ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

ADDRESSING USER COLD START PROBLEM IN AMHARIC YOUTUBE ADVERTISEMENT RECOMMENDATION USING BERT

BY
FIREHIWOT KEBEDE

ADVISOR
Dr. FITSUM ASSAMNEW

A thesis submitted to the School of Electrical and Computer Engineering in partial fulfillment of the requirements for the Degree of Master of Science in Computer Engineering

JUNE, 2024
ADDIS ABABA, ETHIOPIA

ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

The undersigned have examined the thesis titled:

Addressing user cold start problem in Amharic YouTube advertisement recommendation using BERT

BY

Firehiwot Kebede

Approval by Boards of Examiners

Dr. Bisrat Derebssa	_____	_____
Dean, SECE, AAiT	Date	Signature
Dr. Fitsum Assamnew	_____	_____
Advisor	Date	Signature
Dr. Sosina Mengistu	_____	_____
Internal Examiner	Date	Signature
Dr. Menore Tekeba	_____	_____
External Examiner	Date	Signature

Declaration

I, Firehiwot Kebede, declare that this thesis is my original work. All sources of information in this study have been appropriately acknowledged. I further confirm that this thesis has not been submitted either in part or in full for any other requirements to any other learning institution.

Student Name: Firehiwot Kebede Gebrie

Signature: _____

Date: _____

JUNE, 2024

Acknowledgments

Firstly, I want to express my deepest gratitude to the Almighty God for all the blessings He has given me.

Secondly, I would like to express my gratitude to my advisor Dr. Fitsum Assamnew who guided me throughout this research.

I would also like to express my sincere gratitude to the AAiT ICT department, for providing me with the necessary resources, facilities, and technical assistance.

In addition, I would like to express my sincere gratitude to all the Computer Engineering MSc and undergraduate students at AAiT who participated in test data annotation and system evaluation. Without their contributions, completing this research would have been difficult.

Last but not least, I would like to thank my family and friends for their unwavering support, love, and encouragement during all the ups and downs of my academic and personal life.

Abstract

With the rapid growth of the internet and smart mobile devices, online advertising has become widely accepted across various social media platforms. These platforms employ recommendation systems to personalize advertisements for individual users. However, a significant challenge for these systems is the user cold-start problem, where recommending items to new users is difficult due to the lack of historical preference of the user in a content-based recommendation system. To address this issue we propose an Amharic YouTube advertisement recommendation system for unsigned YouTube users where there is no user information like past preference or personal information. The proposed system uses content-based filtering techniques and leverages Sentence Bidirectional Encoder Representations from Transformers (SBERT) to establish sentence semantic similarity between YouTube video titles, descriptions, and advertisement titles. For this research, 4500 data were collected and preprocessed from YouTube via YouTube API, and 500 advertisement titles from advertising and promotional companies. Random samples from these datasets were annotated for evaluation purposes. Our proposed approach achieved a 70% accuracy in recommending semantically related Amharic Advertisements (Ads) to corresponding YouTube videos with respect to the annotated data. At a 95% confidence interval, our system demonstrated an accuracy of 58% to 76% in recommending Ads which are relevant to new users who have no prior interaction history on the platform with the Ads. This approach significantly enhances privacy by reducing the need for extensive data sharing.

Keywords: user cold-start, Content-based filtering, BERT, Online advertisement, recommendation system

Table of Contents

Declaration i
Acknowledgments ii
Abstract iii
List of Figures vii
List of Tables viii
List of Acronyms ix

Chapter 1 **1**

1 Introduction **1**
1.1 Problem Statement 2
1.2 Objectives 2
1.2.1 General Objective 2
1.2.2 Specific Objectives 3
1.3 Contribution 3
1.4 Scope and Limitation 4
1.5 Research Methodology 4
1.6 Organization of the study 5

Chapter 2 **6**

2 Background **6**
2.1 Recommendation Systems 6
2.1.1 Recommendation system approaches 7
2.1.1.1 Collaborative filtering 7
2.1.1.2 Content based filtering 8
2.1.1.3 Hybrid approach 9
2.1.2 Goal of Recommendation Systems 10
2.1.3 Challenges in recommendation systems 11
2.2 Advertisement 11
2.2.1 Digital advertisement 12
2.3 YouTube 12

2.3.1 YouTube Advertisement . . . 13
2.4 Natural Language Processing (Natural Language Processing (NLP)) . . . 13
2.4.1 Low resource language . . . 14
2.4.2 Amharic Language . . . 14
2.5 Related Works . . . 15
2.5.1 Deep learning based recommendation systems . . . 15
2.5.2 Semantic-based recommendation systems . . . 17

