parameters of the Gensim implementation, except for the number of topics. Since the topics identified by LDA appeared to be relatively random, we skip using *RPC* for the selection of the number of topics and again choose to use C_v-coherence for a sequence of $\{5, 6, ..., 100\}$ number of topics with an early stopping mechanism with a patience of 30 iterations.

In Figure 4.12 we can see that the topic model with seven topics gives the highest coherence score. We can also observe that the coherence scores obtained by NMF are generally higher than those for LDA. Although seven is a higher number of topics than the five we found for the LDA topic model, we think it is still too low to capture all the topics we expect to be included in the podcast segment transcriptions. However, in contrast to the LDA topic model, we can also observe some peaks in the coherence values for higher numbers of topics, such as 12, 19, or 29. It might be worthwhile to investigate

[12] https://radimrehurek.com/gensim/models/nmf.html

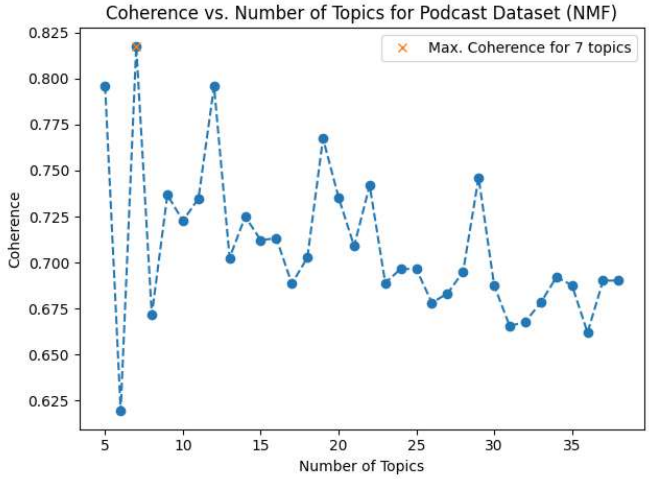

Figure 4.12: Coherence (C_v) vs. Number of topics for the podcast segment NMF model.

these peaks in research that focuses on comparing topic modeling techniques. Due to time constraints and because this is beyond the scope of this work, we will skip this and focus on the findings of the seven-topic model described in the following section.

Identified Topics

The topics identified with the NMF model, shown in Figure 4.13, also look random and do not appear very descriptive, as did they when using the LDA model.

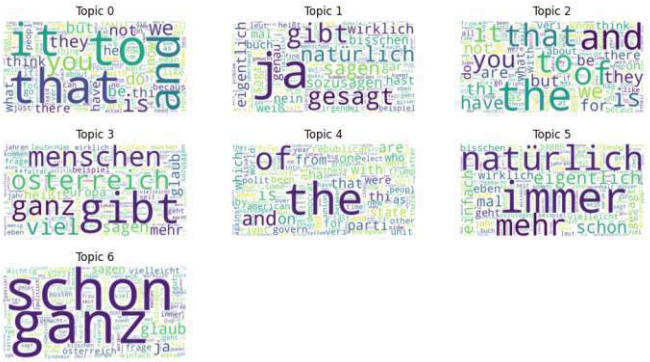

Figure 4.13: Word clouds of the seven identified topics using NMF on the podcast segment dataset.

We can see that in contrast to the LDA model, where one identified topic contained English terms, for NMF even three topics (zero, two, four) consist entirely of English. This highlights the need for data cleaning methods that identify and remove these podcasts or

podcast segments. Otherwise, we cannot extract any other valuable topical information from the NMF model than from the LDA model.

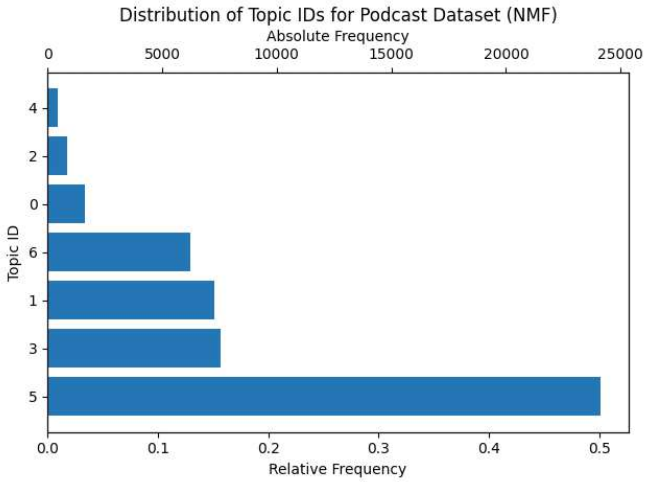

Figure 4.14: Distribution of documents between identified topics for the NMF model fitted on the podcast segment dataset.

Looking at the distribution of podcast segment transcriptions between topics for the NMF model, shown in Figure 4.14, we can see that also like for the LDA model, a large part (approximately 50%) of the documents are assigned to one particular topic (with ID five). Topics with IDs one, three, and five contain almost equally many documents with a share of 10-15%. The remaining documents (approximately 8%) are distributed among topics zero, two, and four, which contain English.

4.5.3 BERTopic

Since topic modeling with LDA and NMF did not yield satisfactory results, we utilize another technique that is based on Transformer models: BERTopic. BERTopic [Gro22] is a topic modeling technique introduced by Grootendorst in 2022 that approaches topic modeling as a clustering task. The author states that it produces coherent topics and is competitive on benchmark datasets using a multi-step process shown in Figure 4.15. The process begins with the creation of vector representations of documents using the Sentence-BERT (SBERT) framework [RG19]. The SBERT framework includes code and models that are tuned to produce semantically meaningful sentence embeddings that can be compared with cosine similarity to retrieve semantically similar texts or sentences.

Since the distance measures used in clustering may vary little with increasing data dimensionality, the next step of BERTopic is to apply UMAP [MHM20] to the previously created document embeddings. UMAP is a dimensionality reduction technique that has been shown to preserve more local and global features of high-dimensional data than

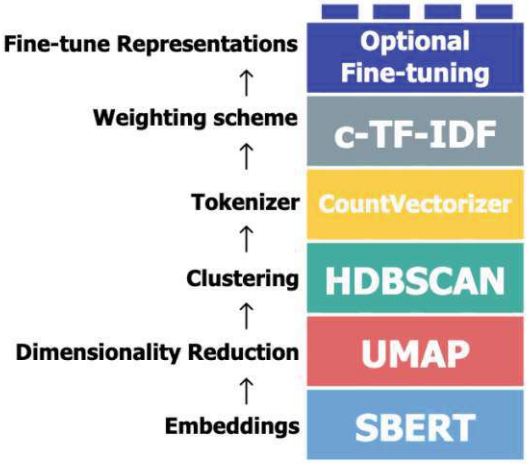

Figure 4.15: Multi-step topic modeling process employed by BERTopic [Gro].

other well-known dimensionality reduction techniques such as PCA [WEG87] or t-SNE [VdMH08] when projecting data to lower dimensionality [Gro22].

After dimensionality reduction, hierarchical density-based spatial clustering of applications with noise (HDBSCAN) [CMS13] is applied to the data. Each resulting cluster then represents an identified topic in the text corpus. The number of clusters (i.e. topics) is automatically determined by the algorithm based on the cluster densities. Additionally, the HDBSCAN algorithm identifies noisy document representations and adds them to a separate noise cluster. After clustering, a count vectorization of all documents in a cluster is performed. This means that all documents belonging to a particular cluster are concatenated into a single document representing a cluster c, this document is then divided into individual terms and the term frequency tf of each term t in the cluster document ($tf_{t,c}$) and in all documents (tf_t) is calculated. With these frequencies, a term of a particular cluster can be weighted with respect to its descriptiveness for the cluster as follows:

$$W_{t,c} = tf_{t,c} * \log(1 + \frac{A}{tf_t}) \tag{4.11}$$

where A is the average number of terms in a cluster [Gro22]. The author refers to this weighting statistic as class-based term frequency-inverse document frequency (c-TF-IDF), a modification of the classical term frequency-inverse document frequency (TF-IDF) [J+97], which measures the importance of a word to a document. A topic identified by BERTopic can now be represented by the N words that produce the highest c-TF-IDF score for a given cluster. As a final and optional step, BERTopic also provides several ways to further fine-tune the representations of a topic, but this step is not performed in this work.

Similarly as for the LDA and NMF topic models, we perform lowercasing, single token

removal, stop word removal, and stemming before applying the Python implementation of BERTopic[13] to the podcast transcriptions.

The following sections describe parameter selection and the topics identified by BERTopic.

Selection of Parameters

Since the number of topics is inherently selected by the HDBSCAN clustering algorithm, we do not search for the number of topics using *RPC* or *topic coherence* as we did for LDA and NMF. Except for setting a fixed random seed of 42, setting the language to *multilingual*, and increasing the number of N words to represent a topic from 10 to 200 to match the LDA and NMF topic visualizations, we use all the default parameters of the BERTopic Python implementation. The multilingual language setting is the only option currently supported by BERTopic other than English and results in the use of the *paraphrase-multilingual-MiniLM-L12-v2*[14] model for generating text embeddings. This SBERT model maps text to a 384-dimensional vector space, is fine-tuned to 50 languages, and works well in clustering or semantic search applications. The number of topics selected by HDBSCAN when applying BERTopic to the data is 277, where one topic with ID minus one contains podcast transcriptions that were identified as outliers.

Identified Topics

As it is not possible to visualize all 277 identified, we focus on the 30 topics to which most of the documents were assigned shown in Figure 4.16.

We can clearly see that the topics identified by BERTopic look much more coherent than when LDA and NMF were applied to the data. We can observe topics ranging from global issues like the war in Ukraine (topic zero), Israel (topic six), the Corona pandemic and vaccination (topic 10), or the climate crisis (topics 12 and 17), to local Austrian issues like media and journalism (topics three and 13), or political parties (topics nine, 21, and 26). We can also see that there are topics that are closely related to the podcast format. Topics seven and 20, for example, probably have podcast snippets associated with them that correspond to intro or outro sections of the podcast. Topic 15 contains terms that probably originate from the podcast "Besser Lesen mit dem FALTER". We can also observe English topics (e.g. topic 16), as we did for the other two topic modeling approaches.

In Figure 4.17, which shows the distribution of documents among the top 30 topics, we can see that most of the podcast transcriptions were assigned to the outlier topic with ID minus one. Since more than 60% of the segment transcriptions are assigned to this topic, we do not expect all of them to be complete outliers, but rather to provide less distinct information signals than other segment transcriptions assigned to regular topics.

[13] https://maartengr.github.io/BERTopic/

[14] https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

Furthermore, it can be observed that the topic with the second most assigned documents is topic zero with about nine percent of all documents. It can be seen that for topics with subsequent IDs (one, two, etc.), the proportion of segment transcriptions assigned to them decreases steadily until it approaches an almost uniform distribution. This again indicates that BERTopic has identified many small but quite concrete topics.

BERTopic showed the most promising results of all applied topic modeling techniques. However, we are still cautious about the results, as the outlier topic has a rather large number of documents assigned to it.

4.6 Language Classification

Given that topics containing English terms are identified by all three applied topic modeling techniques, we decide to use a text classification model to classify the language of the podcast segment transcriptions. For this purpose, we use a Transformer-based model provided on Huggingface[15]. The basis of the model is XLM-RoBERTa [CKG+19], a multilingual version of RoBERTa pre-trained on a filtered Common Crawl[16] containing 100 languages. Common Crawl is a large uncurated dataset of web pages in many languages, collected in periodic snapshots [WLC+20]. After pre-training on the Common Crawl, the model was fine-tuned on the task of classifying text into 51 different languages. For fine-tuning, the MASSIVE dataset [FHP+22] was used, which contains more than one million text utterances in 51 languages. We choose this model for its ease of implementation given its availability on Huggingface and the excellent classification results reported by the author. Figure 4.18 shows the classified languages and how the segment transcriptions are distributed among these languages when applying the above model. Note that the x-axis of the graph is log-transformed to emphasize small frequencies of documents classified as having a particular language.

We can see that, as already suggested by the results of the topic modeling, there are many transcriptions that are classified as non-German. In total, the model classified 18 different languages. German (de-DE) is the most prominent with 45240 records, followed by US English (en-US) and Welsh (cy-GB) with 2634 and 120 records, respectively. The other languages were classified less than 100 times, with some languages like Italian (it-IT) or Finnish (fi-FI) having only one segment transcription assigned. This result is surprising to us because we expected that the data would only contain German. Therefore, we use the classification results to remove the non-German transcripts in subsequent steps.

[15] https://huggingface.co/qanastek/51-languages-classifier
[16] https://commoncrawl.org/

Figure 4.16: Word clouds of the top 30 (of 277) identified topics using BERTopic on the podcast segment dataset.

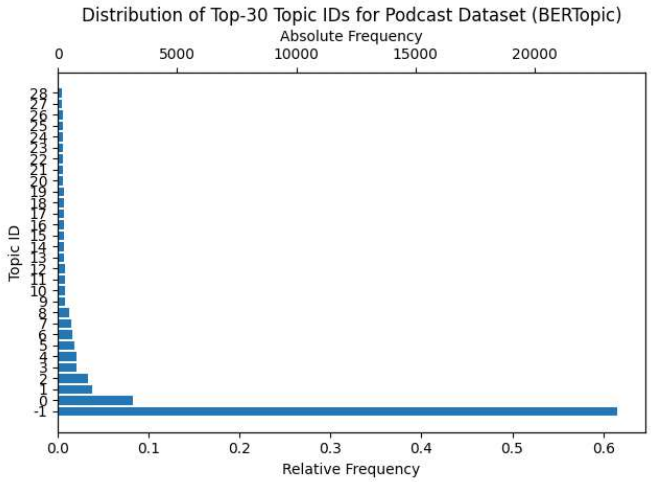

Figure 4.17: Distribution of documents between top 30 (of 277) identified topics for the BERTopic model fitted on the podcast segment dataset.

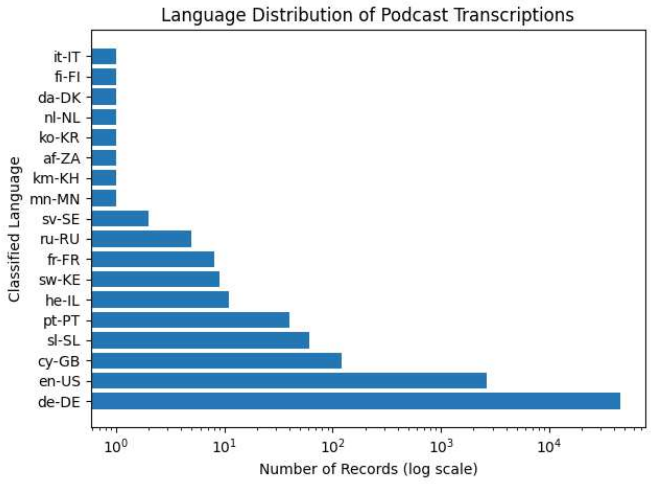

Figure 4.18: Distribution of podcast segment transcriptions between classified languages.

CHAPTER 5

News Article Dataset

This chapter describes the news article dataset used in this work. The dataset was also kindly provided by the industry partner *FALTER* and consists of 119219 news articles from August 1998 to May 2023. In the following sections, we first report and describe statistical measures for the key components included in the provided dataset. We then conduct an in-depth exploration of the textual attributes (i.e. titles and paragraphs) of the dataset using topic modeling and classification methods.

5.1 Components

The present dataset contains 20 attributes for each of the 119219 articles. The attributes include an article ID, a publication date, authors, a department (referred to as *ressort* in the dataset), references to images and other articles, and most interestingly, titles and paragraphs that contain the content of the articles. Since not all attributes are of interest for this research, we examine only the publication dates, ressorts, titles, and paragraphs of the dataset in the following sections.

5.1.1 Publication Date

As mentioned above, the first news article in the dataset was published in August 1998. In Figure 5.1 we can see the number of news articles published over time on a quarterly frequency.

It can be seen that the number of published articles increases slightly but steadily until the fourth quarter of 2008. In this quarter, *FALTER* publishes almost twice as many articles as in the previous quarter. This high publication frequency continues for the next time but decreases again slightly until the time approaches the second quarter (June) in 2023. It is important to note that on the x-axis in Figure 5.1, the labels show only the last month of each second quarter, but the graph shows cumulative counts for each

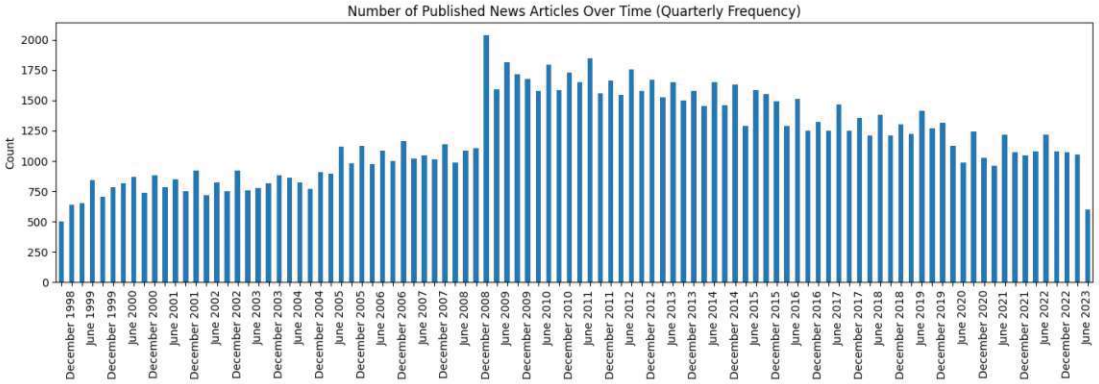

Figure 5.1: Development over time of news article publications on a quarterly frequency.

quarter. Thus, August 1998, which falls in the third quarter of 1998, is not visible in the axis labels.

5.1.2 Ressort

The news article dataset contains 380 distinct ressorts (i.e. departments) among which the articles are distributed. After we lowercased the ressort names, the number of distinct values reduces to 316. Figure 5.2 shows the distribution of news articles between the top 30 most frequent lowercased ressorts.

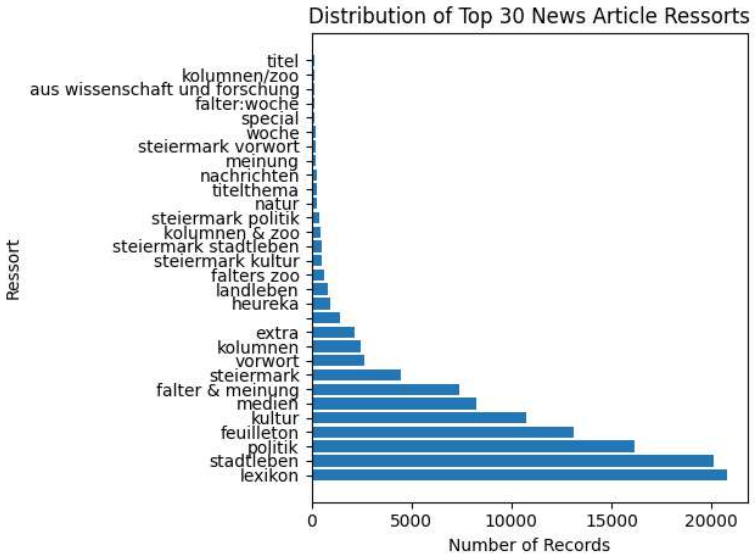

Figure 5.2: Distribution of news articles between top 30 most frequent ressorts.

One can see that the most articles are published under the ressorts *lexicon, stadtleben,*

Mean	Std.	Q 25%	Q 50%	Q 75%	Min	Max
4.899	3.410	2.0	4.0	7.0	1.0	206.0

Table 5.1: Statistics of the number of tokens of news article titles.

politik, feuilleton and *kultur*. Besides that, the graph shows that there are approximately 2000 articles that have an empty string as a value for the ressort attribute.

5.1.3 Title

We now examine the titles of the news articles. As in the analysis of the textual attributes of the podcast dataset, we start with word tokenization using the NLTK word tokenizer. The length of the titles ranges from one to 206 tokens, with a mean of 4.899 and a standard deviation of 3.410. More statistical measures of the number of tokens in the titles of the news articles are shown in Table 5.1.

In Figure 5.3, which shows a boxplot of the number of tokens in the titles of news articles, we can see that there are excessively long titles in the data that could be considered outliers. By manually investigating some of them, we attribute this to the writing style of individual authors, and thus see no need to exclude them from the dataset.

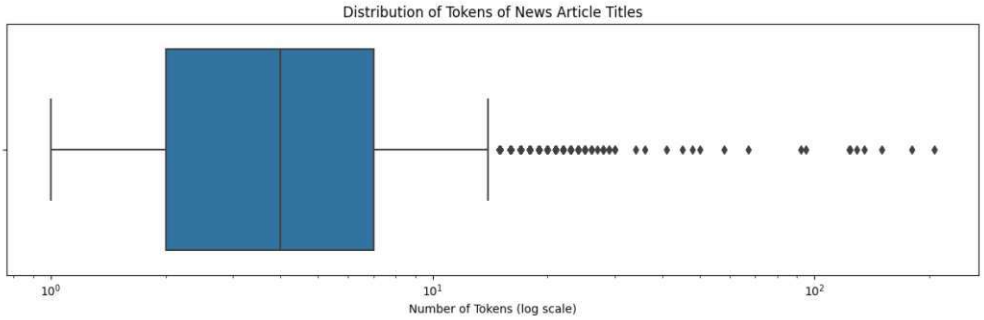

Figure 5.3: Boxplot of number of tokens in the news article titles.

5.1.4 Paragraphs

The paragraphs in the dataset can be thought of as the main source of content for a news article. As the name suggests, this attribute is a list of the paragraphs contained in a given article. In this dataset, an article has a mean of 9.231 paragraphs and a standard deviation of 12.212 paragraphs. We merge all paragraphs of an article into a single string and then perform word tokenization, as we did for all other textual attributes. The length of the article paragraphs ranges from zero (empty article) to 65645, with a mean of 532.509 and a standard deviation of 606.798. This shows that the length of an article can vary widely and should be taken into account for subsequent processing steps. Table

Mean	Std.	Q 25%	Q 50%	Q 75%	Min	Max
532.509	606.798	183.0	340.0	653.0	0.0	65645.0

Table 5.2: Statistics of the number of tokens of merged news article paragraphs.

5.2 shows more statistical measures of the number of tokens in the merged news article paragraphs.

In Figure 5.4, which shows a boxplot of the number of tokens in the merged paragraphs, we can also see articles with an excessive number of tokens. Since these can be problematic for modeling due to the limited number of tokens that Transformer models can process at once, we remove them in subsequent processing steps.

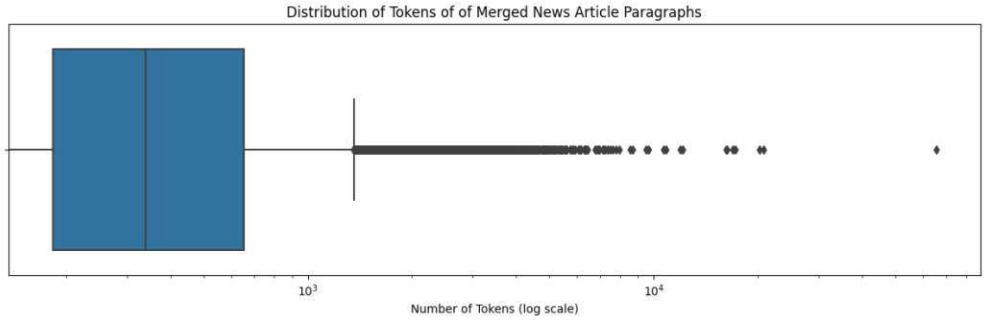

Figure 5.4: Boxplot of number of tokens in the merged news article paragraphs.

5.2 Topic Modeling

Similar to the podcast dataset, we apply topic modeling methods to the data. As before, we use two methods based on term frequency (LDA and NMF) and a Transformer-based method (BERTopic) and compare their results. For the news dataset, we use the merged paragraphs as the data source for topic modeling. Titles or other attributes are not considered.

5.2.1 Latent Dirichlet Allocation (LDA)

Before we feed the tokens of the merged news article paragraphs into the model, we apply similar pre-processing steps as when topic modeling the podcast transcriptions: Lowercasing, single token removal, stop word removal, and stemming. Again, we use C_v-coherence as a measure to find an appropriate number of topics by fitting models for a sequence $\{5, 6, ..., 100\}$ of topic numbers with an early stopping patience of 30 iterations. Since using *RPC* as a measure did not show promising results for the podcast dataset, we do not use this measure for the news article dataset. Again, we use the

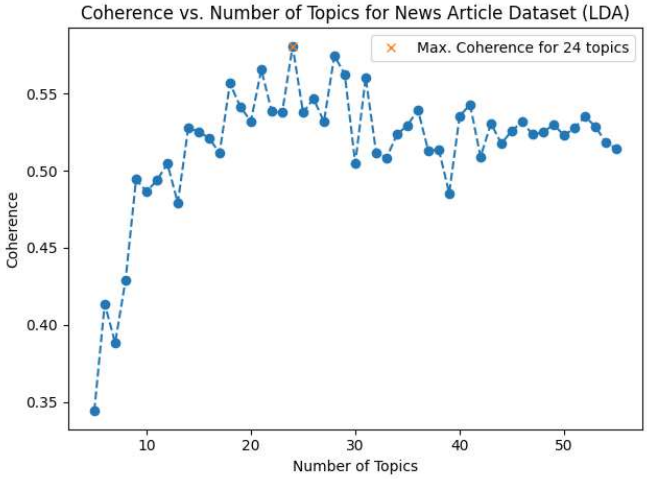

Figure 5.5: Coherence (C_v) vs. Number of topics for the news article LDA model.

default parameters of the Gensim LDA implementation. Figure 5.5 shows the resulting coherence scores for the sequence of numbers of topics.

It can be seen that the coherence-based selection of the number of topics for the LDA model results in $K = 24$ topics, which seems to be a more plausible number than the $K = 5$ selected when LDA was applied to the podcast transcriptions. Looking at the identified topics in Figure 5.6, it can be seen that although the coherence score for the news article dataset is lower than for the podcast transcriptions, the topics look much more coherent and relevant.

The topics identified by LDA for the news article dataset overlap with the topics identified by BERTopic for the podcast dataset. This is a good indicator that both datasets can be used to make content-based cross-domain recommendations. Furthermore, it is interesting to see that there are no non-German topics for the news articles, as was the case for the podcast transcriptions. Looking at how the documents are distributed between topics, as shown in Figure 5.7, we can see that the most prominent topic in the data (ID 21) is about books and movies, followed by topic five, which looks very similar to the outlier topic (ID minus one) identified by BERTopic for the podcast dataset.

5.2.2 Nonnegative Matrix Factorization (NMF)

For the news article dataset, we also perform topic modeling using NMF. We use similar pre-processing steps as for LDA, and also fit NMF models for a sequence $\{5, 6, ..., 100\}$ of numbers of topics, stopping model fitting when the C_v coherence score has not increased for 30 iterations. Figure 5.8 shows the resulting coherence scores for the sequence of numbers of topics.

Figure 5.9 shows a word cloud of topics identified by NMF.

Figure 5.6: Word clouds of the $K = 24$ identified topics using LDA on the news article dataset.

5.2.3 BERTopic

Finally, we also use BERTopic to model the topics contained in the news articles. We use similar pre-processing and parameters as for the podcast segment dataset. Applying BERTopic to the news article dataset results in 430 identified topics. Again, since not all of these topics can be visualized, we show word clouds of the 30 most prominent topics in Figure 5.11.

Also for BERTopic we can observe an overlap between the identified topics for the news article and for the podcast segment dataset. Comparing the identified topics between LDA, NMF and BERTopic for the news article dataset, a more congruent result can be

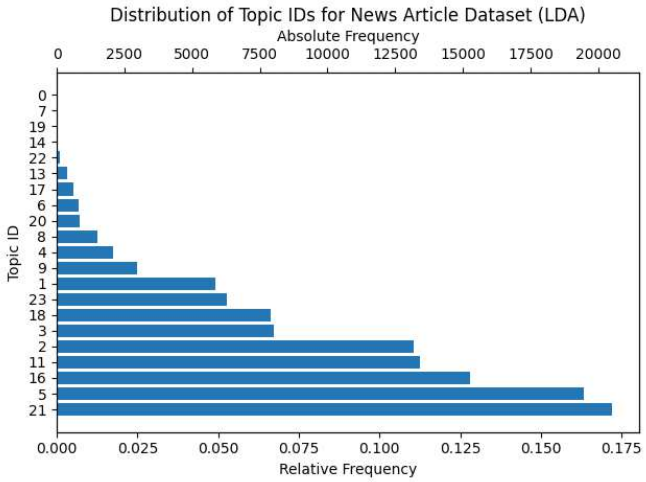

Figure 5.7: Distribution of documents between identified topics for the LDA model fitted on the news article dataset.

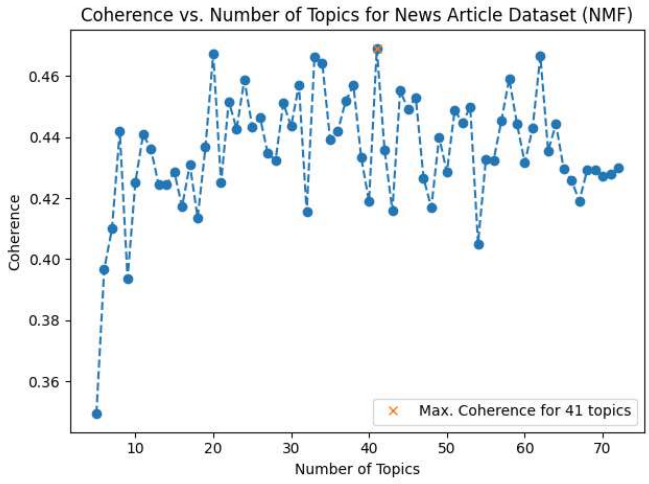

Figure 5.8: Coherence (C_v) vs. Number of topics for the news article NMF model.

observed than was possible for the podcast segment dataset. Looking at the distribution of articles among the top 30 topics shown in Figure 5.12, a similar structure can be seen as for the podcast segment dataset. Most of the documents are assigned to the topic with ID minus one, which corresponds to the outlier topic. For topics with subsequent IDs (one, two, etc.), the proportion of articles assigned to them steadily decreases until it approaches an almost uniform distribution, indicating many small but concrete topics identified by the model.

5.3 Language Classification

Although none of the topics identified by LDA, NMF, and BERTopic are completely non-German, as was the case with the podcast dataset, we still apply a similar language classification model to the news article data. Figure 5.13 shows the classified languages and the distribution of news article records between these languages. Again, the x-axis of the graph is logarithmically transformed to show small numbers of documents classified as a particular language.

We can see that despite the fact that no non-German topics were identified when modeling the topics of the news articles, the model classified a total of four different languages. As expected, the most prominent language is German (de-DE) with a total of 119206 records. The remaining 14 records are distributed between US English (en-US) with nine records, Russian (ru-RU) with four records, and Arabic (ar-SA) with one record. Looking at the records that are not classified as German, we can see that the data actually contains articles that are in English and Russian, but the article classified as Arabic does not contain any paragraphs and is therefore a false positive. Compared to the podcast segment dataset, the number of non-German records is negligible, but non-German records should nevertheless be removed in subsequent processing steps.

Figure 5.9: Word clouds of the top 30 (of 41) identified topics using NMF on the news article dataset.

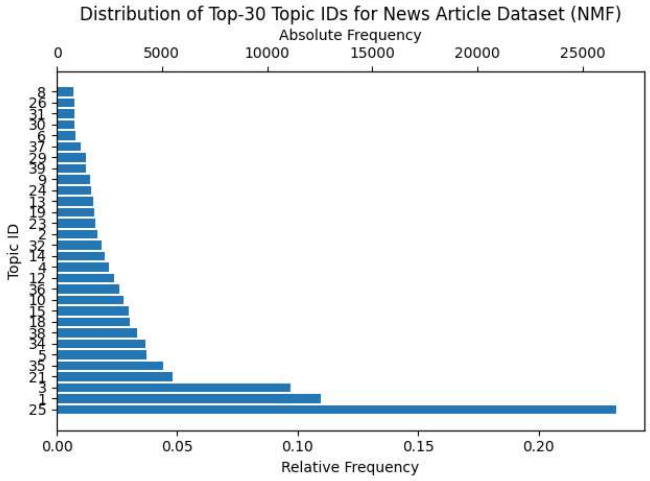

Figure 5.10: Distribution of documents between top 30 (of 41) identified topics for the NMF model fitted on the news article dataset.

Figure 5.11: Word clouds of the top 30 (of 430) identified topics using BERTopic on the news article dataset.

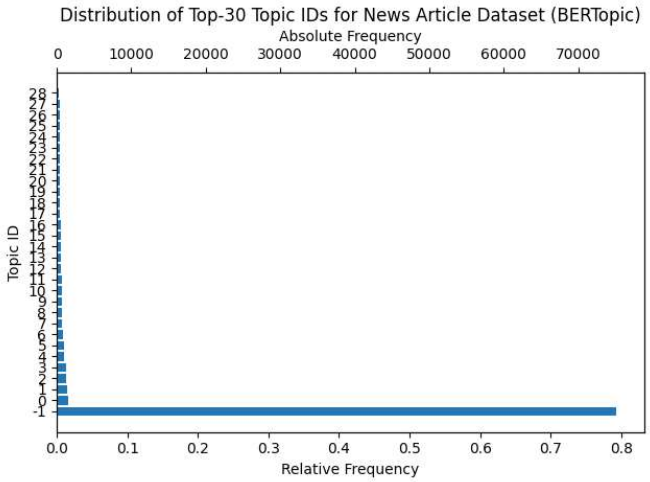

Figure 5.12: Distribution of documents between top 30 (of 430) identified topics for the BERTopic model fitted on the news article dataset.

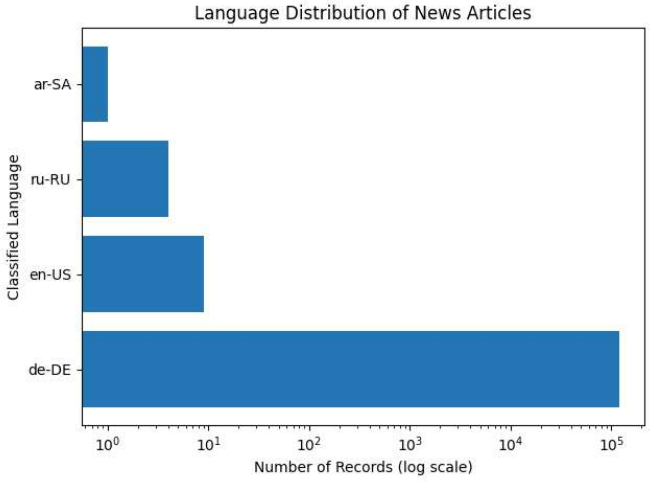

Figure 5.13: Distribution of news articles between classified languages.

CHAPTER 6

First Annotation Study

In this chapter, we describe the design, implementation, and results of the first annotation study. We begin by describing the general approach, followed by the selection of podcast segments. We then describe how we retrieve news articles for each of these segments, and end with an analysis of the results obtained in terms of inter-annotator agreement, relevance, timing, and annotator feedback.

6.1 Approach

The purpose of the annotation study is to build a dataset of podcast segment and news article pairs with labels that reflect the relevance between the two data components. Based on the results of the expert interview (see Chapter 3.2) and in order to reduce the complexity of the data annotation, we decide to use only a binary labeling approach with the labels *relevant* and *non-relevant*. Since one goal of this work is to build a cross-domain recommender system from podcasts to news articles, we only create a dataset in this direction. The same approach could be used to create a dataset from news articles to podcasts, but this is beyond the scope of this work.

To build the dataset, we begin by selecting podcast segments to use in the study (described in Chapter 6.2). For each of the selected segments, we then retrieve news articles that may be relevant to that segment, as described in Chapter 6.3.

Given the selected podcast segments with their potentially relevant news articles, we then create the annotation study using the online survey tool LimeSurvey[1]. We begin the study with questions that assess the academic background and knowledge of *FALTER*'s podcast and news article offerings, as well as questions that collect information about the frequency with which the annotators consume both types of media. We then present

[1] https://www.limesurvey.org/

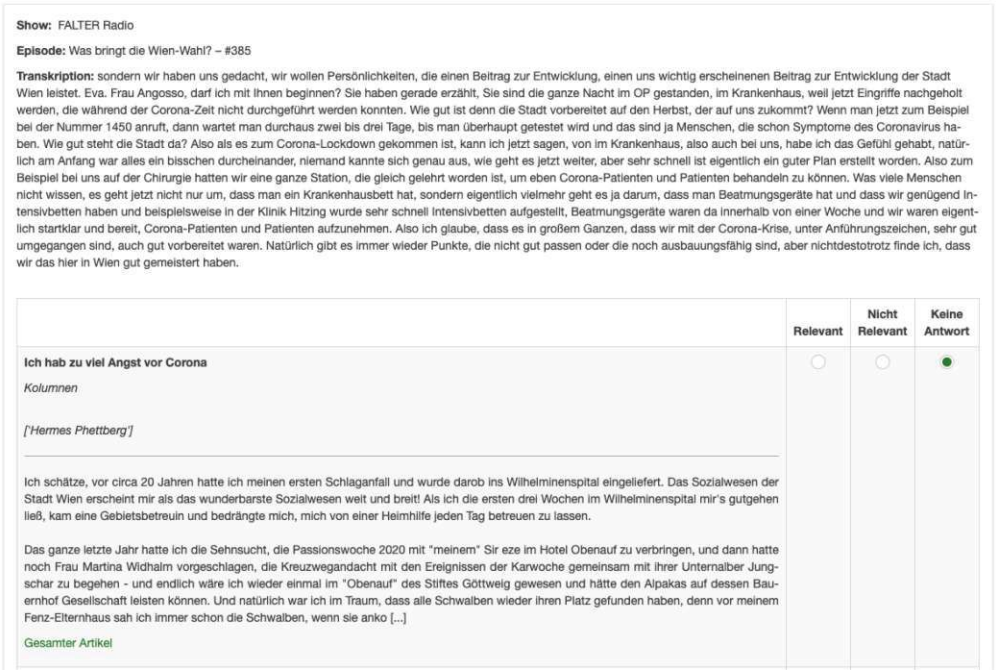

Figure 6.1: Exemplary question in the first annotation study between podcast segments and news articles.

a brief description of how to approach the annotation task, along with relevant and non-relevant podcast/news article pairs. Apart from the instruction that annotators should judge a pairing as *relevant* if they find the article useful for a given podcast segment, no further guidelines or annotation instructions are given.

Each of the following questions now includes the show title, episode title, and transcript of a selected podcast segment. Below the podcast segment information, a list of retrieved news articles is shown. For each article, the title, authors, department, publication date, and first 512 words are displayed, together with a link to the full text of the article. Next to each of the displayed articles is a radio button with the options to annotate the article as *relevant* or *non-relevant*. Each podcast segment news article pair is annotated by exactly three annotators, which gives us the opportunity to measure the agreement between annotators which then reflects the reliability of the performed annotations. Figure 6.1 shows an exemplary question that is presented to the annotators on LimeSurvey.

6.2 Selection of Podcast Segments

Selecting appropriate segments from the large collection of available podcast segments is critical to the success of the annotation study. We first of all perform cleaning of the dataset followed by sampling of podcast segments that are included in the annotation

study. Both steps are described in the following sections.

6.2.1 Data Cleaning

Due to the results of the analysis of the dataset described in Chapter 4, the cleaning of the dataset focuses on two main parts: Token statistics and language.

To identify podcast segments that are too short or too long, we use the distribution of the number of tokens in the transcribed text, which we assume to be normal. Assuming the distribution is normal, we know that for $k = 1.96$, 95% of the values are in the interval $[\mu - k * \sigma, \mu + k * \sigma]$ and that the values outside this interval are potential outliers. Here, μ is the mean number of tokens of the segment transcriptions and σ is the corresponding standard deviation. As is often done in practice, we use $k = 2$ instead of $k = 1.96$ as the boundary within the interval, which also gives this outlier identification method its name: *two-sigma rule* [NKST03]. Applying the two-sigma rule to the number of tokens in the segment transcriptions results in 2023 records being excluded from the dataset.

For language-based data cleaning, we use the results of the language classification model in Chapter 4.6. Excluding all records identified as non-German results in a total of 2898 records, of which 2651 records have not yet been excluded by the token statistics-based cleaning rule. Besides that, we do not perform any additional data cleaning for the first annotation study. It can be argued that the results obtained during topic modeling of the data could also be used for cleaning purposes, but since the results for LDA and NMF appeared to be random and BERTopic identified about 80% of the transcriptions as outliers, we refrain from doing so. This leaves us with a podcast segment dataset of 43464 records available for sampling.

6.2.2 Sampling

We randomly sample 10 podcast segments from each of the four podcast shows, using a random seed of 42. This results in a total of 40 podcast segments used in the annotation study. We choose to use random sampling to avoid introducing bias in the selection of segments. To keep the proportion of segments the same for all shows, we decide to sample each show separately. The number of podcast segments sampled is a trade-off between the variance of podcast segments included in the annotation study and the time required for annotation. Since we want to retrieve multiple news articles for each podcast segment to increase the likelihood of including relevant and non-relevant articles, we decide to use only 40 podcast segments in the study.