Chapter 3 **20**

3 Methodology **20**
3.1 Proposed Method . . . 20
3.2 Data Collection and Processing . . . 20
3.2.1 YouTube API . . . 21
3.2.2 Data cleaning . . . 23
3.2.3 Text Preprocessing . . . 23
3.2.4 Data annotation method . . . 24
3.2.5 K-Means Clustering . . . 25
3.2.6 Elbow Methods (Clustering) . . . 26
3.2.7 TF-IDF . . . 27
3.3 Embedding Models . . . 27
3.3.1 Transformer . . . 30
3.3.2 BERT . . . 33
3.4 Sentence Embedding . . . 35
3.4.1 Sentence Bidirectional Encoder Representations from Transformers 35
3.4.2 Pooling layer . . . 36
3.5 Distance Similarity Measurement . . . 37

Chapter 4 **40**

4 Result and Discussion **40**
4.1 Experimental Setup . . . 40
4.2 Test data preparation . . . 40
4.2.1 Advertsement Data Clustering . . . 41
4.3 Experimental Scenario . . . 43
4.3.1 Experiment I:Model Efficiency in recommendation . . . 43
4.3.2 Experiment II:Model Efficiency in cold start . . . 44
4.4 Evaluation Metrics . . . 44

4.4.1 Accuracy 45
4.4.2 Confidence Interval 45
4.5 Results 46
4.5.1 Experiment I: Proposed model efficiency in recommendation . . . 46
4.5.2 Experiment II: Proposed model efficiency in cold-start 47

Chapter 5 **49**

5 Conclusion and Future work **49**
5.1 Conclusion 49
5.2 Future Works and Recommendations 50
References 51

List of Figures

3.1 Proposed Methodology 21
3.2 YouTube videos data distribution in each category 22
3.3 Sample example from the preprocessed YouTube videos dataset 24
3.4 Transformer architecture [49] 32
3.5 Showcase that Bidirectional Encoder Representations from Transformers (BERT) captures both the left and right context [54] 34
3.6 SBERT architecture to compute similarity scores modified for the research methodology 36
3.7 Angle between two vectors A and B [57] 38

4.1 Elbow Method to decide the number of categories for Amharic advertisement 41
4.2 Comparison of language models in clustering Ads data 42

List of Tables

2.1 Summary of recommendation system approaches 10
2.2 Summary of research approaches to address the cold-start problem in recommendation systems 19

4.1 BERT models fine-tuned for Amharic language 44
4.2 Models performance for YouTube Amharic advertisements on different thresholds of cosine similarity 47

List of Acronyms

Ads Advertisements

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

BPR Bayesian Personalized Ranking

CBF Content Based Filtering

CBOW Continuous Bag of Words

CF Collaborative Filtering

CI Confidence Interval

CLCRec Contrastive Learning for Cold-start Recommendation

ELMo Embeddings from Language Models

GNNs Graph Neural Networks

GPT Generative Pre-trained Transformer

GPU Graphics Processing Unit

LSTM Long Short-Term Memory

MLM Masked Language Modeling

mBERT multi-lingual Bidirectional Encoder Representations from Transformers

NLP Natural Language Processing

NSP Next Sentence Prediction

PMF Probabilistic Matrix Factorization

RNNs Recurrent Neural Networks

RSs Recommendation Systems

SBERT Sentence Bidirectional Encoder Representations from Transformers

SmBERT sentence multi-lingual Bidirectional Encoder Representations from Transformers

SSE sum of squared errors

USE Universal Sentence Encoder

ViTs Vision Transformers

WCSS within-cluster sum of square

TF-IDF Term Frequency-Inverse Document Frequency

Chapter 1

Introduction

The internet has seamlessly integrated into our daily lives, functioning as a crucial resource for both leisure and professional activities. Whether connecting with friends, shopping, searching for travel deals, or using online platforms to enhance professional skills, its versatility offers us numerous benefits [1]. However, many people find themselves sidetracked while browsing the internet, stumbling upon content unrelated to their original intent. This distraction is common, as the vast expanse of information from numerous sources can easily divert users, resulting in uncertainty and deviation from their initial objectives [2]. Acknowledging this challenge, researchers are increasingly focusing on recommender systems to provide personalized services or items amid the overwhelming abundance of information.

Recommendation Systems (RSs) are information filtering systems that recommend appropriate items to users [3]. For example, a recommender system can suggest which product to buy in e-commerce, which movie to watch on Netflix, and which video to watch on YouTube. The main goal of recommender systems is the ability to adapt the suggestions provided to the needs and interests of users. Identifying and recognizing the needs and tastes of users can be predicted based on the knowledge obtained from them [1]. Developing successful recommender systems RSs has become a crucial business objective for many companies in the media and entertainment sectors, as these systems enhance user experience, increase customer satisfaction, and drive revenue growth [4]. However, making these systems more intelligent presents challenges, including data sparsity, privacy concerns, and cold start problems. This research focuses on the cold start problem, which occurs when RSs struggle to suggest content to new users [4].