6.3 Retrieval of News Articles

Given the 40 sampled podcast segments, the goal is to retrieve potentially relevant news articles for each of these segments. Before describing the model used for retrieval, the following section describes the applied data cleaning methods and their results.

6.3.1 Data Cleaning

For the news article dataset, we apply cleaning methods similar to those applied to the podcast segment dataset. For this dataset, we apply the two-sigma rule to the number of tokens in the merged article paragraphs. This results in 5841 records that are excluded from further processing. Using the results of the language classification in Chapter 5.3, 14 more records are excluded. After applying the above cleaning steps, the news article dataset consists of 113365 records that are available for retrieval. We can see that, relative to the size of the news article dataset, many fewer records are excluded than for the podcast segment dataset. This makes sense because text is the medium in which the articles were originally published and there was no transcription model or segmentation involved, as was the case for the podcasts.

6.3.2 Model

Retrieving potentially relevant news articles for each of the sampled podcast segments basically boils down to a document ranking task. Document ranking is essential in many information retrieval related tasks such as question answering or web search [ZML+21]. Document ranking is the task of ranking *documents* based on their *relevance* to a particular *query* [LCS97].

In terms of this study, the news articles represent the *documents* and a podcast segment represents the *query* to which the documents must be ranked. Traditional document ranking algorithms such as BM25 [RWJ+95] are based on the similarity between keywords in the query and documents. These algorithms have the disadvantage that they cannot perform an appropriate ranking if keywords in query and documents are not syntactically similar but only share the same meaning (i.e. query and documents use different terms to describe something). This problem is also known as the vocabulary mismatch problem [ZML+21]. Thus, we approach the retrieval of news articles using a *dense retrieval (DR)* approach. In DR, the query and documents are encoded in a vector space such that their embedding vectors carry their semantic meaning, and thereby DR methods overcome the vocabulary mismatch problem. Using these embeddings, the similarity between a query and documents can then be computed using similarity measures such as *cosine similarity* [LPS16]. The document embeddings that have the highest similarity to the query embedding are then used as retrieved documents. To speed up ad hoc retrieval of documents, their embeddings can be pre-computed and stored in an index, so that only query embeddings need to be computed for retrieval [ZML+21].

For ease of implementation, we use the reproducible information retrieval framework Pyserini [LML+21] for dense retrieval of potentially relevant news articles within the first annotation study. Out of the box, Pyserini includes pre-trained models that can be used for DR. Most of them are trained on the popular human-generated machine reading comprehension dataset (MS MARCO) [NRS+16]. Since most of the data in MS MARCO is in English and we need to process German data, models trained only on MS MARCO cannot be used for this task.

Pyserini does not offer models trained only in German, but only multilingual models. These multilingual models are evaluated using MIRACL [ZTO+23], a multilingual information retrieval dataset covering 18 languages, including German. According to Zhang et al. in [ZTO+23] and a ranking on Pyserini's website[2], the best performing purely DR-based model for German is *mDPR* [ZOML23]. The mDPR model uses separate encoders for query and documents and produces a similarity score for both by taking the dot product of the encoder outputs [ZOML23]. The respective encoders are based on the architecture and weights of multilingual BERT (mBERT) [DCLT18], a BERT model trained on corpora in 104 languages [PSG19]. The combined model is then fine-tuned using the MS MARCO dataset. Since *mDPR* is the only DR-based model supporting German available on Pyserini, we use this model for retrieval in the first annotation study. Since the length of queries and documents for mDPR is limited to 512 tokens, we truncate inputs longer than this before feeding them to the model. Using the mDPR model, we retrieve 20 news articles for each selected podcast segment, resulting in a total of 800 pairs to be annotated. We decide to retrieve 20 articles because this was the number of retrieved results in the annotation study performed in [CJJ+21].

6.4 Results

After all annotators completed the study, the results were exported from LimeSurvey using a csv file. In the following sections, we first analyze the background of the annotators and the agreement between them when annotating pairs in the study. Then, we examine how many of the pairs presented in the study were annotated as relevant and how much time it took the annotators to make the annotations. Finally, we describe the feedback on the study that we received from participants after the study was completed.

6.4.1 Annotator Background

Of the three annotators who participated in the first annotation study, one has a bachelor's degree in journalism and communications, one has a master's degree in business and fashion, and one is currently pursuing a master's degree in data science. All participants are native German speakers. Two of the three participants are familiar with the podcast and news offerings of *FALTER*. The participants consume podcasts one to four times a week. The consumption of news articles ranges from daily to twice a week. All participants report having experience using recommender systems outside of this study.

6.4.2 Inter-Annotator Agreement

Since all podcast segment news article pairs in the study are annotated by three annotators, it is possible to measure the agreement between them. The measurement of the agreement between annotators (also called inter-annotator agreement) shows the reliability of the annotation process, which translates into the reliability of the obtained annotations

[2] https://castorini.github.io/pyserini/2cr/miracl.html

[Art17]. A measure that was proposed as the standard reliability statistic for content analysis and similar data making efforts by A. Hayes and K. Krippendorff [HK07] in 2007 is *Krippendorff's* α. In contrast to other agreement measures, Krippendorff's α has the benefit of being able to work with any number of annotators, any type of labels, and small sample sizes [Kri11] and is therefore used within this work.

Krippendorff's α

The basic idea behind Krippendorff's α is to compare the observed agreement and the probability that the annotators agree purely by chance. For r annotators that annotated n podcast segment news article pairs (further denoted as units) using q categories (in our case $q = 2$ for *relevant* and *non-relevant* annotations), K. Gwet [Gwe11] proposed a multi-step process that can be followed to compute α described below. Note that the formulation below differs from Krippendorff's original formulation in [Kri11], but we choose to describe Gwet's formulation because it is more straightforward and has the same form as other inter-annotator measures such as Fleiss' Kappa [FQ15].

Given an annotation matrix $A \in \mathbb{R}^{n \times r}$ with $a_{i,j}$ representing one of the q categories, the first step is to construct an agreement matrix $R \in \mathbb{N}^{n \times q}$, where every value $r_{i,k}$ is the number of times the unit i was annotated as category k. Using this matrix, one can then calculate the probability that a randomly selected annotator will annotate any given unit as category k with

$$\pi_k = \frac{1}{n} \sum_{i=1}^{n} \frac{r_{i,k}}{\bar{r}}, \textit{where } \bar{r} = \frac{1}{n} \sum_{i=1}^{n} r_i. \tag{6.1}$$

The next step is then to calculate the weighted overall percent agreement of Krippendorff's α as follows:

$$p_a = \frac{1}{n\bar{r}} + (1 - \frac{1}{n\bar{r}}) \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{q} \frac{r_{ik}(\bar{r}_{ik+} - 1)}{\bar{r}(r_i - 1)}, \text{where } \bar{r}_{ik+} = \sum_{l=1}^{q} w_{kl} r_{il}. \tag{6.2}$$

In the above formula, the term $\bar{r}_{ik+}$ represents the number of annotators who annotated a unit i into a particular category, weighted by w_{kl} which represents the agreement between categories. In case of binary nominal data such as in this study w_{kl} are identity weights:

$$w_{kl} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{else.} \end{cases} \tag{6.3}$$

After performing all the above-mentioned calculations, the probability of agreement by chance can be computed as

Show	Krippendorff's α
FALTER Radio	0.145
Scheuba fragt nach	0.261
Besser lesen mit dem F.	-0.049
Klenk + Reiter	0.116
All shows	0.261

Table 6.1: Krippendorff's α values obtained in the first annotation study for all podcast shows

$$p_e = \sum_{k,l}^{n} w_{kl}\pi_k\pi_l. \tag{6.4}$$

With p_a and p_e at hand, Krippendorff's α can then be finally calculated with

$$\alpha = \frac{p_a - p_e}{1 - p_e}[\text{Gwe11}]. \tag{6.5}$$

The value of α can now range from minus one to one. $\alpha = 0$ corresponds to no agreement beyond chance and therefore lack of reliability. $\alpha = 1$ corresponds to perfect agreement between the annotators. Negative values of α indicate inverse agreement [ZCMK16]. Although Krippendorff's original publications [Kri11, HK07] do not define intervals to help interpret values between zero and one, publications such as [DBMB21, Hug21] agree that at minimum values of $\alpha = 0.6$ or $\alpha = 0.667$ are required to use the data to draw further conclusions.

Obtained Agreement

We compute the annotator agreement for each podcast show separately and for all shows combined, shown in Table 6.1. We can see that we obtain a value of $\alpha = 0.261$ for all podcast shows. This represents an agreement that is only slightly higher than the agreement by chance ($\alpha = 0$). We can also see that the agreement is even lower than $\alpha = 0.261$ for some shows, with "Besser lesen mit dem FALTER" even having a negative value. The difference in agreement between shows indicates that the relevance of podcast segments to news articles depends on the podcast format, and the difficulty of assessing relevance varies for each show.

In Table 6.2 we show the agreement obtained between each pair of annotators. We can see the highest agreement between the second and third annotators. The lowest agreement can be seen for the first and third annotators. The agreement between the second and third annotators is almost twice as high as the agreement between all three annotators. This indicates that the first annotator uses a different approach than the second and third annotators and highlights the need for annotation guidelines.

Annotator ID A	Annotator ID B	Krippendorff's α
1	2	0.236
1	3	0.105
2	3	0.478

Table 6.2: Krippendorff's α values obtained between annotators in the first annotation study

Show	# Relevant Pairs	% Relevant Pairs
FALTER Radio	73	36.5
Scheuba fragt nach	40	20.0
Besser lesen mit dem F.	0	0.0
Klenk + Reiter	12	6.0
All shows	125	15.625

Table 6.3: Number of relevant pairs after majority voting in the first annotation study

In general, the agreement reached in this study is too low to draw reliable conclusions using the annotations obtained. Therefore, we need to analyze the results of this study in depth and use the knowledge gained to conduct a second study with higher agreement.

6.4.3 Relevance of Pairs

To finally merge the annotations of each annotator into one label for each of the 800 podcast segment news article pairs, we use majority voting. The results are shown in Table 6.3. Note that the rightmost column shows the relative number of relevant pairs, with a total of 200 pairs for each show and 800 pairs for all shows. We can see that for pairs of all shows, only 125 or 15.625% are relevant after majority voting. The table shows that the most pairs have been annotated as relevant for the shows "FALTER Radio" and "Scheuba fragt nach" with 73 and 40 pairs respectively. It is also interesting to see that after majority voting, there is no relevant pair for "Besser lesen mit dem FALTER". Looking back at Table 6.1 in the previous section, this was also the show having the lowest agreement between annotators. This finding indicates that it is harder to retrieve and assess relevant pairs for this show than for other podcast shows.

6.4.4 Timing

One of the features of the online survey tool LimeSurvey is that it automatically tracks the time it takes users to answer questions. In the case of this annotation study, a question is one of the 40 podcast segments with its 20 retrieved news articles, which are assessed regarding their relevance. Figure 6.2 shows the time that the annotators spent on each question.

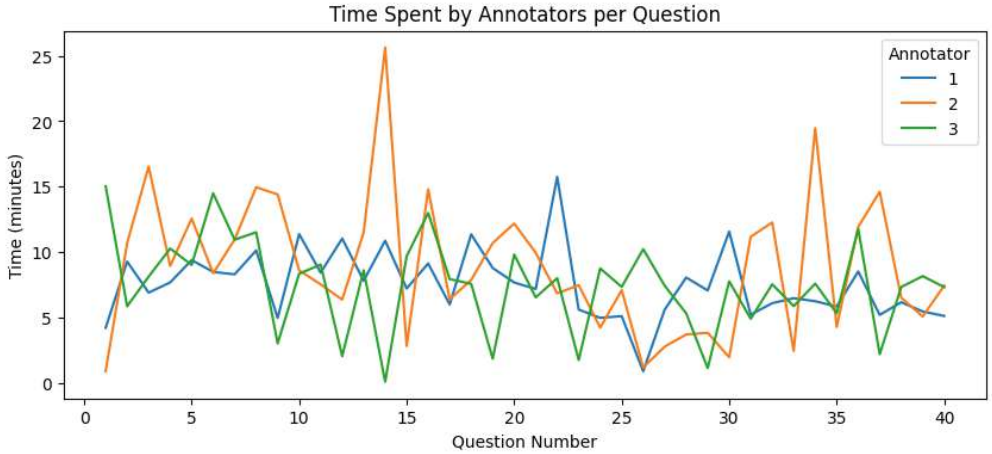

Figure 6.2: Time spent by each annotator per question for the first annotation study.

Note that the high and low peaks in the time spent by the annotators, such as for the second and third annotator for question number 14, can be explained by the way LimeSurvey handles time tracking. If someone leaves the LimeSurvey website open but performs other tasks until they come back to annotate a particular question, the time still runs, as can be seen for annotator two (question 14). If someone annotates all pairs in a question and then closes the site without submitting, the time tracking will start again when the annotator reopens the site. This can be seen for annotator three (question 14). On average, including these peaks, an annotator took 7.974 minutes to answer one question. To complete all annotation tasks in the study, each annotator took 318.944 minutes or 5.316 hours.

6.4.5 Annotator Feedback

After the study is complete, we ask the annotators for feedback on the study. They all report that the annotation process was tedious due to the amount of textual content that had to be captured to annotate the relevance between a podcast segment and a news article. As a result, it was only possible for the annotators to annotate a few questions in a session after they needed to take a break. One of the study participants reported that including the audio file of the podcast segment would help them better understand the segment than only reading its transcription text.

Regarding the selected podcast segments, the study participants report that it was often not possible for them to extract the context of the segment. This occurred because the segments often included generic intro or outro sections or because the hosts engaged in casual conversation without a clear topic. For news articles, the annotators report that it was easy to extract the topics and context. But they also say that in many cases it was incomprehensible why a particular news article was retrieved for a particular podcast segment as there was a lack of topical overlap. Furthermore, study participants say that,

especially in boundary cases, they had problems assessing whether or not a particular pairing is relevant. This also explains the low agreement between annotators and again highlights the need for guidelines.

6.5 Analysis of Causes

The feedback from the annotators described in the section 6.4.5 already gives us many points to improve in order to create a second annotation study. Since one of the biggest problems is the selection of appropriate podcast segments, we examine the used segments in terms of their inter-annotator agreement. We do this using the retrieval scores obtained by applying the mDPR model and the topic probabilities obtained by applying BERTopic to the segments, as described in Chapter 4.5.3. We hypothesize that podcast segments with little contextual information lead to low agreement. We expect that for these segments the mDPR model will produce low retrieval scores and that BERTopic will show no clear activation for a particular topic.

We investigate this by first computing the inter-annotator agreement among the 20 retrieved news articles for each of the 40 podcast segments separately. Since for three podcast segments no relevant article is annotated (i.e., there are only non-relevant annotations), and Krippendorff's α is not defined for the one-category case, this results in 37 values for α. To examine the retrieval scores, we select the retrieval score of the news article that produced the highest score for each podcast segment. Finally, we combine these maximum retrieval scores with the α values obtained for each segment. A graph showing the maximum mDPR retrieval scores versus Krippendorff's α for each podcast segment is shown in Figure 6.3.

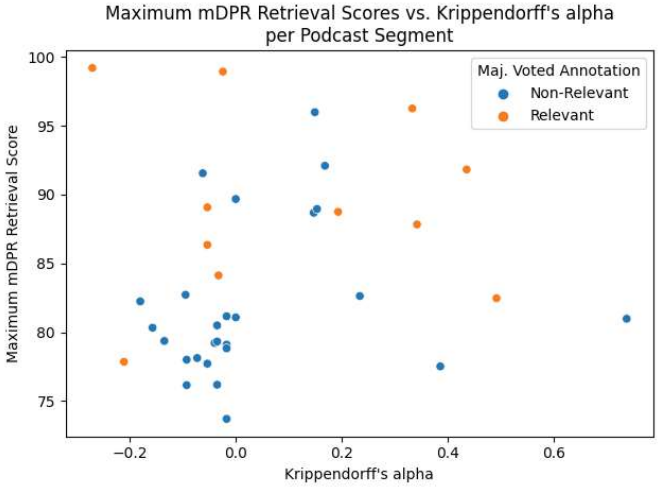

Figure 6.3: Maximum mDPR retrieval scores vs. Krippendorff's α for each podcast segment.

Contrary to our expectation, we do not see that segments with low agreement also produce

lower mDPR retrieval scores. We confirm what we observe visually by calculating the Pearson correlation [BCHC09] of both variables, which results in a value of 0.169.

As mentioned above, we expect that podcast segments with no clear topic activation will contain little contextual information and therefore lead to low agreement between annotators. To investigate this, we use the probabilities obtained by BERTopic fitted to the podcast segment dataset. We use BERTopic and none of the other models (LDA and NMF) because BERTopic showed the most promising results during data exploration. Applying the BERTopic model results in an identified topic along with probabilities for each topic. We select the maximum probability score for each podcast segment in the study and compared it with the obtained α values, shown in Figure 6.4. Note that the probabilities returned by BERTopic are separate probabilities for each topic and therefore do not sum up to one.

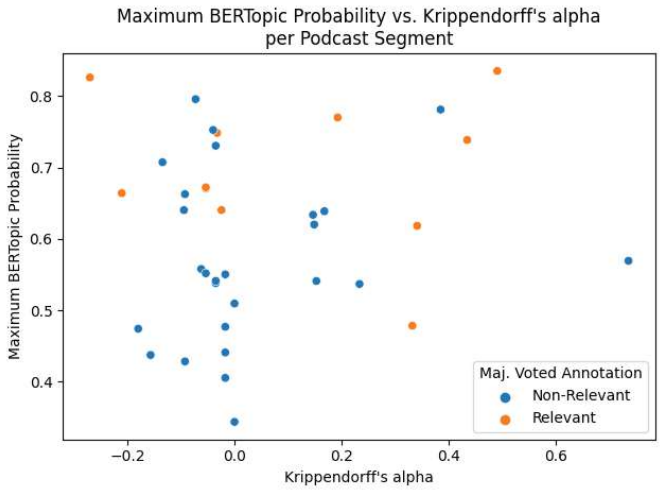

Figure 6.4: Maximum BERTopic probabilities vs. Krippendorff's α for each podcast segment.

Unfortunately, this expectation cannot be confirmed visually as well. Calculating the Pearson correlation between the maximum topic probability of BERTopic and Krippendorff's α values yields a value of 0.142. This again confirms that there is no strong correlation between annotator agreement and topic activation. Thus, we can use neither the retrieval scores nor the topic activations to detect podcast segments that yield low annotator agreement, and therefore have to come up with other approaches for a following annotation study.

CHAPTER 7

Named Entity Recognition (NER)

Two major problems identified in the first annotation study are the lack of guidelines for boundary cases and podcast segments with little contextual information. Since both problems cannot be solved directly by using textual information, we approach them by extracting named entities from podcast segment transcriptions and news article texts. The methods used for and the results obtained extracting named entities are described in the following sections.

7.1 Approaches

Named Entity Recognition (NER) is a widely used term in natural language processing, first mentioned at the Sixth Conference on Message Understanding in 1996 [NS07]. The term NER describes the task of recognizing instances of fixed entities from text, belonging to pre-defined semantic types such as *person*, *location*, *organization*, or numeric types such as *time* and *date* [LSHL20, NS07]. While early systems use rule-based algorithms, more recent methods focus on using machine learning to tackle NER [NS07]. Since the number of different entity types that can be extracted is limited by the datasets used for training the machine learning methods, we will use both methods in this work, described in the following sections.

7.1.1 Machine Learning-Based Entity Recognition

As mentioned above, machine learning-based NER methods are limited by the datasets on which they are trained. The datasets available for NER range from multilingual texts with general entity types to monolingual texts with domain-specific entities such as disease types [JRA23]. However, the only dataset that includes German is the CoNLL-2003 dataset [SDM03], which was provided as part of the SIGNLL Conference on

Computational Natural Language Learning in 2003[1]. The dataset contains 2302 articles in German and English with annotations for the entity types *person (PER)*, *location (LOC)*, *organization (ORG)* and *miscellaneous (MISC)*. Here, entities of type MISC contain instances such as events, brands, or products that cannot be assigned to other types.