1.1 Problem Statement

One of the major challenges in recommendation systems is the cold start problem, as highlighted in several studies [5, 6, 7]. The cold start problem occurs when a recommendation system lacks sufficient data to make accurate recommendations. This can happen in three scenarios: user cold start, item cold start, and system cold start. A user cold start occurs when a new client joins the system. An item cold start happens when a new item is introduced to the system. A system cold start takes place when the system itself is new and lacks sufficient information to make recommendations.

In the context of online advertising, the user cold start problem refers to the difficulty of delivering relevant advertisements to users navigating the platform for the first time, meaning they have no prior interactions with Ads on the platform. To address the user cold start problem, common techniques involve using item attributes related to the user profile. However, this approach often requires the user's information.

This research aims to address the user cold start problem in Amharic YouTube Ads for new users by answering the following research questions:

RQ1 What is the effectiveness of a language model in recommending semantically related Amharic Ads to YouTube videos?

RQ2 What is the effectiveness of the proposed model in mitigating user cold start problems?

1.2 Objectives

1.2.1 General Objective

To address the cold start problem for new users in content-based recommender systems for Amharic YouTube Ads using the BERT Language Model.

1.2.2 Specific Objectives

The following specific objectives are carried out to achieve the general objective of this study:

- To collect datasets of Amharic YouTube video titles, descriptions, and advertisement titles.
- To select a semantic aware algorithm for generating Amharic word embedding.
- To modify the semantic model for recommendation systems.
- To evaluate the performance of the proposed model on cold start users.

1.3 Contribution

In this work, two main contributions are made.

- Our primary contribution involves the collection, annotation, and preprocessing of two comprehensive datasets. The dataset is divided into two sections: YouTube and advertisements. The YouTube section includes attributes such as video IDs, titles, descriptions, and categories, while the advertisement section contains ad URLs and titles. Additionally, we have preprocessed both sections to enhance their Amharic language features. These datasets are available here https://github.com/firekebede/YouTube-advertisement-recommendations-using-BERT.
- We have put forth an approach that utilizes language models to enhance the understanding and representation of Amharic text data, to address the user cold start problem in YouTube advertisement recommendation systems for Amharic videos.

1.4 Scope and Limitation

This research focuses on developing a recommendation system for Amharic language YouTube Ads specifically targeted at Amharic YouTube videos. By integrating algorithms designed to handle the unique aspects of the Amharic language, the study seeks to enhance the effectiveness of Ads recommendations in Amharic YouTube Videos, ensuring they are more relevant, related content-wise, and engaging for the target audience. The research addresses the cold start problem, which occurs when a new or unsigned user watches a video on the platform. The goal is to create a more efficient and personalized advertising experience for Amharic video viewers on YouTube without seeking lots of information from them. Limitations of this study include annotations made by users that numbers and punctuation were removed to optimize the model, potentially impacting the completeness and meaning of the sentences. Despite these limitations, the study's solutions provide a step ahead for the advertisement recommendation system.

1.5 Research Methodology

To achieve the research objectives, the following procedures were followed:

I. **Literature Review:** Lots of research literature on cold start problems in recommendation have been reviewed to formulate the problem, identify the gaps, and propose an approach.

II. **Proposed an Approach:** After defining a precise problem, a potential solution approach is proposed.

III. **Data collection and Preparation:** Data are collected from the YouTube platform relevant to the proposed solution and labeled for evaluation.

IV. **Verify Proposed Approach:** Finally, the proposed approach is evaluated using the labeled dataset.

1.6 Organization of the study

The remaining sections of this thesis are organized as follows: Chapter 2 presents fundamental concepts related to RSs theoretical backgrounds, online advertisement, YouTube, Amharic languages, and related works. In Chapter 3 the proposed approach is presented. Experiments to evaluate the proposed approach and discussions on results are presented in Chapter 4, and Chapter 5 discusses the conclusions and identifies the gaps that can be researched in the future.

Chapter 2

Background

This chapter offers a thorough exploration of recommendation systems (RSs), digging into their various categories and the challenges they face. It provides a detailed analysis of advertising and YouTube advertisement strategies, offering valuable insights into their workings. Additionally, it introduces the context of the Amharic language and highlights the challenges encountered in dealing with low-resource languages. The last part includes a brief discussion of related works.

2.1 Recommendation Systems

RSs are information filtering systems designed to suggest relevant items to users. These systems analyze user preferences, behaviors, and other data to provide personalized recommendations to enhance the user experience by helping them discover content that aligns with their interests [3]. It is an effective tool for overcoming choice overload by recommending items that align with users' interests [5]. RSs are vital tools across many industries, such as e-commerce, finance, media, communications, and entertainment. They analyze customer preferences and behaviors to offer personalized suggestions that match individual needs [8]. These systems help businesses better understand customers and deliver more relevant and appealing recommendations. The main objectives are to enhance customer satisfaction and foster loyalty to boost overall revenue. RSs ensure customers find what they are looking for and make them more likely to return for future interactions.