The model that achieves state-of-the-art results for German NER on CoNNL-2003 is described by Schweter and Akbik in [SA20]. The authors employ a multilingual XLM-RoBERTa (XLM-R) Transformer model [CKG+19] that was pre-trained on data from a cleaned Common Crawl corpus [WLC+20] for 100 different languages. Schweter and Akbik add a final linear layer to XLM-R and then fine-tune the model using CoNNL-2003. Unlike other Transformer-based NER approaches, the authors include the surrounding context of a sentence to compute the word-level embeddings. They refer to this method as document-level features [SA20] and show that it results in improved performance. Schweter and Akbik publish a German trained model on the open source machine learning platform Huggingface[2], which we use in this work. Using their own natural language processing framework "FLAIR" [ABB+19], this model can be easily applied to the podcast segment and news article datasets.

7.1.2 Rule-Based Entity Recognition

We expect that time-related information can also be useful to assess the relevance between a podcast segment and a news article. Since the CoNNL-2003 dataset does not contain an entity type for time or date and therefore no German machine learning-based NER model exists, we also apply a rule-based recognition method.

Rule-based entity recognition can be achieved by hand-crafting rules using methods such as regular expressions [NS07]. This process of rule-crafting can be tedious, especially for relatively unstructured entities such as time or date. Instead of creating our own rules, we use the rule-based entity recognition framework Duckling [Fac]. Duckling was developed by Facebook and currently supports 49 languages, including German. The framework uses regular expressions to recognize and parse 13 dimensions from text to structured data. Among others, *time* is one of the dimensions supported by default. We apply Duckling to both datasets, extracting only entities related to time. The rule-based approach has the disadvantage that the results may include instances that are not related to the desired entity type. Since this is also the case for the podcast segment and news article datasets, we finally clean up the results by excluding all entities that do not contain numeric information. We further denote this cleaned entity as *DATE*.

[1] https://www.clips.uantwerpen.be/conll2003/
[2] https://huggingface.co/flair/ner-german-large

7.2 Results

In the following sections, we describe the results of named entity recognition using the podcast segment and news article datasets. For each dataset, we report the distributions of recognized entities using machine learning-based and rule-based methods. In addition, we report the number of distinctly recognized entity types for both datasets.

7.2.1 Podcast Dataset

After applying machine learning-based NER to the podcast dataset, we visualize the distribution of the number of recognized entities for each of the four types of entities, shown in Figure 7.1. Looking at the calculated mean values displayed in the titles of each histogram, we can see that *LOC*, with a value of 2.837, is the entity most frequently recognized in the segment transcriptions. It is immediately followed by the *PER* entity with a mean of 2.826. This means that on average, each document contains more than two instances of the entity types mentioned above. The entities *ORG* and *MISC* are less present, with mean values of 1.617 and 0.517 respectively. We can further see that all distributions are right-skewed, with only a few records having many entities.

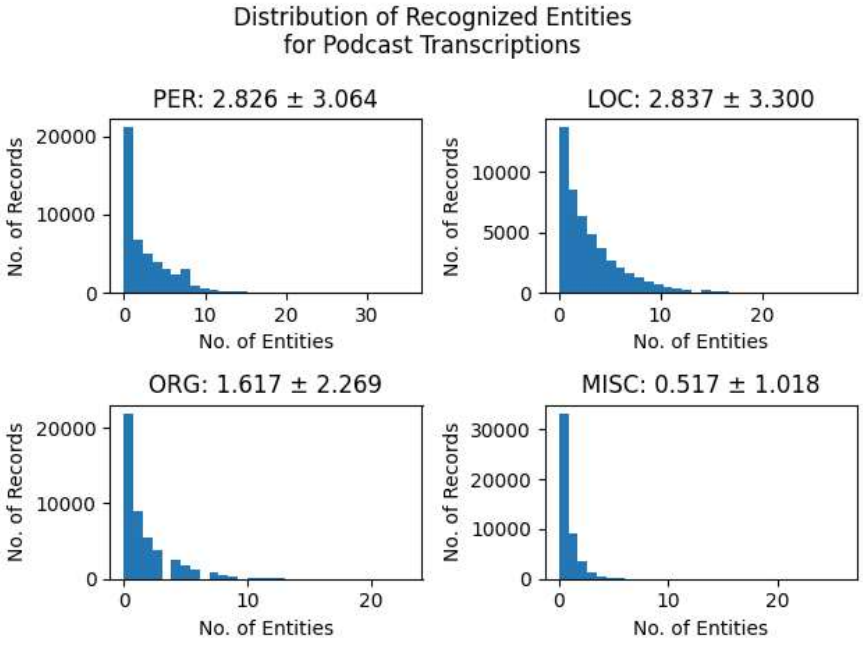

Figure 7.1: Distribution of number of recognized entities per document using machine learning-based NER on the podcast dataset.

Looking at the distribution of the *DATE* entity obtained by applying the rule-based NER, shown in Figure 7.2, we can see that this entity is also not very prominent in the segment dataset. With a mean of 0.544, a *DATE* entity is present in only about

every second segment transcription. This low frequency could be due to the cleaning performed. However, in this work, we prefer the quality of the instances to the frequency of occurrence.

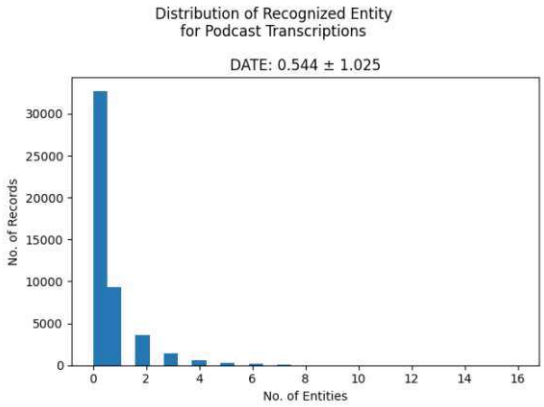

Figure 7.2: Distribution of number of recognized time related entities per document using rule-based NER on the podcast dataset.

Finally, in Figure 7.3 the distribution of the distinctly recognized entity types for NER based on machine learning and rules is shown. The graph shows the number of records for which at least one instance of a particular entity type was recognized. On average, each podcast segment transcription contains 2.635 different recognized entity types. We employ this graph because we expect the number of entity types to be an indicator of the contextual information contained in the data, which can then be used for data cleaning and annotation guidelines.

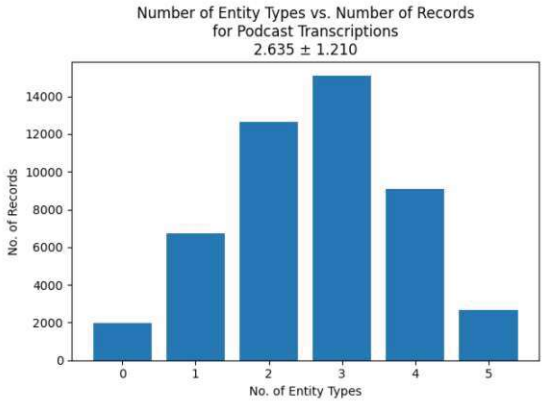

Figure 7.3: Distribution of number of distinct entity types per document using machine learning-based and rule-based NER on the podcast dataset.

7.2.2 News Article Dataset

Also for the news article dataset, we visualize the distribution of recognized entities using machine learning-based NER in Figure 7.4. We can observe that the entity type that is clearly most prominent is PER with a mean of 10.362 instances per record. The second most present entity type is LOC with a mean of 6.036. Directly followed by ORG with 4.376 instances and MISC with 3.116 on average.

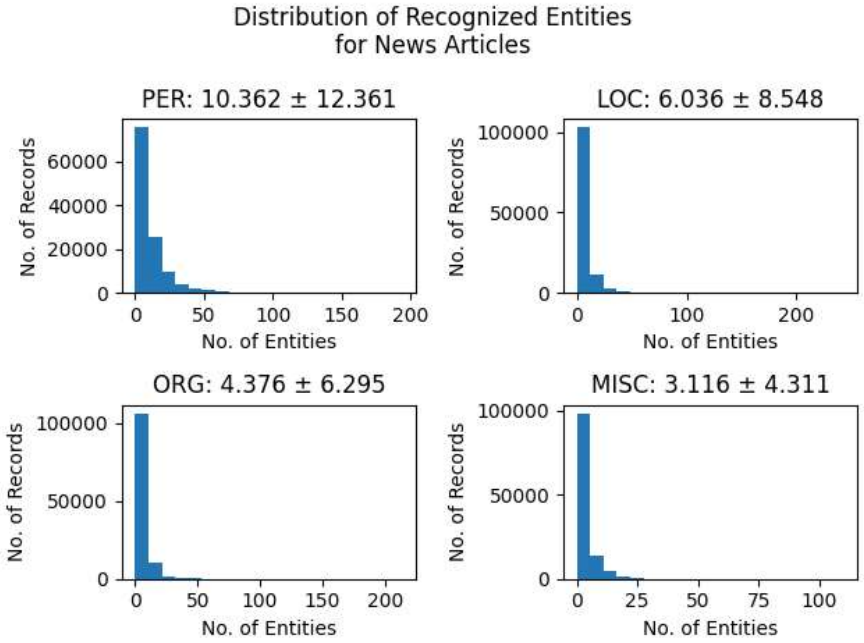

Figure 7.4: Distribution of number of recognized entities per document using machine learning-based NER on the news dataset.

As for the podcast dataset, we also visualize the distribution of recognized entities using rule-based NER in Figure 7.5. We can see that, on average, each news article contains 3.126 instances of the entity *DATE*.

In general, we can see that more entities were recognized for the news article dataset than for the podcast dataset. Reasons for this are probably the longer length of the documents, which provides more space for entities, but also the contextual density of the articles. Not only is the average number of recognized entities for all entity types higher, but we can also observe higher standard deviations for the news article dataset. This indicates that there are articles with many fewer or more instances.

Again, we visualize the distribution of the number of distinctly recognized entity types, shown in Figure 7.6. We see that the distribution is left-skewed and that for most of the news articles an entity instance was recognized for *PER*, *LOC*, *ORG*, *MISC*, and *DATE*. The calculated mean of 4.006 again shows that the news articles contain not only more

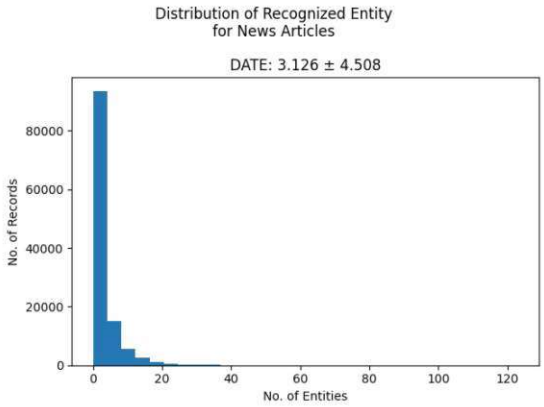

Figure 7.5: Distribution of number of recognized time related entities per document using rule-based NER on the news dataset.

instances for certain entity types, but also more different entity types than the podcast segments.

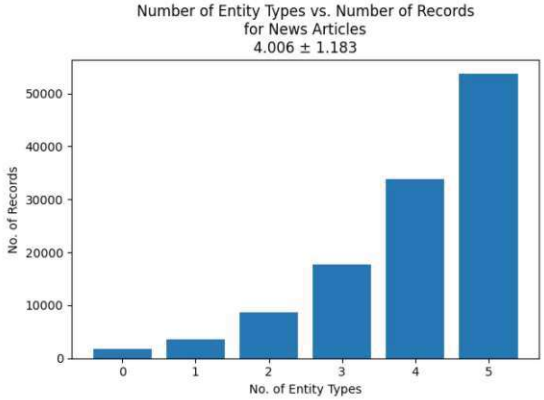

Figure 7.6: Distribution of number of distinct entity types per document using machine learning-based and rule-based NER on the news dataset.

CHAPTER 8

Second Annotation Study

As with the first annotation study, this chapter describes the design, implementation, and results of the second, improved annotation study. We begin by describing the general approach and improvements derived from the first study, followed by the selection of podcast segments. We then describe how we retrieve news articles for each of these segments, and end with an analysis of the results obtained in terms of inter-annotator agreement, relevance, timing, and annotator feedback.

8.1 Approach

The goal of this study is to use the knowledge gained from the first annotation study to create a second study that results in higher inter-annotator agreement. Again, we start by selecting appropriate podcast segments to use in the study. Here, we apply advanced data cleaning methods before sampling the segments, with the goal of excluding segments that contain a low amount of contextual information. The process of selecting podcast segments is described in further detail in Section 8.2.

Afterwards, we again retrieve potentially relevant news articles for each podcast segment. For the second annotation study, we experiment with three new different retrieval models and finally use the one that retrieves the most promising articles. The complete retrieval of news articles is described in more detail in Section 8.3.

Given the new pairs of podcast segments and news articles, we create another annotation study using LimeSurvey. After feedback from the first study, we include the audio files of the podcast segment to enhance the comprehension time of the content. Moreover, to address the feedback on the tedious annotation process due to the amount of content, we introduce tables of recognized entities and highlighted them in both the podcast segment and the news article texts. An exemplary question that includes the aforementioned improvements is shown in Figure 8.1.

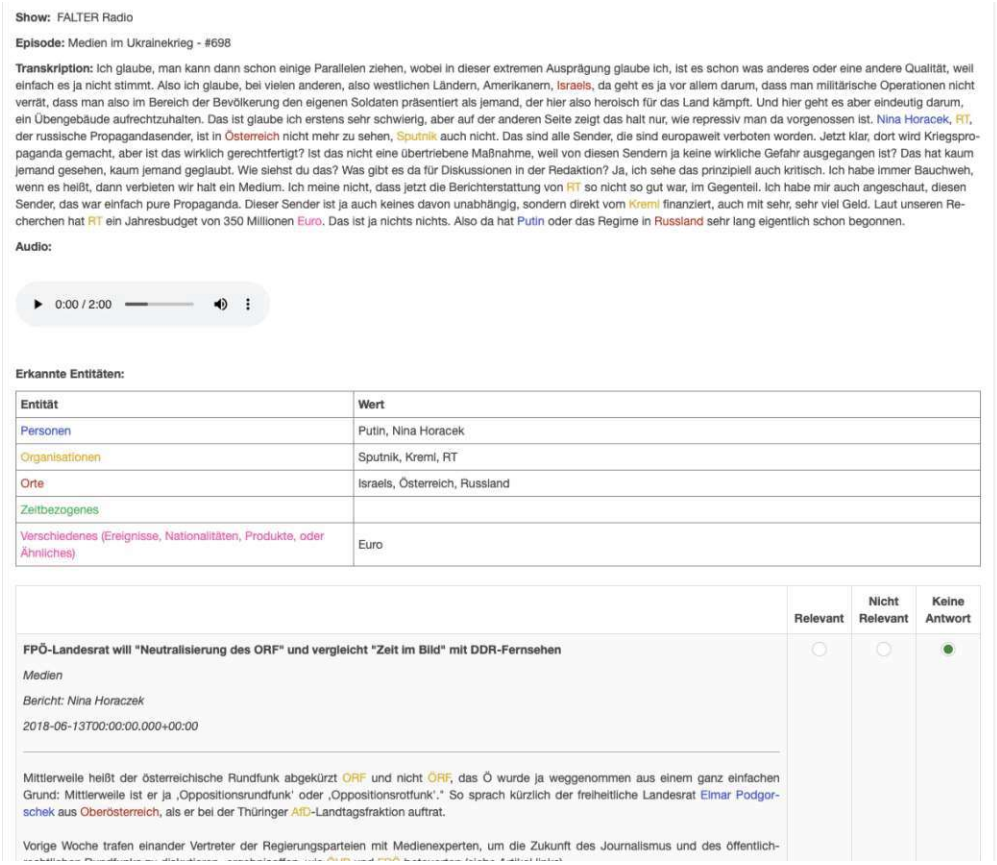

Figure 8.1: Exemplary question in the second improved annotation study between podcast segments and news articles.

In addition to the above improvements, we define an annotation guideline for boundary cases. When annotators are in no doubt, we instruct them to annotate a pair as relevant as they did in the first study. Only if the annotators are not sure how to annotate a pair, we tell them to annotate the pair as relevant if there are *semantically similar* instances between the podcast segment and the news article for at least two different entity types, and otherwise as non-relevant. *Semantically similar* means that synonyms and instances with the same meaning are acceptable. For example, consider the recognized entities for the podcast segment in Figure 8.1. In a boundary case, a news article that mentions "Russia's President" as an instance for the PER entity ("Putin" is an instance of the podcast segment) and "Russia Today" as an instance for the ORG entity ("RT" is an instance of the podcast segment) should be annotated as relevant. Since we found that few pairs were annotated as relevant in the first study and we want to lower the barrier for a pair to be annotated as relevant, we decide to use a threshold of only two different entity types in the guideline.

8.2 Selection of Podcast Segments

As in the first annotation study, we first perform data cleaning on the podcast segment dataset. In the second study, in addition to the cleaning applied in the first study, we apply more sophisticated methods based on NER, the position of the segment in the audio file and audio features. We then again sample podcast segments from the cleaned dataset. Both steps are described in the following sections.

8.2.1 Data Cleaning

After applying the cleaning methods based on the number of tokens in the podcast segments and their classified language used in the first annotation study, we end up with 43464 remaining segments. We then remove all segments for which less than three entity types were recognized. This rule affects 21309 segments, of which 19276 segments have not yet been excluded by the previous cleaning methods. We decide to use a threshold of three as we expect this to dramatically increase the contextual quality of the segments. Using three also increases the probability that podcast segments will be useful for annotation, especially according to the defined boundary guideline.

During the expert interview (see Chapter 3.2), the podcast creator mentioned that most podcasts contain an intro and an outro. This was also confirmed by the results of the first annotation study. Thus, the next step is to clean the podcast segments based on their position in the audio file. We exclude all segments that come from the first or last five minutes of a podcast episode. Although the podcast creator reported that the length of intro and outro sections is only up to three minutes, we set this threshold at five minutes to ensure that all problematic segments are excluded. Applying this rule affects 5813 segments, 3920 of which have not yet been excluded by other cleaning rules.

Finally, we perform data cleaning based on the audio features that we extract from the audio of the podcast segments. We consider audio features for data cleaning because we observed that certain podcast segments in the dataset contain music or silence. These segments have a high probability of generating incorrect transcriptions and can't be easily identified by purely text-based approaches, because transcription errors can also arise from non-verbal audio. To perform audio-based cleaning, we use the YAMNet [Pla20] audio classification model. YAMNet originates from Google Research and is based on the convolutional neural network architecture MobileNet [HZC+17]. The model is trained on the AudioSet [GEF+17] dataset, which consists of human-annotated 10-second audio segments. For each raw audio file, YAMNet converts the waveform into mel spectrograms, which are used as input for the convolutional neural network. These spectrograms represent the distribution of frequencies in the audio file over time and provide a visual representation of the audio that the neural network can process. For each 960 milliseconds of audio in the file, YAMNet predicts the probability of 521 classes (such as speech, music, or silence) and generates an embedding vector that captures the key features of the audio content.

We use YAMNet because it was also employed by Abigail et al. in [AMT+21] when they enriched the dataset by Clifton et al. [CRY+20] with audio features. We apply the model to all podcast segments. To obtain a single probability distribution for each 120-second podcast segment, we aggregate the predicted probabilities across the 960-millisecond windows, using both the mean and maximum for each class. The classes we focus on to create a data cleaning rule are *Silence* and *Music*. These classes are relevant, since segments with a high probability of containing silence or background music are more likely to have corrupted transcriptions. For example, a podcast segment with silence or background music may leads to incomplete or incorrect text output during transcription. For the aggregated mean and maximum probability distributions, we visualize how many records are affected by a data cleaning rule for both classes when we choose a certain probability threshold, as shown in Figure 8.2.

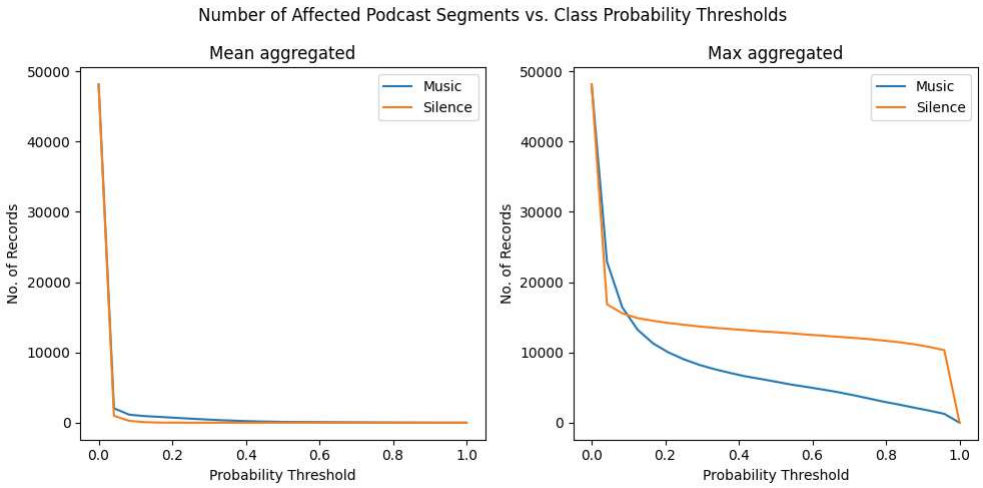

Figure 8.2: Number of affected records vs. probability thresholds for mean and maximum aggregated YAMNet Silence and Music class distributions.