2.1.1 Recommendation system approaches

Since the early development of recommendation systems, numerous approaches have been proposed [5],[9]. These methods can be categorized into several classes based on shared characteristics. Some of the most common classifications of recommendation approaches include collaborative filtering, content-based filtering, and hybrid methods. Each class utilizes distinct techniques to analyze user data and predict preferences, contributing to the effectiveness and diversity of recommendation systems.

2.1.1.1 Collaborative filtering

Collaborative Filtering (CF) is a filtering algorithm, widely used across various platforms. Its concept mirrors how we often make decisions in our daily lives. Just like seeking recommendations from friends or relying on reviews before trying something new, CF taps into the opinions and experiences of other users stored within the system. Here users provide ratings for items like products, movies, music, books, and more. As these ratings accumulate, the system gathers valuable information. It utilizes this data to identify users with similar tastes or preferences. Subsequently, by leveraging these similarities and the preferences of like-minded users, the system recommends items to each user.

For instance, if you enjoy a particular type of book and other users who share your taste also like a certain novel, the system might recommend that novel to you. It takes into account both the popularity of items among other users and how closely your preferences align with theirs. Some well-known examples of successful implementation of this approach include Amazon.com and grouplens.org. These platforms leverage collaborative filtering to enhance user experiences by offering personalized recommendations tailored to individual tastes and preferences [1].

One key advantage of collaborative filtering is its ability to generate personalized recommendations without the need for content-based analysis. By relying on the behavior and preferences of users, this method can recommend items that are likely to be of interest based on similar users' choices. This approach can capture complex and nuanced user preferences that are difficult to model using explicit features of items, often resulting in highly accurate recommendations. Additionally, as the system gathers more user data, it can improve over time, becoming more adept at predicting user preferences.

A significant challenge arise in CF is the cold start problem, where the system struggles to provide recommendations for new users or items due to a lack of sufficient data. Without historical interactions, the system cannot make meaningful suggestions. Furthermore, collaborative filtering can suffer from scalability issues, as the computational requirements increase with the number of users and items. This can lead to performance bottlenecks in large-scale systems. Another limitation is the potential for popularity bias, where the system tends to favor well-known items over niche ones, potentially reducing diversity in recommendations.

2.1.1.2 Content based filtering

Content-based filtering generates recommendations based on similarities of new items to those that the user liked in the past by exploiting the descriptive characteristics of items.[1] For example, if a user has shown interest in a particular type of movie, the system will recommend movies that fall into that category. The content of each item is represented as a set of descriptors or terms that are inherent to the item.

Content Based Filtering (CBF) suggests items to users based on the characteristics or features of items they have liked or interacted with in the past. The fundamental idea is that if a user has enjoyed certain items before, they are likely to enjoy similar items in the future. Each item—be it a movie, book, or product—is represented by a set of attributes that describe its content. A user's preferences are captured by creating a profile based on the attributes of items they have previously liked or interacted with. When generating recommendations, the system compares the features of previously liked items with those of new, unseen items. It then ranks these new items based on their similarity to the user's established preferences and recommends the top-ranked items.

The advantages of content-based filtering include personalization, explainability, and its ability to recommend new items or users without needing extensive historical data. Personalization ensures that recommendations are tailored to individual users' tastes. The system's recommendations are explainable because they are based on clear attributes of the items. Additionally, content-based filtering can effectively recommend new items to users by matching item attributes with user profiles.

However, content-based filtering has limitations. One significant issue is overspecialization, where users may only receive recommendations for items very similar to their past preferences, potentially limiting exposure to new and diverse content. To mitigate these limitations, content-based filtering is often combined with other recommendation techniques, such as collaborative filtering or hybrid approaches. These combinations create more robust and effective recommendation systems by integrating various methods to enhance user experience and recommendation accuracy.

2.1.1.3 Hybrid approach

A hybrid recommendation approach combines multiple recommendation techniques to improve the accuracy and effectiveness of the recommendations. This approach can integrate collaborative filtering, content-based filtering, and other methods like demographic filtering or knowledge-based systems. Using the strengths of different methods, a hybrid approach can overcome the limitations of individual techniques, such as the cold-start problem in collaborative filtering or the limited scope of content-based filtering. For example, a hybrid system might use collaborative filtering to identify similar users and their preferences, and then refine these recommendations using content-based filtering based on the specific attributes of items that the user has shown interest in. This leads to more personalized and relevant recommendations, enhancing user satisfaction and engagement[1]. Table 2.1 provides a general summary of these approaches with examples.