We can see that for the maximum aggregated probability distributions, more records are affected over the full range of probability thresholds. We decide to exclude all podcast segments where the maximum aggregated probability is greater than 0.4 for the *Music* or *Silence* classes. We use this threshold because we expect it to be a good compromise between the number of excluded records and the inclusion of false negatives. The application of this final cleaning rule affects 13235 records for *Silence* and 6807 records for *Music*, of which 4155 and 1224 records, respectively, were not excluded by other cleaning rules.

Overall, in the data cleaning process in the second annotation study, out of a total of 48138 podcast segment records, 2023 records are removed due to their length (i.e. their number of tokens), 2651 records are removed due to their classified language, 19276 records are removed due to less than three different distinctly present named entity types, 3920 records are removed due to their position in the audio file, 4155 records are removed

due to an activation higher than 0.4 of the YAMNet class *Silence*, and 1224 records are removed due to the YAMNet class *Music*. After applying all of the aforementioned cleaning steps, there are 12142 podcast segments that can be used for sampling.

8.2.2 Sampling

Of the 40 podcast segments sampled in the first annotation study, 30 segments were excluded by applying the more sophisticated data cleaning rules described in the previous section. In Table 8.1, we show for each podcast show how many of the segments sampled in the first study were not excluded.

Show	# Segments Not Excluded
FALTER Radio	3
Scheuba fragt nach	3
Besser lesen mit dem F.	1
Klenk + Reiter	3

Table 8.1: Number of podcast segments sampled in the first annotation study that are not excluded by the data cleaning described in Chapter 8.2.1.

It can be seen that most of the segments were excluded for "Besser lesen mit dem FALTER". This show had the lowest inter-annotator agreement and no relevant pairs in the first study. We believe that the low level of contextual information in the podcast segments contributed to this. This suggests that the improved data cleaning successfully removed some problematic segments.

To get back to 10 segments for each podcast show, we again randomly sample nine new podcast segments for "Besser lesen mit dem FALTER" and seven segments for the other shows using a random seed of 42.

8.3 Retrieval of News Articles

Given the 10 podcast segments from the first annotation study and the 30 newly sampled podcast segments, we again perform data cleaning and then experiment with three different models to retrieve potentially relevant news articles. These models include the term frequency-based model BM25 [RWJ+95] and two Transformer-based methods, STS-RoBERTa [May20] and JinaAI sentence embeddings [MKS+24]. All of these steps are described in the following sections.

8.3.1 Data Cleaning

Again, we first perform data cleaning based on the number of tokens in the merged news article paragraphs and the classified language, as in the first annotation study. After applying these cleaning rules, 113365 news article records remain to be processed. As

with the podcast segments, we apply a new cleaning rule based on the number of different entity types recognized for the news articles. We exclude all articles in which less than three different types were detected. This affects 14079 records, of which 13068 were not excluded by the other two rules. After applying all of the above cleaning rules, the news article dataset used in the second annotation study consists of 100297 records.

8.3.2 BM25

As briefly described in Chapter 6.3.2, BM25 [RW94] is perhaps the best known document retrieval model based on term frequencies of queries and documents [KDVBL20]. There are many different versions of this model in the literature, all of which are referred to as BM25. In this work, we use the Pyserini [LML+21] implementation of BM25, which wraps the Java implementation of the Apache Lucene open source search library [BMII12]. For a query q and a document d, the BM25 model used in Lucene is defined as follows:

$$bm25(q,d) = \sum_{t \in q} \log(1 + \frac{N - df_t + 0.5}{df_t + 0.5}) * \frac{tf_{td}}{k_1 * (1 - b + b * (\frac{L_{dlossy}}{L_{avg}})) + tf_{td}} \tag{8.1}$$

where N is the number of documents in the collection, df_t is the number of documents that contain a term t, and tf_{td} is the number of times a document d contains a term t. L_{avg} is the average number of terms in all documents in the collection. L_{dlossy} is the length of document d compressed to one byte (i.e. one out of 256 values). k_1 and b are tunable hyperparameters, which are set to $k_1 = 0.9$ and $b = 0.4$ by default in Lucene [RWJ+95, KDVBL20]. As in the first study, we use this model to retrieve 20 news articles for each sampled podcast segment and create an annotation study in LimeSurvey, which we evaluate later on.

8.3.3 STS-RoBERTa

Since we expect Transformer-based retrieval models to outperform simple term frequency-based models, we also experiment with such methods in the second study. STS-RoBERTa [May20] is a model based on XLM-RoBERTa [CKG+19]. It has been pre-trained on a large dataset [RG20] containing more than 50 languages. STS-RoBERTa is fine-tuned using the English and German subsets of the multilingual Semantic Text Similarity (STS) benchmark dataset [May21]. This dataset is a collection of datasets containing text from image captions, news headlines, and user forums. The model is designed to generate semantically meaningful embeddings for German and English texts, which can then be compared using cosine similarity [May20]. It is published on the machine learning platform Hugginface[1] and is also included in the Python sentence embedding framework SBERT [RG19]. Due to the availability of the model in SBERT, it can be used in Pyserini [LML+21] without any modifications. Using this model, we also retrieve 20 potentially relevant news articles for each podcast segment and create an annotation

[1] https://huggingface.co/T-Systems-onsite/cross-en-de-roberta-sentence-transformer

study in LimeSurvey. Since the length of queries and documents is limited to 512 tokens, we truncate inputs longer than this.

8.3.4 Jina Sentence Embeddings

The last method we use to retrieve potentially relevant news articles in the second annotation study is based on Jina sentence embeddings [MKS+24]. Other Transformer models, such as mDPR (used in the first study) and STS-RoBERTa, are not able to handle long documents, so their input often has to be truncated to 512 tokens. This limitation is overcome by the Jina sentence embeddings proposed by Günther et al. [MKS+24]. The authors propose a modified BERT model architecture, which they denote as JinaBERT. This architecture is able to generate sentence embeddings for long documents, using an efficient method to encode the position of tokens in input and output sequences.

In a regular BERT model, the absolute position of a token in the input sequence is injected by adding sine and cosine function outputs of different frequencies to the word embeddings. This step is called *positional encoding*. When training a BERT model, these positional encodings are generated up to a fixed length of input tokens (e.g. 512) and the model learns to interpret the position of a token relative to this maximum length [DCLT18]. Although the sine and cosine functions can theoretically produce positional encodings for longer sequences after the model has been trained, the model's attention mechanism is unable to properly interpret these encodings not seen during training, and so this approach does not extrapolate well [PSL21]. Simply using larger maximum sequence lengths in training leads to higher memory and computational requirements for model training and inference, and is therefore often not feasible.

The JinaBERT model overcomes this issue by replacing the positional encodings in BERT with an approach called *Attention with Linear Biases (ALiBi)* [PSL21]. Using ALiBi, no positional encodings are added to the word embeddings, but instead the attention scores are directly biased in proportion to the relative distance between tokens in the input and output sequences [PSL21]. This means that tokens that are further apart in a sequence have a greater bias added to their attention score, making it harder for these tokens to attend to each other. In contrast to positional encodings, this method is able to generalize well to long input sequences without additional learnable parameters or computational overhead [PSL21]. The use of ALiBi instead of positional encodings allows the JinaBERT model to work with input up to 8912 tokens long.

The authors train the JinaBERT model from scratch using a cleaned version of the CommonCrawl dataset [WLC+20]. The model is then fine-tuned to produce semantically meaningful embeddings in two steps. First, 385 million text pairs from 40 datasets are used to teach the model how to generate similar embeddings for semantically similar texts. Then, another collection of datasets, including datasets such as MS MARCO [NRS+16] and Natural Questions [KPR+19], is used to teach the model to better distinguish between relevant and related but irrelevant text passages. A detailed description of all data used for pre-training and fine-tuning the model can be found in a previous publication of

the authors [GMG+23]. Günther et al. publish six different models, including a large model for generating embeddings for German and English texts on the machine learning platform Huggingface[2]. This model is used in this work.

Given the fine-tuned Jina sentence embedding model, the authors perform an evaluation using the *Massive Text Embedding Benchmark (MTEB)* [MTMR22]. MTEB is a collection of 58 datasets covering eight tasks, including STS, clustering, retrieval, and others. Jina's sentence embeddings show state-of-the-art performance on the tasks in this collection. In addition, Günther et al. publish a table comparing their performance with other models on a German and English subset of MTEB, shown in Figure 8.3. We can see that the

Model Name	jinaai/jina-embeddings-v2-base-de	multilingual-e5-base	multilingual-e5-large	T-Systems-onsite/cross-en-de-roberta-sentence-transformer	distiluse-base-multilingual-cased-v2
STS22 (STS)	**59.07%**	55.95%	56.59%	38.17%	35.73%
GermanSTSBenchmark (STS)	**88.32%**	78.27%	82.06%	84.72%	75.23%
MIRACL (Reranking)	64.30%	67.34%	**70.36%**	55.02%	57.20%
XMarket (Retrieval)	**18.66%**	12.72%	14.12%	5.48%	2.09%
GermanDPR (Retrieval)	79.36%	78.83%	**82.09%**	42.68%	62.08%
GerDaLIR (Retrieval)	**20.03%**	7.56%	7.63%	0.49%	1.37%
TenKGnadClusteringP2P (Cluster)	42.84%	41.92%	**44.83%**	23.56%	35.49%
TenKGnadClusteringS2S (Cluster)	23.69%	**33.01%**	30.84%	9.65%	18.93%
STS22 (de-en)	55.97%	54.93%	**56.60%**	39.65%	47.51%
Average - STS	**67.79%**	63.05%	65.08%	54.18%	52.82%
Average - Reranking	64.30%	67.34%	**70.36%**	55.02%	57.20%
Average - Retrieval	**39.35%**	33.04%	34.61%	16.22%	21.85%
Average - Cluster	48.99%	48.88%	**50.39%**	33.04%	38.97%
Average - All	**55.11%**	53.08%	**55.11%**	39.62%	42.71%

Figure 8.3: Comparison of Jina Sentence Embedding model performance on a German and English subset of MTEB [Gü24].

Jina embedding model outperforms three of the four other models and shows similar performance to one model averaged over all tasks. We can also see that the Günther et al. model outperforms all other models, and especially the *T-Systems-onsite/cross-en-de-roberta-sentence-transformer* model described in 8.3.3, when performing retrieval tasks. As for the other two models described in the previous sections, we retrieve potentially relevant news articles using Jina embeddings and create an annotation study on LimeSurvey.

The Pyerini framework, which we used to retrieve news articles with the other models, does not support the use of Jina embeddings. Therefore, for this approach, we manually create an index containing the embedding vectors of all available news articles using the *Facebook AI Similarity Search (FAISS)*[3] library. For each podcast sample used in the study, we query this index using the vectors of the transcription texts created using Jina embeddings, and select the news articles with the highest cosine similarity [LPS16] to the given sample vector as the retrieved articles.

[2]https://huggingface.co/jinaai/jina-embeddings-v2-base-de
[3]https://github.com/facebookresearch/faiss

8.3.5 Selected Retrieval Model

In the previous sections, we described how we created potential annotation studies in LimeSurvey using three different models to retrieve news articles. By comparing the resulting studies, we find that all models show better results, as was the case when mDPR was used in the first study. The retrieved news articles look most promising for the Jina embeddings, which could be explained by the larger context used by the model when creating embeddings. Therefore, we decide to use Jina embeddings for the second annotation study. This decision is also supported by the results in Figure 8.3.

8.4 Results

In the following sections, we describe the results of the second annotation study. As for the first study, we first analyze the background of the annotators and the agreement between them when annotating the pairs in the study. Then, we examine how many of the pairs presented in the study were annotated as relevant and how much time it took the annotators to make the annotations. Finally, we describe the feedback on the study that we received from participants after the study was completed.

8.4.1 Annotator Background

Unfortunately, the annotator with a background in journalism and communications who participated in the first study was unable to participate in the second study. As a result, we gained a new study participant with a master's degree in business and fashion. The new participant is a native German speaker and is not familiar with the podcast and news offerings of *FALTER*. They consume podcasts twice a week and news articles up to five times a week and are experienced in using recommender systems. Except for the annotator with a background in journalism and communication, the group of annotators remains the same as in the first study.

8.4.2 Inter-Annotator Agreement

We again report the agreement among the three annotators using *Krippendorff's* α[HK07]. Table 8.2 shows the obtained α values for each podcast show and all combined podcast shows. We can see that we obtain a value of $\alpha = 0.777$ for all podcast shows. This is a drastically improved agreement between the annotators compared to the $\alpha = 0.261$ obtained during the first annotation study. We can see that the agreement between the different podcast shows still varies, but it is increased for all of them. "Besser lesen mit dem FALTER" still has the lowest agreement with $\alpha = 0.432$. In contrast to the first study, "Klenk + Reiter" has the highest agreement, followed by "FALTER Radio" and "Scheuba fragt nach".

In Table 8.3 we again show the obtained agreement between each pair of two annotators for all shows. We can see less differences in the agreement as we observed after the evaluation of the first annotation study. The highest agreement is obtained between the first and

Show	Krippendorff's α
FALTER Radio	0.799
Scheuba fragt nach	0.623
Besser lesen mit dem F.	0.432
Klenk + Reiter	0.819
All shows	0.777

Table 8.2: Krippendorff's α values obtained in the second annotation study for all podcast shows.

Annotator ID A	Annotator ID B	Krippendorff's α
1	2	0.801
1	3	0.735
2	3	0.795

Table 8.3: Krippendorff's α values obtained between annotators in the second annotation study.

Show	# Relevant Pairs	% Relevant Pairs
FALTER Radio	104	52.0
Scheuba fragt nach	71	35.5
Besser lesen mit dem F.	3	1.5
Klenk + Reiter	16	8.0
All shows	194	24.25

Table 8.4: Number of relevant pairs after majority voting in the second annotation study.

second annotators, followed by the second and third annotator. The lowest agreement is seen for the first and third annotator. In general, we can see that improvements made in the second study largely increase the obtained inter-annotator agreement. Except for "Besser lesen mit dem FALTER", we obtain α values greater than 0.6, which allows us to use the data to draw reliable conclusions in subsequent steps.

8.4.3 Relevance of Pairs

To obtain a joint annotation for each pair, we again perform majority voting, as in the first annotation study. The absolute and relative number of podcast segment and news article pairs annotated as relevant is shown in Table 8.4.

We can see that for pairs of all shows, 194 or 24.25% were annotated as relevant after majority voting. This is an increase of 69 pairs or 8.625% compared to the first annotation study and indicates that improvements in data cleaning and retrieval of news articles lead to more relevant pairings. Again, "FALTER Radio" is the podcast show with the most

pairs annotated as relevant, followed by “Scheuba fragt nach”. In contrast to the first study, where no pairs were annotated as relevant for “Besser lesen mit dem FALTER”, three relevant pairs were annotated for this show in the second study. In general, we expect that the differences in relevant pairs per show are explained by the structure of the podcasts and topics of the news articles, and not by the methods used to create the annotation studies.

8.4.4 Timing

For the second annotation study, we also examine the time it took an annotator to answer a question. Each question represents one of the 40 podcast segments with its 20 retrieved news articles. Figure 8.4 shows the time that the annotators spent on each question. As in the first study, we observe high and low peaks in the time spent, which can be

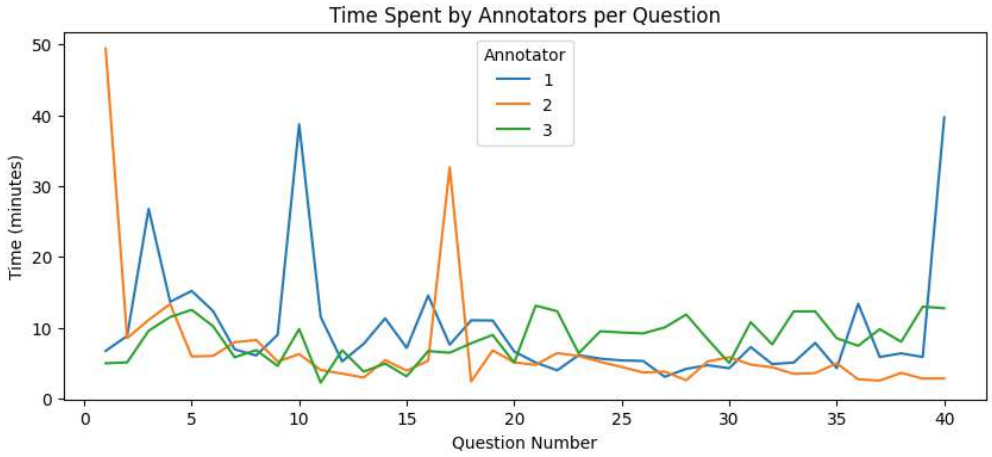

Figure 8.4: Time spent by each annotator per question for the second annotation study.

explained by the LimeSurvey time tracking system. Including this peak in an average calculation, each annotator took 8.343 minutes to answer one question, which is 0.369 minutes or 4.6% longer than in the first annotation study. The time it took an annotator to complete all tasks in the second annotation study is 333.721 minutes or 5.562 hours. Again, these results should be viewed with care, due to the outlier peaks included in the data.

8.4.5 Annotator Feedback

After completing the second annotation study, we again ask the participants for feedback. All participants report that the recognized entities displayed for the podcast segments and news articles helped them understand the data more quickly. One annotator reported that the included audio file helped them better understand the transcriptions of the podcast segment. Despite these helpful modifications, the participants reported that it was still tedious to annotate the pairs, and they were only able to answer a few questions after having to take a break. Regarding the pairs used in the study, the annotators reported that contextual quality improved dramatically compared to the first study. However, they also said that especially for the segments of "Besser lesen mit dem FALTER" and "Klenk + Reiter" it was still difficult to extract the contextual information of the two-minute podcast segment. Regarding the retrieved news articles, participants felt that their assignment was more reasonable, which is also reflected in the number of relevant pairs annotated in the second study. Finally, they say that the boundary case guideline is helpful in deciding which annotation to use, but a few times applying the guideline also led to the exclusion of relevant pairs due to missing entities.

CHAPTER 9

Modeling: A Learning-to-Re-Rank Approach

The final step in building a content-based cross-domain recommender system between podcasts and news articles described in this work is modeling. In this chapter, we describe the modeling approach and the model used, followed by the features used. We then describe the evaluation metrics employed, the train-test split, and the model training procedure, and finally report the results obtained for different podcast representations, models, and podcast shows.

9.1 Approach

One goal of this work is to evaluate the effectiveness of different podcast representations in terms of their recommendation performance. Besides textual attributes such as titles and episode descriptions, the main medium of podcast content is audio. In Chapter 4.4 we already described how audio can be transformed into a textual medium using automatic speech recognition. However, in this work, we also want to investigate whether features derived directly from the audio signal, without first transforming it into text, can also improve recommendation performance.

In the previous sections, we outlined several models that create meaningful embeddings of textual data, which can then be used to measure semantic similarity between podcast segments and news articles. Moreover, as part of the data cleaning performed in Chapter 8.2.1, we described the YAMNet audio classification model, which is also capable of creating embeddings that meaningfully represent audio. If we now want to recommend a news article based on the audio embeddings of a podcast segment, we quickly run into a problem. The text embeddings and the audio embeddings are generated by different models and may not even have the same dimensionality. Both the podcast segment and

the news article are represented by embeddings from disjoint vector spaces, so a direct similarity measure between them is not possible.

An approach to solving this problem is representation learning. Here, representation learning means using a model that is learned to directly project different media types that are semantically similar to a common representation (i.e., into a common embedding space). In [RKH+21], Radford et al. propose *Contrastive Language-Image Pre-training (CLIP)*, a method capable of projecting text and images into a common space. In publications such as [GRHD21, WSKB22], this approach has also been extended to audio data. However, during the first and second annotation studies, we observed that the models used for the representation of podcasts and news articles are crucial for the success of a content-based cross-domain recommender system. The aforementioned representation learning models in the literature use general multilingual datasets, while our data consists of German spoken language content and German texts. Therefore, we expect that these models would have to be specifically fine-tuned to be useful in our content-based cross-domain recommender scenario. It can be argued that the podcast segment audio and the obtained transcription could be used to perform the fine-tuning, but this is beyond the scope of this work.

Another approach is not to measure the similarity of podcast segments and news articles directly, but to delegate the similarity quantification to another model. In other words, given a pair of podcast segment features and news article features, the model's task is to predict whether the two are semantically similar or not. Applying this model to all news articles in the dataset, while using a fixed podcast segment, can produce a ranking based on the similarity of the podcast segment and the news articles. This approach is referred to as *Learning-to-Rank (LTR)*. The method used for LTR can be a simple general model such as a Support Vector Machine (SVM) [YK12, QZW+07] or a model developed specifically for ranking applications [CQL+07, LZG+14, PB21, Bur10].

Since ranking models can use features from disjoint spaces, we will follow an LTR approach in this work. However, training a model that can produce a good ranking using all available news articles can be a difficult task that requires a large amount of annotated data. Since we only collected 800 annotated pairs during the annotation study, we aim to build a model that only re-ranks candidate news articles retrieved using Jina sentence embeddings. In other words, we use a two-step process to build the recommender system, which is visualized in Figure 9.1. The first step is to retrieve 20 news articles based on the transcription text of a podcast segment using the Jina sentence embeddings. The second step is re-ranking, where diverse features of a podcast segment (described in Section 9.3) and news articles are concatenated and fed pairwise to a ranking model. The ranking model then aims to sort relevant articles to the top of the list of retrieved articles. The model we use to build this re-ranker is *LambdaMART* [Bur10] and is described in the following section.

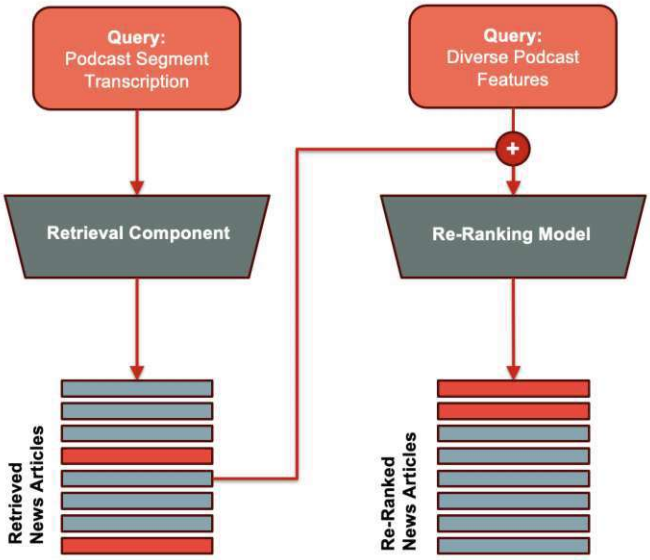

Figure 9.1: Simplified schematic overview of the learning-to-re-rank modeling approach. Articles in grey represent non-relevant articles, articles in red represent relevant articles.

9.2 LambdaMART

LambdaMART is a Learning-to-Rank model that has been successfully applied to a wide variety of problems [Bur10]. LambdaMART is a combination of the *LambdaRank cost calculation* and the gradient boosting technique *Multiple Additive Regression Trees (MART)* [Bur10]. A complete description of LambdaRank and MART is beyond the scope of this thesis. However, in the following sections, we summarize the key ideas of LambdaRank first and then MART and the combination of the two.

Let q be a query (e.g., a podcast segment), P a ranked list of items (e.g., news articles ranked based on the results of a retrieval model), and REL a list of relevance labels obtained during the annotation study. REL has the same ranking as P, and each label $rel_i \in REL$ reflects how relevant a given item $p_i \in P$ is to the query q. Given REL, a utility function like the Discounted Cumulative Gain (DCG)[JK02], which measures the quality of the ranking of the documents, can be computed as follows:

$$DCG = \sum_{i=1}^{|REL|} = \frac{rel_i}{log_2(i+1)}. \tag{9.1}$$

The core idea behind the LambdaRank cost function is to swap the position in the ranked list for each pair of items $p_i, p_j \in P$ and to recompute the utility function for each swap. The difference in C between the swapped list and the original list is accumulated for each item. These accumulated differences in C of swapped items are denoted as Δ. Note that the utility function can be any function that measures the goodness of the model,

such as Mean Average Precision (MAP) or Mean Reciprocal Rank (MRR) [Bur10], and *DCG* is used in this example only for simplicity. Based on the computed Δ values, the items for each query can be optimally ordered with respect to their annotations. If a model is trained that accurately predicts these Δ's using features from queries and items, this model can be used for ranking.

This is where MART comes in. MART is an ensemble of gradient-boosted trees, where each tree tries to correct the mistakes made by the previous trees. Thus, each tree in the ensemble does not directly predict the pairwise Δ's, but rather a set of λ's, which are the Δ's weighted by the error of the previous trees. These error weights are obtained by computing the gradient of the difference between the predicted and desired Δ's with respect to the samples used to build the tree. The final prediction is then made by aggregating the predictions of all trees in the ensemble using a simple sum [Bur10].

We use LambdaMART for this work for three reasons: First, it works directly with features from disjoint embedding spaces, and we do not need to apply techniques such as representation learning first. Second, since each tree in LambdaMART is a simple decision tree, the complexity of the full ensemble model can be controlled by hyperparameters, such as the maximum number of trees to use or the maximum tree depth. Reducing the complexity of the model can be particularly desirable to cope with overfitting (i.e., undergeneralization) of the model when working with a small amount of annotated data. Third, due to the popularity of LambdaMART, it is included in several open source packages such as XGBoost [CHB+15] or LightGBM [KMF+17] and can thus be easily implemented. In this work, we use the LightGBM Python implementation of LambdaMART, which is referred to in the framework as *LGBMRanker*[1].

9.3 Features

As a lot of effort has already been spent in this project, we will not create new features, but will reuse features already created during the annotation studies. For the podcasts, we use the Jina sentence embeddings of the transcriptions and the YAMNet embeddings of the segment audio in a mean and maximum aggregated setting. In addition, we also compute Jina sentence embeddings for the episode titles and episode descriptions, so we have a total of five different features to represent a podcast. For the news articles, we just use the Jina sentence embeddings of the merged paragraphs. How these podcast and news article features are combined to train the desired re-ranking model is described further in Chapter 9.6.

Note that we only use the Jina embeddings to represent the textual attributes of the data, since these features already showed the best results when retrieving the news articles during the second annotation study. It can be argued that more and different features may be beneficial for training a re-ranking model, but this needs to be done in future work.

[1] https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRanker.html

9.4 Evaluation Metrics

To evaluate the models used in this work, we use the two ranking quality metrics Mean Reciprocal Rank (MRR) [Cra09] and Normalized Discounted Cumulative Gain (NDCG) [JK02]. In addition, we define a *beyond-accuracy* metric that measures the topical diversity of the recommendations produced. All of these metrics are described in the following sections. As is common in the literature [VH03, CMY+20, Sob21, CMY+21] and is inherently done by MRR, we average the computed metrics over all queries (i.e., podcast segments). Since the calculation of ranking quality metrics for podcast segments without relevant annotated news articles results in zero, we exclude such podcast segments from the calculation.

9.4.1 Mean Reciprocal Rank (MRR)

The MRR [Cra09] is a popular metric used in information retrieval and recommender system evaluation. For a given query q and a ranked list of recommended items, the Reciprocal Rank (RR) calculates the reciprocal rank of the first relevant item in the list. Averaging the computed reciprocal ranks over all queries in a collection then yields the MRR. More formally, the MRR for a collection of queries Q is computed as follows:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{9.2}$$

where $rank_i$ is the rank of the first relevant item in a list of recommendations [Cra09].

9.4.2 Normalized Discounted Cumulative Gain (NDCG)

Another metric commonly used in information retrieval and in the evaluation of recommendation systems is the Normalized Discounted Cumulative Gain (NDCG) [JK02]. Unlike MRR, NDCG considers not only the rank of the first relevant item, but the rank of all relevant items in a list of recommendations. This makes this metric particularly useful when the goal is to present more than one recommendation to the user. Thus, the NDCG is often calculated at different levels K, where K is the index of the recommendation list up to which the NDCG is calculated. The metric is then written as NDCG@K. For example, if $K = 3$, the NDCG will only be calculated for the first three items in the ranked list of recommendations.

Before we show how the NDCG is calculated, we have to modify the DCG already described in equation 9.1 to include the calculation up to the level K. Again, given a query q, a ranked list of recommended items P, and REL a list of relevance labels, where each label $rel_i \in REL$ reflects how relevant a recommended item $p_i \in P$ is to the query q, the DCG@K is computed as follows:

$$\text{DCG@}K = \sum_{i=1}^{K} \frac{rel_i}{log_2(i+1)}. \tag{9.3}$$

In our setting, $rel_i \in \{0, 1\}$, where $rel_i = 0$ means that a item has been annotated as non-relevant and $rel_i = 1$ means that it has been annotated as relevant.

We can see that the relevance label is discounted by the index at which it appears in the recommendation list, i.e. the DCG is higher for a list with relevant items at the top of the list than for a list with relevant items at the bottom of the list. The second component needed to calculate the NDCG is the Ideal Discounted Cumulative Gain (IDCG). The IDCG@K can be calculated using equation 9.3 from above, but instead of using REL, the IDCG uses $IREL$ for calculation. $IREL$ is the ideal ranked list, sorted descending according to the relevance labels. For example, if REL contains relevant labels at ranks two and four and non-relevant labels at all other ranks, the ideal ranked list will contain relevant labels at ranks one and two.

Now, given DCG@K and IDCG@K, the NDCG@K can be finally calculated as

$$\text{NDCG@}K = \frac{\text{DCG@}K}{\text{IDCG@}K}\text{[JK02]}. \tag{9.4}$$

To evaluate the models used in this work, we compute the NDCG@K for the sequence of $K \in \{1, 3, 5, 10, 15, 20\}$. Since we believe that in the podcast news article scenario, three is a reasonable number of items to display to the user, we choose the NDCG@3 as our target metric for evaluation.

9.4.3 Topical Diversity

Especially in content-based recommender systems, filter bubbles can be a major problem. Therefore, evaluating recommender systems using metrics that provide information about result diversity is especially important for such systems. One method often used to evaluate result diversity is intra-list similarity (ILS). The ILS was first defined by Ziegler et al. [ZMKL05] in 2005, when they proposed an approach to diversify the topics in a book recommendation setting. For a set P of recommended items, ILS is computed as follows

$$\text{ILS}(P) = \frac{\sum_{p_i \in P} \sum_{p_j \in P, p_i \neq p_j} sim(p_i, p_j)}{2} \tag{9.5}$$

where sim is an arbitrary function that measures the similarity between items p_i and p_j.

Today, it is more common to report the average ILS [JBJ23], as it was first defined as *Diversity* by Bradley et al. [BS01] in their work on improving recommendation diversity in content-based recommender systems. In this work, we use the topic assignments obtained during the analysis of the news article dataset (see Chapter 5.2) to quantify the similarity of two recommended items. The resulting metric, which we denote as *intra-list topical diversity* (ILTD), is then computed as follows:

$$\text{ILTD}(P) = \frac{\sum_{p_i \in P} \sum_{p_j \in P, p_i \neq p_j} sim_{TA}(p_i, p_j)}{(|P|(|P| - 1))/2} \tag{9.6}$$

with sim_{TA} defined as

$$sim_{TA}(p_i, p_j) = \begin{cases} 1 & \text{if TA}(p_i) = \text{TA}(p_j) \\ 0 & \text{else} \end{cases} \tag{9.7}$$

where $\text{TA}(p_i)$ is the assigned topic of a particular topic model for the recommended item p_i. Our used topical diversity metric ILTD is very similar to the intra-list topical diversity used by Sertkan and Neidhardt in [SN23], but differs in the similarity measure used for two recommended items.

9.5 Train-Test Split

To measure the performance of our model on unseen data, we split our annotated dataset into train and test parts. As is the case for datasets such as MS MARCO [NRS+16], we perform the split in a query-aware manner (that is, podcast segment-aware) so that a query present in the training set is not also present in the test set. We also stratify the split based on whether or not a query has annotated relevant items, ensuring that both sets contain queries with and without relevant items. To keep the distribution of podcast shows similar for both the training and test splits, we perform the splitting separately for each podcast show. Figure 9.2 shows an overview of the resulting train-test split described above.

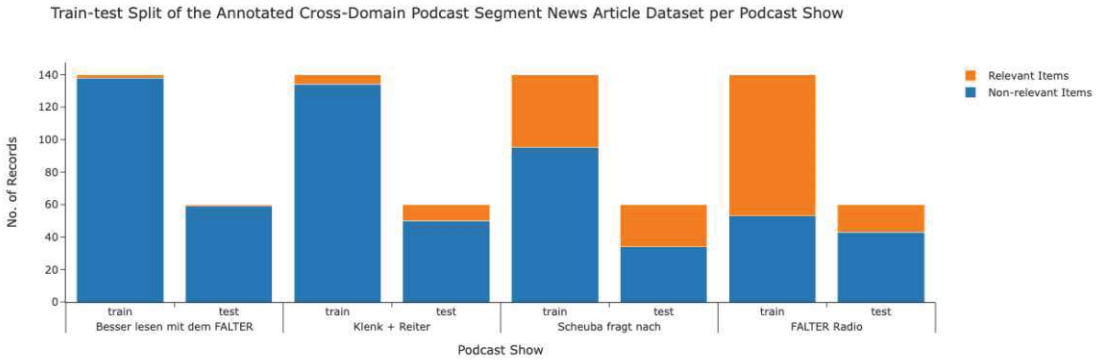

Figure 9.2: Distribution after train-test split for the annotated cross-domain podcast segment news article dataset per show.

Note that for "Besser lesen mit dem FALTER", only three relevant items were annotated, and thus the train set contains only two relevant items and the test set only one relevant item. It is also important to note that the distribution of relevant and non-relevant items per show is not completely equal, because we only performed an approximate stratification on a query basis, using the binary information of whether a query has relevant items annotated or not as a stratification variable.

9.6 Training

To evaluate the effectiveness of different podcast features when building a model, our approach is to train not just one model, but one model for each combination of podcast features, using the same features to represent news articles for each model. That being said, we start by creating combinations of the five podcast features described in Section 9.3 in all lengths. This results in a list of 31 feature sets. The smallest feature sets include only one feature, such as the Jina embeddings of the podcast episode title, and the largest feature set includes all five available features. For each of the 31 sets, we concatenate the features in the set and the constant news article features to obtain one feature vector for each record that represents a podcast segment news article pair.

Given the 31 different matrices representing the features of our train dataset, we train a LambdaMART re-ranker for each of them. To optimize the hyperparameters of each of these models, we train them using 100 different hyperparameter settings obtained from the Optuna hyperparameter optimization framework [ASY+19] using a random seed of 42. For each discrete numerical parameter, we specify an interval of values from two to 128. For each continuous numerical parameter, we specify an interval of values from 1e-8 to 1.0. A description of all the hyperparameters we used in the optimization can be found on the LightGBM website[2].

For each hyperparameter setting, we measure performance in terms of our target ranking quality metric, NDCG@3, using query-aware 10-fold cross-validation on the training set. Query-aware 10-fold cross-validation means that we divide the training set into 10 folds, again so that queries do not overlap between folds. We then train a model using nine of the 10 folds and evaluate it using the remaining fold. Each of the 10 folds is used once as a validation fold, resulting in the training of 10 different models, each using one of the 10 different validation folds. The performance across all validation folds is then averaged and reflects the performance on the training data. This is done to ensure that the model does not overfit (i.e., undergeneralize) and will work on unseen data. The results of each feature set and hyperparameter setting, 3100 different results in total, are stored in the experiment tracking platform Weights & Biases[3]. Finally, after training is complete, we use the model that scored the highest in NDCG@3 for each podcast feature set to make predictions on the test set.

9.7 Results

In the following sections, we report the results obtained during the modeling. First, we show a comparison of different podcast segment representations (that is feature sets) for the trained re-ranking model. Follwed by a comparison of the re-ranking approach to other pure retrieval models used in this work.

[2]https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRanker.html
[3]https://wandb.ai/site

9.7.1 Comparison of Feature Sets

As mentioned above, for each of the LambdaMART re-ranking models using one of the 31 different podcast feature combinations trained with 100 different hyperparameter settings, we select the model that yields the highest NDCG@3 for each feature set separately. For each model, we compute the evaluation metrics described in 9.4. To examine differences in performance, we perform a paired two-sided Student's t test at the significance level 5%, as also suggested in [SAC07]. For the sake of readability, we only report the three best and three worst feature configurations for the re-ranking model in Table 9.1 and Table 9.2. The features "TITLE", "DESC", and "TRANS" correspond to the Jina embeddings of the episode title, episode description and the segment transcription of the podcasts. The features "YMEAN" and "YMAX" are the YAMNet embeddings of a podcast segment using mean and maximum aggregation. The "+" indicates the concatenation of these features. All model configurations were trained using Jina embeddings of the merged news article paragraphs.

Note that we do not report the topical diversity for the different feature sets, since it is the same for all models. This is the case since the re-ranking does not affect the selected items, but only their ranking.

ID	Features	MRR	NDCG@1	NDCG@3	NDCG@5
a	DESC+TRANS+YMAX+YMEAN	**0.764**	**0.667**	**0.667**^{d}	0.574^{d}
b	TITLE+TRANS	0.739	**0.667**	0.617^{d}	**0.607**^{d}
c	YMAX+YMEAN	0.693	**0.667**	0.568	0.573
d	TITLE+TRANS+YMAX+YMEAN	0.606	0.5	0.451	0.464
e	DESC+YMAX	0.554	0.333	0.432	0.495
f	DESC+TITLE+YMAX+YMEAN	0.468	0.333	0.372	0.454

Table 9.1: MRR, NDCG@1, NDCG@3, NDCG@5 for the three best (ID = a, b, c) and the three worst (ID = d, e, f) performing podcast feature sets with respect to NDCG@3 when training a LambdaMART re-ranker. A higher value represents higher ranking quality. The ID column identifies specific feature settings. The best results are highlighted in bold. Second best results are underlined. Superscripts indicate significant differences from the model with a particular ID, using a paired two-sided Student's t-test at $\alpha = 5\%$.

We can see that the best performing re-ranker (ID = a) in terms of MRR, NDCG@1, and our target metric, NDCG@3, was trained using the episode description, segment transcription, and mean and maximum aggregated YAMNet audio embeddings to represent a podcast. The second best re-ranker (ID = b) with respect to the target metric uses the title and transcription embeddings. This model is also the one with the best performance in terms of the NDCG at higher levels $K = 10, 15, 20$. It is interesting to see that although all metrics are higher for the best three re-rankers than for the worst three, only for some metrics significant differences are detected between the models with ID = a and ID = d and ID = b and ID = d and f. Also, for the 25 other models using

ID	Features	NDCG@10	NDCG@15	NDCG@20
a	DESC+TRANS+YMAX+YMEAN	0.569	0.691	0.755^{d}
b	TITLE +TRANS	**0.616**^{d}	**0.706**^{f}	**0.764**^{d}
c	YMAX+YMEAN	0.515	0.597	0.724
d	TITLE+TRANS+YMAX+YMEAN	0.496	0.596	0.697
e	DESC+YMAX	0.544	0.643	0.706
f	DESC+TITLE+YMAX+YMEAN	0.508	0.549	0.679

Table 9.2: NDCG@10, NDCG@15, NDCG@20 for the three best (ID = a, b, c) and the three worst (ID = d, e, f) performing podcast feature sets with respect to NDCG@3 when training a LambdaMART re-ranker. A higher value represents higher ranking quality. The ID column identifies specific feature settings. The best results are highlighted in bold. Second best results are underlined. Superscripts indicate significant differences from the model with a particular ID, using a paired two-sided Student's t-test at $\alpha = 5\%$.

other feature sets, which are not shown here, no significant differences were detected. This is contrary to our expectation that one re-ranking model using a particular feature set would outperform all others.

9.7.2 Comparison of Models

In this section, we compare the performance of our re-ranking approach, to all retrieval models we used to generate news article candidates during the annotation studies. These retrieval models are mDPR (see Section 6.3.2), STS-RoBERTa (see Section 8.3.3), BM25 (see Section 8.3.2) and Jina sentence embeddings (see Section 8.3.4) which we use to generate the recommendation lists fed to the re-ranker. For the re-rankers, we display the best and worst model configuration (that are the models with ID = a and ID = f in Table 9.1 and 9.2). The results in terms of ranking quality metrics are reported in Table 9.3 and 9.4. The results regarding the beyond-accuracy measure intra-list topical diversity are shown in Table 9.5.