Approach	Description	Mechanism	Example
Content-Based Filtering (CBF)	Recommends items based on their attributes and the user's profile.	Analyzes item attributes and user profiles to find items similar to user preferences.	A news website recommending articles based on a user's reading history.
Collaborative Filtering (CF)	Recommends items based on similar users' preferences and interactions.	Uses user-item interactions to find patterns and similarities among users.	An online retailer recommending products based on purchases by users with similar buying patterns.
Hybrid Systems	combine multiple techniques for improved accuracy.	Integrates different recommendation approaches to leverage the strengths of each method.	A streaming service recommending shows based on user viewing history and trending content.

Table 2.1: Summary of recommendation system approaches

2.1.2 Goal of Recommendation Systems

The main goal of recommender systems is the ability to adapt the suggestions provided to the needs and interests of users. Identifying and recognizing the needs and tastes of users can be predicted based on the knowledge obtained from them [1]. There are many use cases of RSs, like personalized content which produces dynamic recommendations for various audience types, which helps to enhance the on-site experience. It also improves product search since it assists in classifying the product according to its features.

RSs are crucial in the entertainment and media industries, where the abundance of available movies, TV shows, and other content can overwhelm users. By analyzing user preferences and behaviors, these systems provide personalized suggestions that help customers navigate the vast digital landscape, discover content they'll enjoy, and save time. Consequently, having a successful recommendation system has become a pivotal business objective for many companies in the media and entertainment sectors, as it enhances the user experience, increases customer satisfaction, and drives revenue growth [4]. RSs are widely used in video streaming platforms like YouTube 2.3 in this thesis, where recommending relevant videos or advertisements to the right viewers is important.

2.1.3 Challenges in recommendation systems

The rapid expansion of recommendation engines demonstrates that businesses worldwide are eager to harness this technology's potential. However, leveraging these powerful tools effectively poses significant challenges that must be navigated carefully [10].

- **Data Sparsity:** The lack of interaction between users and items, resulting in a small number of ratings or interactions, makes it challenging to generate recommendations that impacts the system due to a general lack of interaction data [6].
- **Cold Start Problem:** Insufficient data for new users or items makes it difficult to provide accurate recommendations. This happens when a new user joins the system or new items are added to the catalog. In these situations, the recommendation algorithm struggles to predict the preferences of the new user or the ratings for the new items due to a lack of historical interaction data. This lack of information makes it difficult to provide accurate recommendations initially.
- **Privacy:** There is a always a trade off between accurate recommendation and user [11]. The more the algorithm knows about a customer, the better it can make accurate recommendations. However, many customers are reluctant to share their personal information, especially after several high-profile data leaks in recent years. Without this data, the recommendation engine can't perform as effectively as expected.

2.2 Advertisement

Encouraging people to purchase specific goods or services and fostering brand loyalty are crucial aspects of marketing. The emergence of a wide variety of products and services has intensified competition among producers. Advertising plays a pivotal role in marketing by persuading consumers to make purchases. Advertisement is a communication medium between the producers and consumers that catches the eyes and makes them buy the product. With rapid technological advancements, new and innovative products are continually being introduced to the market. This has amplified the need for effective advertising, especially in the post-industrial era, where the competition is fiercer and the market is more saturated. Consequently, businesses must leverage advertising more strategically to capture consumer attention and drive sales [12]. There are many ways of advertising products, and this paper focuses on digital advertisement.

2.2.1 Digital advertisement

Digital advertisement [13], as a vast market, has gained significant attention in various platforms ranging from search engines, third-party websites, social media, and mobile apps. This kind of advertising differs significantly from traditional advertising. It allows marketers to target their consumers with greater precision and care. Examples of online advertising include contextual advertisements on search engine results pages, banner Ads, blogging, social media network advertisements, flash Ads, and online classified advertising. This targeted approach helps marketers reach specific audiences, enhancing the effectiveness of their advertising campaigns and increasing the likelihood of engagement and conversions. The prosperity of online campaigns is a challenge in online marketing and is usually evaluated by user response through different metrics, such as clicks on advertisement (ad) creatives, subscriptions to products, purchases of items, or explicit user feedback through online surveys. Recent years have witnessed a significant increase in the number of studies using computational approaches, including machine learning methods, for user response prediction.

2.3 YouTube

YouTube is a video-sharing platform where users can watch, like, share, comment on, and upload videos [14]. Accessible on PCs, laptops, tablets, and mobile phones, YouTube offers several main functions: users can search for and watch videos, create personal YouTube channels, upload their own videos, and interact with other content through likes, comments, and shares. Additionally, users can subscribe to or follow other YouTube channels and users, as well as create playlists to organize and group videos.

As a free-to-use service, YouTube provides a valuable space for teenagers to explore their interests. Many young people use YouTube to watch music videos, comedy shows, instructional guides, recipes, and various life hacks. It also serves as a platform for teens to follow their favorite vloggers, subscribe to other YouTubers, and keep up with celebrities they admire. This makes YouTube not only a source of entertainment but also a tool for learning and staying connected with popular culture.

2.3.1 YouTube Advertisement

YouTube advertising is a powerful tool for businesses to reach a wide audience and promote their products or services [15]. It works by allowing advertisers to create video Ads that are displayed to users on the YouTube platform. There are several types of YouTube Ads, each with its unique features and benefits [16].

- **Skippable in-stream Ads:** These Ads play before, during, or after a video, and users can skip them after 5 seconds. Advertisers only pay if a viewer watches more than 30 seconds of the ad or interacts with it. These Ads are effective for increasing brand awareness, but it's important to hook viewers in the first 5 seconds to keep their attention
- **Non-skippable in-stream Ads:** Similar to skippable in-stream Ads, these Ads play before or during videos, but users cannot skip them. They can last up to 15 seconds, and advertisers pay per 1000 views.
- **Bumper Ads:** These 6-second or shorter Ads play before or during a video and cannot be skipped. Advertisers pay based on the number of views. Bumper Ads are effective for transmitting short advertising messages and grabbing viewers' attention to generate interest in the brand
- **Discovery Ads:** appear in YouTube search results and on the right sidebar of watch pages. They are in text form with a picture that leads to the promoted video after clicking. Advertisers pay when viewers click the ad and start watching. Discovery Ads are effective for promoting brand videos on the platform, as users are already interested in the topic

2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a cutting-edge machine learning technology that empowers computers to understand, interpret, and manipulate human language with remarkable accuracy. In today's data-driven world, organizations are flooded with massive volumes of voice and text data, captured from countless communication channels like emails, text messages, social media feeds, videos, audio, and more [17]. NLP software has become a game-changer, enabling these organizations to automatically process this data, analyze the underlying intent or sentiment, and respond to human communication in real-time, at an unprecedented scale.

NLP has unlocked new frontiers in customer service, sales, marketing, and beyond. Imagine a world where computers can comprehend the logic and complexities of human language, just like another person would. NLP is making this vision a reality, transforming the way organizations interact with their audiences, deliver personalized experiences, and drive business growth. It's a technology that is redefining the boundaries of human-machine collaboration, paving the way for a future where perfect communication and understanding between humans and computers is the norm [18].

2.4.1 Low resource language

Low-resource languages refer to languages that have limited digital data and resources available, such as annotated corpora, dictionaries, and language models. Developing high-performing NLP systems like in this thesis recommendation systems. These languages pose unique challenges compared to well-resourced languages like English [19].

Amharic, a low-resource language, faces significant challenges in language model performance due to the limited availability of training data and the complexity of the language itself. This scarcity of data can lead to reduced model performance and accuracy, making it difficult to fine-tune language models for tasks such as machine translation and recommendation. The datasets used in this thesis are texts collected from Amharic YouTube videos and Ads that contribute to solving the issue stated above.

2.4.2 Amharic Language

Amharic, the second most widely spoken Semitic language after Arabic, is a vibrant tongue with a unique writing system and deep cultural significance. Written from left to right using the Fidel (ፊደል) script, Amharic employs a syllable-based system where consonants and vowels coexist within each graphic symbol. This distinctive feature contributes to its complexity and beauty [20].

The 'Fidel' (ፊደል) script consists of 34 core characters, each having seven variations to represent different vowels. In addition, Amharic includes 51 labeled characters, 20 numeric symbols, and 8 punctuation marks, totaling over 310 unique characters. This extensive character set reflects the language's morphological complexity and inflectional nature, making it a fascinating subject of study for linguists [21].

Amharic is the official working language of the federal government of Ethiopia and many regional states, underscoring its importance in the country's administration and daily life [22]. It is used in government documents, media, education, and literature, playing a crucial role in uniting Ethiopia's diverse population.

Research has shown that Amharic's linguistic structure, with its complex verb conjugations and noun inflections, poses unique challenges and opportunities for language processing and computational linguistics [23]. Studies have explored the implications of its syllabic writing system on literacy and education, highlighting both the benefits and difficulties faced by learners [24]. Amharic's rich oral tradition and literary heritage, including ancient texts and modern literature, offer a window into Ethiopia's history, culture, and societal values. The language's adaptability and resilience have allowed it to thrive in the digital age, with increasing efforts to develop Amharic language technology, including keyboards, fonts, and translation tools.

2.5 Related Works

Numerous research efforts have been undertaken to address the cold start problem in recommendation systems. In this section, various papers that have deployed unique methods to tackle the cols start issue are reviewed. Research papers that utilize recent methods like deep learning and semantic embedding-based methods are discussed.