We can see that our proposed approach using a LambdaMART-based re-ranker outperforms all other models in terms of ranking quality metrics. The models with ID = a and b are even significantly outperformed for the metrics MRR, NDCG@3, NDCG@5, NDCG@10, NDCG@15 and NDCG@20. For NDCG@10, NDCG@15 and NDCG@20, the best re-ranker also outperforms BM25 with ID = c. Furthermore, we can see that for NDCG@3, NDCG@5, NDCG@10 and NDCG@20, the worst re-ranker shows the second best performance. For the other metrics, the second best model is the Jina embedding model without re-ranking.

It is important to note that the re-ranking model does not significantly outperform the Jina sententence embeddings for any ranking quality metric, using a 5% significance level. For STS-RoBERTa with ID = b, we observe that all metrics are zero. This is due to the

ID	Model	MRR	NDCG@1	NDCG@3	NDCG@5
a	mDPR	0.167	0.167	0.117	0.085
b	STS-RoBERTa	0.000	0.000	0.000	0.000
c	BM25	0.519	0.500	0.323	0.234
d	Jina sentence embeddings	0.584^{b}	0.500	0.362	0.379
e	Best re-ranker (LambdaMART)	$\mathbf{0.764}^{ab}$	$\mathbf{0.667}^{b}$	$\mathbf{0.667}^{ab}$	$\mathbf{0.574}^{ab}$
f	Worst re-ranker (LambdaMART)	0.468^{b}	0.333	0.372	0.454

Table 9.3: Comparison of best and worst LambaMART re-ranking models against all retrieval models in terms of MRR, NDCG@1, NDCG@3, NDCG@5. A higher value represents higher ranking quality. The ID column identifies specific models. The best results are highlighted in bold. Second best results are underlined. Superscripts indicate significant differences from the model with a particular ID, using a paired two-sided Student's t-test at $\alpha = 5\%$.

ID	Model	NDCG@10	NDCG@15	NDCG@20
a	mDPR	0.055	0.055	0.055
b	STS-RoBERTa	0.000	0.000	0.000
c	BM25	0.183	0.186	0.186
d	Jina sentence embeddings	0.488^{ab}	0.579^{abc}	0.675^{abc}
e	Best re-ranker (LambdaMART)	$\mathbf{0.569}^{abc}$	$\mathbf{0.691}^{abc}$	$\mathbf{0.755}^{abc}$
f	Worst re-ranker (LambdaMART)	0.508^{ab}	0.549^{ab}	0.679^{a}

Table 9.4: Comparison of best and worst LambaMART re-ranking models against all retrieval models in terms of NDCG@10, NDCG@15, NDCG@20. A higher value represents higher ranking quality. The ID column identifies specific models. The best results are highlighted in bold. Second best results are underlined. Superscripts indicate significant differences from the model with a particular ID, using a paired two-sided Student's t-test at $\alpha = 5\%$.

fact that no article annotated as relevant was retrieved across all podcast segments using this model.

Looking at the topical diversity measurements of the different models, we can see that mDPR with ID = a has the highest result diversity when LDA and NMF are used for evaluation. When BERTopic is used to calculate the ILTD scores, BM25 (ID = c) has the highest diversity. The second best results are obtained by STS-RoBERTa when looking at the numbers produced by LDA and BERTopic and by BM25 when looking at the ILTD produced by NMF. The ILTD using NMF and BERTopic for mDPR is even significantly different from the Jina sentence embedding-based results. This is also true for STS-RoBERTa using LDA and BERTopic and BM25 using BERTopic to compute the ILTD.

ID	Model	ILTD_{LDA}	ILTD_{NMF}	$\text{ILTD}_{BERTopic}$
a	mDPR	**0.060**	**0.060**^{bdef}	0.023^{def}
b	STS-RoBERTa	<u>0.197</u>^{def}	0.143	<u>0.022</u>^{def}
c	BM25	0.278	<u>0.096</u>	**0.014**^{def}
d	Jina sentence embeddings	0.382	0.122	0.058
e	Best re-ranker (LambdaMART)	0.382	0.122	0.058
f	Worst re-ranker (LambdaMART)	0.382	0.122	0.058

Table 9.5: Comparison of best and worst LambaMART re-ranking models against all retrieval models in terms of the intra-list topical diversity (ILTD) using LDA, NMF, and BERTopic topic models. A lower value represents higher topical diversity. The ID column identifies specific models. The best results are highlighted in bold. Second best results are underlined. Superscripts indicate significant differences from the model with a particular ID, using a paired two-sided Student's t-test at $\alpha = 5\%$.

It is particularly interesting to see that when the embedding-based mDPR model is used for retrieval, the topic diversity calculated using a term frequency-based topic model (LDA and NMF) is the highest. When using a term frequency-based retrieval model (BM25), the topic diversity is highest when using an embedding-based topic model (BERTopic) for the calculation of ILTD. Finally, we can observe that the models that show the best performance in terms of ranking quality metrics (ID = d, e, f) show the lowest result diversity across all three metrics. Note that the reported numbers are similar for models with ID = d, e, f because the re-ranking only affects the order of the results, not the retrieved items.

9.7.3 Comparison of Podcast Shows

Finally, we compare the performance of the best LambdaMART-based re-ranker across the four different podcast shows in the test set. We do this by selecting only test set predictions of podcast segments associated with a particular podcast show, as shown in Figure 9.2 in the train-test split section. Because different queries (i.e., podcast segments) and different numbers of queries with different numbers of relevant and non-relevant news articles are used for each show, we do not perform significance tests to compare the results here. The calculated ranking quality metrics are shown in Tables 9.6 and 9.7. The beyond-accuracy metric ILTD is shown in Table 9.8.

We can see that our proposed approach shows the best results in terms of ranking quality metrics for the podcast show "Klenk + Reiter". For all metrics except NDCG@10, the proposed approach shows the highest values for this show. Moreover, for MRR, NDCG@1, NDCG@3 even optimal values of one are achieved. The second best performance is shared between "FALTER Radio" and "Scheuba fragt nach" with "Scheuba fragt nach" also having optimal values for MRR and NDCG@1. The re-ranking approach shows the worst results when making recommendations based on podcast segments from "Besser lesen

Show	MRR	NDCG@1	NDCG@3	NDCG@5
FALTER Radio	0.750	0.500	0.765	0.626
Besser lesen mit dem F.	0.083	0.000	0.000	0.000
Scheuba fragt nach	**1.000**	**1.000**	0.735	0.670
Klenk + Reiter	**1.000**	**1.000**	**1.000**	**0.854**

Table 9.6: Comparison of podcast shows using best LambdaMART re-ranker in terms of MRR, NDCG@1, NDCG@3, NDCG@5. A higher value represents higher ranking quality. The best results are highlighted in bold. Second best results are underlined.

Show	NDCG@10	NDCG@15	NDCG@20
FALTER Radio	0.634	0.737	0.799
Besser lesen mit dem F.	0.000	0.270	0.270
Scheuba fragt nach	**0.730**	0.770	0.875
Klenk + Reiter	0.684	**0.860**	**0.913**

Table 9.7: Comparison of podcast shows using best LambdaMART re-ranker in terms of NDCG@10, NDCG@15, NDCG@20. A higher value represents higher ranking quality. The best results are highlighted in bold. Second best results are underlined.

mit dem FALTER", where we can even observe that NDCG@1, NDCG@3, NDCG@5 and NDCG@10 are zero. It is important to note that for this show only two relevant articles were included in the training set and one relevant article in the test set. This fact may explain the exceptionally poor performance for podcast segments of this show.

Show	ILTD_{LDA}	ILTD_{NMF}	$\text{ILTD}_{BERTopic}$
FALTER Radio	0.484	0.116	0.084
Besser lesen mit dem F.	0.323	0.119	0.044
Scheuba fragt nach	0.474	0.160	0.082
Klenk + Reiter	**0.246**	**0.109**	**0.029**

Table 9.8: Comparison of podcast shows using best LambdaMART re-ranker in terms of the intra-list topical diversity (ILTD) using LDA, NMF, and BERTopic topic models. A lower value represents higher topical diversity. The best results are highlighted in bold. Second best results are underlined.

In terms of topical diversity, we can see that "Klenk + Reiter" has the lowest values and thus the highest diversity of results. This is an interesting result because it means that for this show the most relevant but also the most diverse recommendations are generated. This was not expected based on the finding in the previous section that higher ranking performance comes at the expense of result diversity. Again, these results show that the ranking quality and diversity of recommendations is strongly influenced by the different

structure and content of individual podcast shows.

9.8 Recommendations Beyond Segments

So far, we have only described how to build a recommender system capable of recommending news articles based on podcast segments. However, in a real-world setting, it is probably more desirable to display recommended news articles based on full podcast episodes, and this can be easily achieved using the following method:

Given a podcast episode divided into segments, data cleaning as described in Section 8.2.1 can be performed to remove segments with little contextual information from the recommendation process. Now, using the N remaining segments, the top K recommendations for each segment can be generated by first performing item retrieval followed by re-ranking as described in this chapter. This results in N different ranked lists of K news article recommendations for an entire podcast episode. These N ranked lists can then be combined into a final list using fusion algorithms such as Borda Count [AM01] or Condorcet Fuse [MA02]. If relevance scores between full podcast episodes and news articles are available, there are even optimizeable algorithms like Bayes Fuse [AM01] or Slide Fuse [LTCD08] that can combine these ranked lists based on these relevance scores. Given this final combined list, the desired number of recommendations can be presented to the user by selecting the first items on the list.

CHAPTER 10

Discussion

In this chapter, we discuss the main findings of this work. First, we discuss the results gained for the data and annotations, followed by differences in the representation of podcasts. Finally, we review the discoveries made when building and evaluating a content-based cross-domain recommender system between podcasts and news articles.

10.1 Data and Annotation

An extensive amount of work in this thesis went into building a cross-domain dataset between podcast segments and news articles that can be used to evaluate a content-based recommender system. The work was divided into data analysis (described in Chapter 4, 5 and 7), data preparation (described in Chapter 6 and 8), and data annotation (also described in Chapter 6 and 8).

We found that although we were working with textual data for both podcasts (i.e., their transcriptions) and news articles, the two data domains are very different in structure. Apart from the origin and length of entities of both data domains, we found that the transcriptions of podcast segments derived from the audio signal contain less topical and contextual information than the texts of news articles. Thus, applying the well-known term frequency-based topic modeling techniques LDA and NMF to both data domains unexpectedly led to random results for the podcast data, while the same models and settings produced reasonable results for the news articles. Only using the Transformer-based topic modeling technique BERTopic, we were able to identify topics which we found to be reasonable and overlap with the topics contained in the news articles.

One finding from BERTopic was that, contrary to our expectations, the podcast and news article datasets contained records that were not in German. Applying a language classification model to both datasets, we found that almost 3000 podcast segments and only 14 news articles could be classified into different languages. Although this discovery

could be made directly from the news article texts, for the podcast dataset, it was necessary to create transcriptions first. Otherwise, this information would have been hidden in the audio signal.

Using podcast and news article datasets, we built a first annotation study. The first step was to sample podcast segments, followed by retrieving potentially relevant news articles using a multilingual dense retrieval model. With these pairs of podcast segments and news articles, we then conducted an annotation study with three annotators. The results of this study showed that the annotation of relevance between podcast segments is a highly subjective task, and the agreement between the annotators was quite low. Furthermore, we observed strong differences in the agreement and the number of pairs annotated as relevant between the different podcast shows. After the study, we conducted a detailed analysis of the results and collected feedback from the participating annotators. All annotators reported that some of the podcast segments were of low contextual quality and that some of the retrieved news articles appeared to be random.

Based on the results of the analysis and the collected feedback, we developed an improved approach for a second annotation study. One improvement was to apply advanced data cleaning methods based on named entity recognition, audio features, and segment position within the full podcast episode to improve the contextual quality of the selected podcast segments. Another improvement was to explore different models for retrieving news articles. Here, we found that the choice of the news article retrieval model plays a crucial role in building an annotation study and that the quality of the retrieved articles differed greatly between the different models. In addition to the improvements mentioned above, we created a guideline to help annotators decide which annotation to select in boundary cases.

The completion of this improved annotation study results in an increase in annotator agreement of 51%, clearly showing that the proposed improvements were successful. However, even for the improved annotation study, the results show a strong difference in annotator agreement and the number of pairs annotated as relevant between the different podcast shows. This outcome is contrary to our expectations and points to the need for modifications for different podcast shows when creating a dataset for evaluating content-based recommender systems for podcasts.

In conclusion, we find that although we can derive textual representations from podcasts, these textual representations are very different from texts in news articles. We found that the structure of these texts can also vary greatly between different podcast shows and even between parts of particular podcast episodes. These findings should be kept in mind when conducting research that involves textual representations of podcast audio. Another important finding is that designing and conducting an annotation study is a very complex and time-consuming task. While reviewing the literature on podcast recommender systems and content-based podcast retrieval methods, we found that several authors used evaluation proxies, such as measuring whether recommended items belong to a category similar to the query items. However, we believe that unless a large open dataset

is available for research or can be collected online (i.e., by tracking users), performing annotation studies, as done in this work, is the only valid approach for system evaluation.

10.2 Podcast Representation

In this study, we investigated different attributes of podcasts and different methods of representing them for use in a recommender system scenario. We described the statistics of metadata attributes, such as titles and episode descriptions, that come with every regular podcast. For these two attributes, we again observed differences in length and structure between particular podcast shows. While two of the shows used fixed naming schemes for episode titles that provided no informational value, two other podcast shows included titles that provided a concise overview of the main topic discussed in the episode. For the episode descriptions, we did not find fixed schemes, but still large differences in the number of words used for description.

By feeding the audio signal from the podcast episodes into an automatic speech recognition model, we created transcriptions of the spoken word content. We evaluated the quality of these transcriptions by manually creating reference transcriptions for samples of all podcast shows. Again, we see differences in the errors obtained between the different podcast shows. We argue that these differences in errors can be explained by the dialect used and the number of people participating in the podcast episode.

Since the main source of content in a podcast is spoken word content, and the derived transcriptions are textual representations of those spoken words, the transcriptions are the attributes that have the highest information density beyond the audio signal. Due to advances in natural language processing and research conducted in content-based podcast retrieval, where podcast transcriptions are also the main attribute used, podcast transcriptions are inherently the most useful attributes for content-based retrieval and recommendation scenarios. Furthermore, in a podcast search setting, research has shown that using podcast transcriptions yields the most relevant search results. However, since no such comparison has been made in research, we investigated whether using episode titles, episode descriptions, and features derived from the audio signal can be beneficial in a content-based recommender system scenario. To represent textual attributes, we created embeddings using the same model we used to retrieve potentially relevant news articles in the second annotation study. To extract audio features, we used the well-known audio classification and embedding model YAMNet. We then built 31 different combinations of podcast features and used them to train re-ranking models, each using one of the feature combinations. When comparing the effectiveness of the resulting models, we found that, contrary to expectations, no combination of podcast features significantly outperformed all others. However, as we found in the evaluation of the annotation studies, there is a strong difference between the four podcast shows included in the dataset. We saw that there is a strong divergence between the ranking quality metrics and the topical diversity. Especially interesting is the finding that the podcast show “Klenk + Reiter” achieved the best results in both evaluation criteria, while we found that topical diversity is at

the expense of ranking quality when we examined the measures aggregated for all shows. Although these were not our prior expectations, these differences between podcast shows were already predicted by the podcast expert during the qualitative interview.

It can be argued that the results observed in the effectiveness comparison of podcast attributes are caused by the limited amount of training data, the way we created features from the attributes, or the re-ranking approach in general. Furthermore, it could be argued that the results of our evaluation are biased by the podcast segment transcription-based approach when retrieving potentially relevant news articles in the annotation study. All of the above points have some validity, and we therefore recommend repeating the effectiveness comparisons of podcast representations in subsequent research. Before repeating the experiments, the subsequent work should focus on building a large dataset. Annotations in this dataset should be obtained offline and at the segment level, as was done in this work, but also online, using user clicks as implicit indicators of relevance between podcast episodes and news articles. Beyond comparing the effectiveness of different attributes of podcasts, further research should investigate whether features can be extracted from podcast shows that explain the difference in recommendation quality between them. Furthermore, it should be investigated whether it would be beneficial to use different segment lengths or representations for different podcast shows, and based on the extracted features, methods should be tailored to the specific properties of the shows.

10.3 Building and Evaluating a Content-Based Cross-Domain Podcast Recommender

Besides comparing the effectiveness of different podcast representations in a recommender system, the main goal of this work was to investigate how to build and evaluate a content-based cross-domain podcast recommender system. The basis for achieving this goal was the creation of an annotated dataset between podcast segments and news articles that could be used for training and evaluation of models. We found that this process is tedious and contains many pitfalls, such as insufficient data cleaning, selection of podcast segments without contextual information, retrieval of arbitrary news articles, and lack of guidelines. After applying several modifications to our initial annotation approach, we were finally able to obtain a dataset with annotations that were reliable enough to draw conclusions.

The simplest form of a content-based cross-domain recommender system was already created when we retrieved potentially relevant news articles during the construction of the annotation studies. In total, four different models were explored for this task, three based on the Transformer architecture and one based on term frequencies. We further refined the recommender system by training a re-ranker model, which re-ranks the retrieved news articles according to the retrieval model finally used. Using the annotated dataset for evaluation, we saw that our proposed two-step re-ranking approach showed substantial improvements compared to all other models used in this work. Also, the model used for news article retrieval in the second study (Jina sentence embeddings)

outperformed all other models except the re-rankers in terms of ranking quality metrics, although this outperformance did not reflect a significant difference from all models. Another finding was that although the Jina sentence embedding model and especially our re-ranking approach outperformed all other models in terms of ranking quality metrics, this was at the expense of the three topic diversity measures computed using the two term frequency-based topic models LDA and NMF and the Transformer-based topic model BERTopic. Interestingly, we could observe that the embedding-based retrieval model mDPR achieved the highest diversity when evaluated with with the term frequency-based topic models, whereas the term frequency-based BM25 model achived the highest diversity when evaluated with the Transformer-based BERTopic. Since our final modeling approach led to the lowest diversity of results, improving the result diversity offers an opportunity for further research in this direction.

In summary, in this work we have proposed an approach to build and evaluate a non-personalized content-based cross-domain recommender system between podcasts and news articles that outperforms baseline models. The proposed system should be seen as a first version that can be used to collect data based on user interactions. Furthermore, we have shown how this system can be evaluated in terms of recommendation performance (i.e., ranking quality) and beyond accuracy metrics (i.e., topical diversity) using a manually annotated dataset. It can be argued that the evaluation contains a bias due to the way in which potentially relevant news articles were retrieved in the annotation study. This is true to some extent. To make the comparisons more meaningful, experiments should be repeated using more manually annotated data as well as data based on real user interactions. Collecting data based on interactions between users and items is especially interesting, since it also enables the application of cross-domain approaches that go beyond the similarity of content.

CHAPTER 11

Conclusion

This chapter describes the conclusion of this research. First, we summarize the main steps performed and the findings obtained during this thesis. Then, we describe the contributions to the formulated research questions made by this work. Finally, we describe limitations and motivate future work in the area of content-based cross-domain podcast recommender systems.

11.1 Summary

In this work, we described the complete process necessary to build and evaluate a content-based cross-domain recommender system from scratch. The first step in this process was to review the existing approaches in the literature. We found that existing solutions for podcast recommender systems either use closed-source datasets or lack validity in evaluation. Furthermore, we found that although there are several publications in the literature regarding content-based podcast retrieval and cross-domain recommender systems, none of the approaches could be directly applied to our problem. After reviewing the literature, we conducted an expert interview with a podcast creator from *FALTER*, which helped us gather requirements and constraints for building the recommender system.

Given this strong theoretical foundation, we obtained podcast and news article datasets from our industry partner. We conducted an in-depth analysis of both datasets using descriptive statistics and topic modeling techniques. At this stage of the work, we were already able to discover strong differences between podcast shows, as well as between podcasts and news articles in general. Using both datasets, the next step was to create relevance labels between them by building and running an annotation study. In doing so, we found that this process is time-consuming, not trivial, and without major improvements to our initial study approach, we would not be able to achieve sufficient agreement between annotators, and thus reliable data.

Following the results of the first study, we created a second annotation that included improvements in podcast segment selection, news article retrieval, and overall study design. Improvements of the study were aided by the extraction of audio features and named entities from the present datasets. Conducting this improved study ultimately resulted in higher inter-annotator agreement, which was satisfactory to us.

Based on the annotated dataset, we then developed a recommender system approach. Since only a limited amount of annotated data was available for training and evaluation, we decided to employ a learning-to-re-rank approach using a tree-based re-ranking model that optimized the order of retrieved news articles already used in the annotation study. Since one goal of this work was to investigate the effectiveness of different podcast representations, we trained not just one model, but 31 different models, using all possible combinations of podcast representations. The evaluation of these models showed that our approach is able to outperform all other models used in this work in terms of ranking quality measures. At the same time, we observed that ranking quality seems to come at the expense of topical diversity in our setting and that there is a strong divergence of ranking quality between the different podcast shows. With respect to the different podcast feature sets, we did not observe that one type of representation significantly outperformed all others.