2.5.1 Deep learning based recommendation systems

Yinwei Wei et al. [25] presented an approach to address the item cold-start problem in recommendation systems using contrastive learning, which is a type of self-supervised learning where the model learns to distinguish between similar and dissimilar pairs of data points. The researchers reformulate cold-start item representation learning from an information-theoretic standpoint, aiming to maximize the mutual dependencies between item content and collaborative signals. Extensive experiments conducted on four publicly accessible datasets demonstrate that the proposed Contrastive Learning for Cold-start Recommendation (CLCRec) method significantly outperforms state-of-the-art approaches in both warm- and cold-start scenarios. This research effectively leverages item content to mitigate the challenges associated with the item cold-start problem that arises in collaborative filtering.

Feng et el. [26] proposes a hybrid deep learning-based strategy that aims to enrich user and item profiles to improve recommendations in user cold-start scenarios. They highlight the importance of hybrid approaches in addressing the new user cold start problem in recommendation systems. By combining collaborative filtering with content-based and deep learning techniques, they use Probabilistic Matrix Factorization (PMF) with Bayesian Personalized Ranking (BPR). The researchers have developed better solutions to enhance the performance and personalization of recommendations, especially for users with limited interaction history.

Volkovs et el. [27] presented DropoutNet which is a novel approach by leveraging deep learning techniques, specifically dropout neural networks, to address the cold start problem in CF. DropoutNet innovatively integrates dropout mechanisms during the training phase of neural networks. Dropout is a regularization technique where randomly selected neurons are ignored (dropped out) during training. This prevents the network from becoming overly reliant on specific neurons, thus promoting generalization. In the context of DropoutNet, applying dropout simulates the cold start conditions by artificially creating sparse data scenarios. This trains the network to be robust against such sparsity, enabling it to handle real-world cold start situations effectively. The method enhances the system's ability to predict user preferences and item relevance even with minimal initial data. The paper demonstrates that DropoutNet outperforms traditional methods in both user and item cold start cases.

Tuval et el. [7] present an approach to address the cold start issue in recommender systems. The core innovation of this research lies in the use of hypercube graphs, which efficiently represent the relationships between different content attributes. By mapping user preferences onto these graphs, the system can make recommendations even with minimal user ratings. This approach significantly enhances privacy since it reduces the need for extensive data sharing. Their approach leverages hypercube graphs for user modeling where users and items are represented as vertices in a hypercube structure, and edges between them indicate interactions, such as ratings. This graph-based structure allows the system to infer user preferences with minimal input data. Plus they utilize graph-based learning techniques, including Graph Neural Networks (GNNs) and graph embedding methods.

The authors validate their method through extensive experiments involving over 1,000 users in domains like restaurants and movies. They demonstrate that their model not only outperforms standard machine learning algorithms when dealing with fewer ratings but also remains competitive with larger datasets. Their approach simplifies the training process and requires less computational effort, making it an attractive solution for practical applications. The study shows that with as few as ten ratings, the system can make reliable recommendations, thus effectively addressing the cold start problem.

Choi et al. [28] presented a framework that utilizes weak supervision signals, such as item metadata and user interactions, to generate pseudo-labels for new items. Through a rigorous evaluation process, the authors demonstrate the efficacy of their approach in alleviating item-side cold-start problems by effectively incorporating weak supervision signals into the recommendation process. The paper offers valuable insights into the potential of weak supervision techniques in enhancing the robustness and scalability of recommender systems.

2.5.2 Semantic-based recommendation systems

Jeevamol and Renumol [29] show how their system leverages domain-specific ontologies which can define the relationships between concepts within a particular domain of knowledge to enhance recommendation accuracy and solve the new user cold-start problem. Providing a detailed exposition of the system architecture, algorithms, and evaluation methodologies, the authors offer valuable insights into the potential effectiveness of their approach. The paper discussed the importance of ontology-based modeling in capturing the rich semantics inherent in e-learning content, thereby enabling more personalized and contextually relevant recommendations for students.

Yadav et al. [30] uses knowledge graph embedding techniques. The authors propose a method that leverages link prediction in a knowledge graph to develop a semantic-based recommender system. They apply graph embedding techniques to extract the semantics of explainable recommendations, which provides better semantic explanations for the recommended items. The proposed method is validated using the MovieLens dataset, and the results show that factorization-based scoring functions provide better semantic recommendations. The paper contributes to the field of recommender systems by addressing the problems of cold-start and sparsity. The use of knowledge graph embedding techniques allows for more effective feature learning and improves the overall performance of the recommender system. Additionally, the paper highlights the importance of semantic explanations in recommender systems, which can enhance user trust and satisfaction.

Misztal-Radecka et al. [31] presents an approach to mitigate the cold-start problem in recommendation systems, where insufficient data about new users or items challenges accurate recommendations. The Meta-User2Vec model, an extension of the User2Vec model derived from Doc2Vec, applies unsupervised learning principles to create embeddings for users and items using metadata labels, enabling recommendations without extensive historical data. Their model integrates user and item metadata to form comprehensive representations, combining user profiles (interaction history and metadata) and item descriptions to map them into a shared embedding space. This facilitates accurate predictions even without prior interaction data. Emphasizing ”extreme cold-start” scenarios where both user and item histories are unknown, the model bridges user and item metadata to make recommendations when traditional methods fail due to data scarcity. Comparing the Meta-User2Vec model with approaches like embedded collaborative filtering and Context2Vec, the paper highlights its superiority in handling complete cold-start situations. Unlike methods that concatenate metadata with item descriptions, Meta-User2Vec uses metadata as a separate input, leading to enhanced performance.