In conclusion, we believe that this thesis has vividly illustrated the work and steps needed to build and evaluate a content-based cross-domain recommender system from podcasts to news articles from scratch. Furthermore, we have highlighted the pitfalls that can occur when building such a system, and through the creation of our annotated dataset, we have formed a strong basis for further research in this direction.

11.2 Contribution

In this section, we describe the contributions made to the state-of-the-art with respect to the research questions formulated in the introductory section of this thesis. The first formulated research question

- **RQ1:** What is the comparative effectiveness of episode titles, episode descriptions, transcriptions, and audio features as representations for podcasts in terms of recommendation performance?

can be answered by this work as follows: Contrary to our initial expectations, in the proposed recommender system approach, we do not observe that one type of representation significantly outperforms all others in terms of recommendation performance. However, due to advances in natural language processing and research already done in the area of content-based podcast retrieval, podcast transcriptions are inherently the most useful representation. We know that our results regarding RQ1 are strongly influenced by the data we used for evaluation, and therefore we recommend repeating experiments related to this question with larger and different datasets.

Upon completion of this work, we answer the first part of the second formulated research question

- **RQ2a:** How can a cross-domain recommender system between podcast segments and news articles be effectively built?

like the following: In this work, we used four different text retrieval models to retrieve the news article candidates to be used in the annotation studies. These models form the baseline cross-domain recommender systems between podcast segments and news articles in this work. We proposed a two-step approach, including a retrieval stage and a re-ranking stage, to perform content-based recommendation. Our proposed solution outperformed all baseline models in terms of recommendation performance using only a limited amount of annotated data, thus showcasing how to effectively build a cross-domain recommender system employing only item contents.

Finally, this works answers the second part of the second research question

- **RQ2b:** How well does this system perform in terms of recommendation performance, and beyond-accuracy measures compared to a baseline?

as follows: Using our manually annotated cross-domain dataset, we evaluated the proposed recommender system approach in terms of recommendation performance and beyond-accuracy measures. Specifically, we used the MRR and NDCG ranking quality metrics as recommendation performance measures, and a self-defined topical diversity metric based on the topic modeling results of three models as a beyond-accuracy measure. In terms of recommendation performance, our proposed approach outperformed all baselines, but not all baselines were significantly outperformed. In terms of beyond-accuracy measures, our approach was unable to outperform any of the baselines. However, an examination of the results at the podcast show level showed that there is a strong relationship between specific podcast shows and their result diversity.

In summary, this work provides quantitative and qualitative insights into the design and implementation of an annotation study between podcasts and news articles. Moreover, the work also contributes insights into building a content-based cross-domain recommender system and highlights possible pitfalls. Finally, this thesis describes limitations of the described approaches and motivates opportunities for future research.

11.3 Limitations and Future Work

As described in the discussion of the results of this work, the main limitation is the availability and reliability of data for training and evaluation. The lack of open source user-item interaction data in the podcast domain hinders the ability to build personalized collaborative filtering or sequence-based recommender systems. Although we have created an annotated dataset in this study that can be used for building and evaluating content-based recommender system approaches, this dataset contains a bias induced by the selection of annotation candidates. Therefore, this fact should always be kept in mind when analyzing the results obtained for evaluation using this dataset. Therefore, our main suggestion for future work in the area of content-based cross-domain podcast recommender systems, and podcast recommender systems in general, is the collection of larger datasets. These datasets should be collected using manual annotation as described in this paper, but also by tracking the user behavior of a deployed recommender system.

Furthermore, we found strong differences in performance between different podcast shows used in this work. We recommend investigating the origin of these differences and exploring the possibility of using different approaches for different show categories. Also, including more than the four podcast shows of *FALTER* in the investigation.

List of Figures

List of Tables

Bibliography

[ABB+19] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In Waleed Ammar, Annie Louis, and Nasrin Mostafazadeh, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[ABD+19] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *CoRR*, abs/1912.06670, 2019.

[AM01] Javed A Aslam and Mark Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284, 2001.

[AMT+21] Abigail Alexander, Matthijs Mars, Josh C Tingey, Haoyue Yu, Chris Backhouse, Sravana Reddy, and Jussi Karlgren. Audio features, precomputed for podcast retrieval and information access experiments. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 3–14. Springer, 2021.

[Art17] Ron Artstein. Inter-annotator agreement. *Handbook of linguistic annotation*, pages 297–313, 2017.

[ASY+19] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[BCHC09] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[BFG+21] Alexander Bondarenko, Maik Fröbe, Marcel Gohsen, Sebastian Günther, Johannes Kiesel, Jakob Schwerter, Shahbaz Syed, Michael Völske, Martin Potthast, Benno Stein, et al. Webis at trec 2021: Deep learning, health misinformation, and podcasts tracks. In *The Thirtieth REtrieval Conference Proceedings (TREC 2021). National Institute of Standards and Technology (NIST), Special Publication*, pages 500–335, 2021.

[BFWC20] Greg Benton, Ghazal Fazelnia, Alice Wang, and Ben Carterette. Trajectory based podcast recommendation. *arXiv preprint arXiv:2009.03859*, 2020.

[Bla13] Ann E Blandford. Semi-structured qualitative studies. Interaction Design Foundation, 2013.

[BMII12] Andrzej Białecki, Robert Muir, Grant Ingersoll, and Lucid Imagination. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval*, page 17. sn, 2012.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[Bou09] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40, 2009.

[BS01] Keith Bradley and Barry Smyth. Improving recommendation diversity. 2001.

[Bur10] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

[CHB+15] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[CJJ+21] Ben Carterette, Rosie Jones, Gareth F Jones, Maria Eskevich, Sravana Reddy, Ann Clifton, Yongze Yu, Jussi Karlgren, and Ian Soboroff. Podcast metadata and content: Episode relevance and attractiveness in ad hoc search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2247–2251, 2021.

[CKG+19] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019.

[CLRP13] Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.

[CMS13] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[CMY+20] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*, 2020.

[CMY+21] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. Ms marco: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1566–1576, 2021.

[CODP19] Christophe Coupé, Yoon Mi Oh, Dan Dediu, and François Pellegrino. Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Science Advances*, 5(9):eaaw2594, 2019.

[CP09] Andrzej Cichocki and Anh-Huy Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 92(3):708–721, 2009.

[CQL+07] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[Cra09] Nick Craswell. Mean reciprocal rank. *Encyclopedia of database systems*, pages 1703–1703, 2009.

[CRY+20] Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones. 100,000 podcasts: A spoken English document corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5903–5917, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[DBMB21] S Dobbrunz, F Brunner, JL Müller, and P Briken. Interrater reliability of the criteria-based assessment of criminal responsibility in paraphilic disorders. *Der Nervenarzt*, 92:1–8, 2021.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[ESH15] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th international conference on world wide web*, pages 278–288, 2015.

[Fac] Facebook. Language, engine, and tooling for expressing, testing, and evaluating composable language rules on input strings.

[fal] Falter : Die wochenzeitung aus wien. `https://cms.falter.at/b2b/falter_sonderbeilagen/falter/`. Last accessed: 17.10.2023.

[FHP+22] Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages, 2022.

[FQ15] Rosa Falotico and Piero Quatto. Fleiss' kappa statistic without paradoxes. *Quality & Quantity*, 49:463–470, 2015.

[GAR08] Patrick Gratz, Adrian Andronache, and Steffen Rothkugel. Ad hoc collaborative filtering for mobile networks. In *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008)*, pages 355–360. IEEE, 2008.

[GEF+17] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[GMG+23] Michael Günther, Louis Milliken, Jonathan Geuter, Georgios Mastrapas, Bo Wang, and Han Xiao. Jina embeddings: A novel set of high-performance sentence embedding models. *arXiv preprint arXiv:2307.11224*, 2023.

[GMMCQ21] Lidia Gurdiel, Javier Morales Mediano, and Jenny Cifuentes Quintero. A comparison study between coherence and perplexity for determining the number of topics in practitioners interviews analysis. 12 2021.

[GNO20] Petra Galuščáková, Suraj Nair, and Douglas W Oard. Combine and re-rank: The university of maryland at the trec 2020 podcasts track. 2020.

[GRHD21] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image. *Text and Audio*, 2021.

[Gro] Maarten Grootendorst. Visual overview of bertopic algorithm. `https://maartengr.github.io/BERTopic/algorithm/algorithm.html#visual-overview`. Last accessed: 21.03.2024.

[Gro22] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[Gwe11] Kilem L Gwet. On the krippendorff's alpha coefficient. *Manuscript submitted for publication. Retrieved October*, 2(2011):2011, 2011.

[Gü24] Michael Günther. Hugginface model card for jina embeddings v2 base de. `https://huggingface.co/jinaai/jina-embeddings-v2-base-de`, 2024. Last accessed: 03.05.2024.

[Hev07] Alan R Hevner. A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19, 2007.

[HK07] Andrew F. Hayes and Klaus Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1(1):77–89, 2007.

[HK15] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[HLY+21] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122, 2021.

[HSH21] Sebastian Hofstätter, Mete Sertkan, and Allan Hanbury. Tu wien at trec dl and podcast 2021: Simple compression for dense retrieval. In *Proceedings of Text REtrieval Conference (TREC)*, 2021.

[Hug21] John Hughes. krippendorffsalpha: An r package for measuring agreement using krippendorff's alpha coefficient, 2021.

[HZC+17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[imd] Imdb statistics. `https://www.imdb.com/pressroom/stats/`. Accessed: 2024-05-15.

[J+97] Thorsten Joachims et al. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *ICML*, volume 97, pages 143–151. Citeseer, 1997.

[JBJ23] Mathias Jesse, Christine Bauer, and Dietmar Jannach. Intra-list similarity and human diversity perceptions of recommendations: the details matter. *User Modeling and User-Adapted Interaction*, 33(4):769–802, 2023.

[JCC+20] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth JF Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu TREC. Podcasts track overview 2020. In *The 29th Text Retrieval Conference (TREC) notebook. NIST, Gaithersburg, MD, USA*, 2020.

[JCC+21] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth JF Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu TREC. Podcasts track overview 2021. In *The 30th Text Retrieval Conference (TREC) notebook. NIST, Gaithersburg, MD, USA*, 2021.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[JRA23] Basra Jehangir, Saravanan Radhakrishnan, and Rahul Agarwal. A survey on named entity recognition—datasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017, 2023.

[JWYF17] Hamed Jelodar, Yongli Wang, Chi Yuan, and Xia Feng. Latent dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *CoRR*, abs/1711.04305, 2017.

[JZS+21] Rosie Jones, Hamed Zamani, Markus Schedl, Ching-Wei Chen, Sravana Reddy, Ann Clifton, Jussi Karlgren, Helia Hashemi, Aasish Pappu, Zahra Nazari, Longqi Yang, Oguz Semerci, Hugues Bouchard, and Ben Carterette. Current challenges and future directions in podcast information access. *CoRR*, abs/2106.09227, 2021.

[KC16] Sneha Khatwani and MB Chandak. Building personalized and non personalized recommendation systems. In *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pages 623–628. IEEE, 2016.

[KCP15] Da Kuang, Jaegul Choo, and Haesun Park. Nonnegative matrix factorization for interactive topic modeling and document clustering. *Partitional clustering algorithms*, pages 215–243, 2015.

[KDVBL20] Chris Kamphuis, Arjen P De Vries, Leonid Boytsov, and Jimmy Lin. Which bm25 do you mean? a large-scale reproducibility study of scoring variants.

In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 28–34. Springer, 2020.

[KMF+17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[KPJK16] Hanna Kallio, Anna-Maija Pietilä, Martin Johnson, and Mari Kangasniemi. Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing*, 72(12):2954–2965, 2016.

[KPR+19] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[Kri11] Klaus Krippendorff. Computing krippendorff's alpha-reliability, 2011.

[LCS97] Dik L Lee, Huei Chuang, and Kent Seamons. Document ranking and the vector-space model. *IEEE software*, 14(2):67–75, 1997.

[lis] Podcast stats: How many podcasts are there? `https://www.listennotes.com/podcast-stats/`. Accessed: 2024-05-15.

[LLZ+23] Xinting Liao, Weiming Liu, Xiaolin Zheng, Binhui Yao, and Chaochao Chen. Ppgencdr: A stable and robust framework for privacy-preserving cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4453–4461, 2023.

[LML+21] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362, 2021.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[LPS16] Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6, 2016.

[LSC+17] Jixiong Liu, Jiakun Shi, Wanling Cai, Bo Liu, Weike Pan, Qiang Yang, and Zhong Ming. Transfer learning from app domain to news domain for dual cold-start recommendation. In *RecSysKTL*, pages 38–41, 2017.

[LSHL20] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70, 2020.

[LTCD08] David Lillis, Fergus Toolan, Rem Collier, and John Dunnion. Extending probabilistic data fusion using sliding windows. In *Advances in Information Retrieval: 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings 30*, pages 358–369. Springer, 2008.

[LZG+14] Yanyan Lan, Yadong Zhu, Jiafeng Guo, Shuzi Niu, and Xueqi Cheng. Position-aware listmle: A sequential learning process for ranking. In *UAI*, volume 14, pages 449–458, 2014.

[MA02] Mark Montague and Javed A Aslam. Condorcet fusion for improved retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 538–548, 2002.

[May20] P. May. Cross english german roberta for sentence embeddings. `https://huggingface.co/T-Systems-onsite/cross-en-de-roberta-sentence-transformer`, 2020. Last accessed: 03.05.2024.

[May21] Philip May. Machine translated multilingual sts benchmark dataset. 2021.

[MBS00] Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400, 2000.

[MHM20] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[MKS+24] Isabelle Mohr, Markus Krimmel, Saba Sturua, Mohammad Kalim Akram, Andreas Koukounas, Michael Günther, Georgios Mastrapas, Vinit Ravishankar, Joan Fontanals Martínez, Feng Wang, et al. Multi-task contrastive learning for 8192-token bilingual text embeddings. *arXiv preprint arXiv:2402.17016*, 2024.

[MMD+04] I. McCowan, Darren Moore, J. Dines, D. Gática-Pérez, Mike Flynn, P. Wellner, and H. Bourlard. On the use of information retrieval measures for speech recognition evaluation. 2004.

[MOG08] Junta Mizuno, Jun Ogata, and Masataka Goto. A similar content retrieval method for podcast episodes. In *2008 IEEE Spoken Language Technology Workshop*, pages 297–300. IEEE, 2008.

[MTMR22] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

[MV15] Apra Mishra and Santosh Vishwakarma. Analysis of tf-idf model and its variant for document retrieval. In *2015 international conference on computational intelligence and communication networks (cicn)*, pages 772–776. IEEE, 2015.

[MXM+24] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Jie Zhou. Triple sequence learning for cross-domain recommendation. *ACM Transactions on Information Systems*, 42(4):1–29, 2024.

[NCP+20] Zahra Nazari, Christophe Charbuillet, Johan Pages, Martin Laurent, Denis Charrier, Briana Vecchione, and Ben Carterette. Recommending podcasts for cold-start users based on music listening and taste. *CoRR*, abs/2007.13287, 2020.

[NKST03] Hung T Nguyen, Vladik Kreinovich, Gennady N Solopchenko, and Chin-Wang Tao. Why two sigma? a theoretical justification. In *Soft Computing in Measurement and Information Acquisition*, pages 10–22. Springer, 2003.

[NNT+10] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, JCDL '10, page 215–224, New York, NY, USA, 2010. Association for Computing Machinery.

[NRS+16] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human-generated machine reading comprehension dataset. 2016.

[NS07] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[PB21] Przemysław Pobrotyn and Radosław Białobrzeski. Neuralndcg: Direct optimisation of a ranking metric via differentiable relaxation of sorting. *arXiv preprint arXiv:2102.07831*, 2021.

[Pla20] M. Plakal. Yet another mobile network (YAMNet). `https://github.com/tensorflow/models/tree/master/research/audioset/yamnet`, January 2020. Last accessed: 27.04.2024.

[pod] The business research company: Podcasting global market report 2024. `https://www.thebusinessresearchcompany.com/report/podcasting-global-market-report`. Accessed: 2024-05-15.

[PSG19] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? *CoRR*, abs/1906.01502, 2019.

[PSL21] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

[PT94] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[PXS+20] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. MLS: A large-scale multilingual dataset for speech research. In *Interspeech 2020*. ISCA, oct 2020.

[QZW+07] Tao Qin, Xu-Dong Zhang, De-Sheng Wang, Tie-Yan Liu, Wei Lai, and Hang Li. Ranking with multiple hyperplanes. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 279–286, New York, NY, USA, 2007. Association for Computing Machinery.

[RA23] Muhammad Mukti Raharjo and Fatchul Arifin. Machine learning system implementation of education podcast recommendations on spotify applications using content-based filtering and tf-idf. *Elinvo (Electronics, Informatics, and Vocational Education)*, 8(2):221–230, 2023.

[RBH15] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408, 2015.

[Ree12] Colorado Reed. Latent dirichlet allocation: Towards a deeper understanding. 2012.

[RG19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[RG20] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*, 2020.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[RKX+22] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. 12 2022.

[RNS+18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[RRS22] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*. Springer US Imprint: Springer, 3rd ed. 2022 edition, 2022.

[ŘS10] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[RW94] Stephen E Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 232–241. Springer, 1994.

[RWJ+95] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[SA20] Stefan Schweter and Alan Akbik. Flert: Document-level features for named entity recognition. *arXiv preprint arXiv:2011.06993*, 2020.

[SAC07] Mark D Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632, 2007.

[SDM03] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

[SGZ21] Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3875–3879. IEEE, 2021.

[Smi12] Reginald D. Smith. Distinct word length frequencies: distributions and symbol entropies, 2012.

[SN23] Mete Sertkan and Julia Neidhardt. On the effect of incorporating expressed emotions in news articles on diversity within recommendation models. *decision-making*, 3:11, 2023.

[Sob21] Ian Soboroff. Overview of trec 2021. In *30th Text REtrieval Conference. Gaithersburg, Maryland*, 2021.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[VH03] Ellen M Voorhees and Donna Harman. Overview of trec 2003. In *Trec*, pages 1–13, 2003.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[Wil06] Peter Willett. The porter stemming algorithm: then and now. *Program*, 40(3):219–223, 2006.

[WLC+20] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France, May 2020. European Language Resources Association.

[Wri15] Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.

[WSKB22] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4563–4567. IEEE, 2022.

[XPX+16] Zhou Xing, Marzieh Parandehgheibi, Fei Xiao, Nilesh Kulkarni, and Chris Pouliot. Content-based recommendation for podcast audio-items using natural language processing techniques. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2378–2383. IEEE, 2016.

[YK12] Hwanjo Yu and Sungchul Kim. Svm tutorial-classification, regression and ranking. *Handbook of Natural computing*, 1:479–506, 2012.

[YYH+14] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[ZCMK16] Antonia Zapf, Stefanie Castell, Lars Morawietz, and André Karch. Measuring inter-rater reliability for nominal data–which coefficients and confidence intervals are appropriate? *BMC medical research methodology*, 16:1–10, 2016.

[ZCP+15] Weizhong Zhao, James J Chen, Roger Perkins, Zhichao Liu, Weigong Ge, Yijun Ding, and Wen Zou. A heuristic approach to determine an appropriate number of topics in topic modeling. In *BMC bioinformatics*, volume 16, pages 1–10. Springer, 2015.

[ZCW+19] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. Dtcdr: A framework for dual-target cross-domain recommenda-tion. 2019.

[ZMKL05] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 22–32, New York, NY, USA, 2005. Association for Computing Machinery.

[ZML+21] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512, 2021.

[ZOML23] Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, and Jimmy Lin. Toward best practices for training multilingual dense retrieval models. *ACM Transactions on Information Systems*, 42(2):1–33, 2023.

[ZTO+23] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131, 2023.

[ZWC+20] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, and Xiaolin Zheng. A graphical and attentional framework for dual-target cross-domain recommendation. pages 3001–3008. International Joint Conferences on Artificial Intelligence Organization, 3 2020. Main track.

[ZWC+21] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. Cross-domain recommendation: challenges, progress, and prospects. *arXiv preprint arXiv:2103.01696*, 2021.

[ZZH+23] Chuang Zhao, Hongke Zhao, Ming He, Jian Zhang, and Jianping Fan. Cross-domain recommendation via user interest alignment. In *Proceedings of the ACM Web Conference 2023*, pages 887–896, 2023.

[ZZL+22] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. A survey on cross-domain recommendation: taxonomies, methods, and future directions. *ACM Transactions on Information Systems*, 41(2):1–39, 2022.