Evaluations using datasets such as MovieLens and Deskdrop demonstrate that the Meta-User2Vec method outperforms other techniques in cold-start settings, particularly when user and item metadata are available, showcasing its practical applicability in real-world scenarios with continuously emerging new users and items.

Table 2.2 summarises the paper above with respect to the methodology implemented, dataset us and, the cold start addressed.

Author(s)	**Methodology**	**Dataset**	**Addressed Cold Start**
Yinwei Wei et al. [25]	Contrastive Learning	MovieLens, Yelp, Amazon	Item
Volkovs et al. [27]	DropoutNet	CiteULike and ACM RecSys 2017	User, Item
Jeevamol and Renumol [29]	Ontology-based Modeling	Learning Object Metadata	User
Choi et al. [28]	Weak Supervision Signals	MovieLens	Item
Yadav et al. [30]	Knowledge Graph	MovieLens	Item
Misztal-Radecka et al. [31]	Meta-User2Vec	MovieLens, Deskdrop	User, Item

Table 2.2: Summary of research approaches to address the cold-start problem in recommendation systems

The related works discussed above tires to solve the user or item cold start problems in recommendation systems using user or user's neighbor rating as a resource in their dataset. The papers use semantic-aware models to analayis items attribute and to solve item cold start. In this study, we have tried to solve the user cold start problem that arises in Amharic advertisement recommendation systems, for a specific case of the YouTube platform. Using only an Amharic YouTube video title and description with an Amharic advertisement video title for solving user cold start problems that arise in the platform for unsigned users.

Chapter 3

Methodology

In this chapter, the implemented methodology for addressing the user cold start problem in Ads recommendation of YouTube for Amharic language videos is discussed. Data collection, preprocessing, embedding language models, and similarity measurement methods are discussed.

3.1 Proposed Method

The general architecture of the proposed approach is shown in Figure 3.2. This architecture consists of three core modules.

- Data Collection and Processing
- Sentence Embedding
- Similarity Measurement

YouTube videos and advertisement textual content (title and descriptions) are collected via the YouTube API. This data is preprocessed and then fed into embedding language models. Subsequently, a similarity measurement is made and top k Ads recommendations are generated for each YouTube video. Detailed information about these three processes is provided in the following sub-sections.

3.2 Data Collection and Processing

The experiment carried out in this thesis made use of two datasets. The first dataset, YouTube video titles and descriptions contains the textual context of Amharic YouTube videos collected via YouTube API. The second dataset contains Ads URL and title which contains the textual information of the advertisements. Ads datasets are collected from advertisement production companies' YouTube pages.

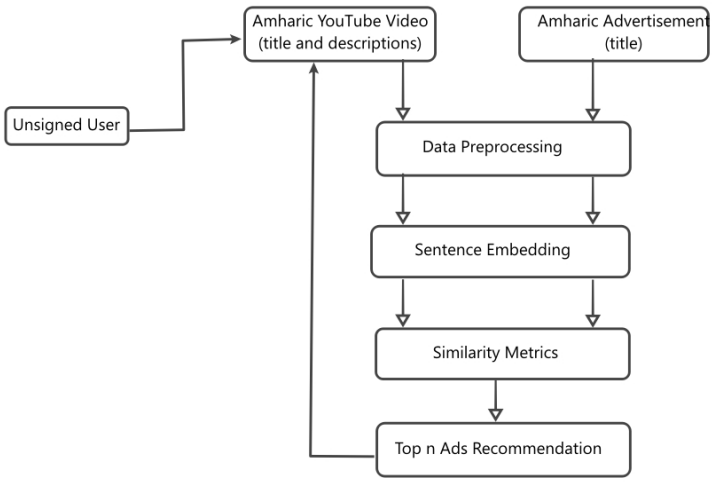

Figure 3.1: Proposed Methodology

3.2.1 YouTube API

The YouTube Application Programming Interface (API) is a powerful set of tools and protocols provided by YouTube, enabling developers to access and interact with YouTube's vast data and services programmatically. This API serves as a gateway, allowing developers to seamlessly integrate YouTube's functionalities into their applications and websites [32].

Using the YouTube API, developers can unlock a wide range of capabilities, from retrieving video metadata and channel information to managing video uploads, playlists, and user interactions. This API empowers developers to create innovative, YouTube-integrated experiences that enhance user engagement, content discovery, and content management [33].

In this thesis, YouTube API was used to collect YouTube video's Video IDs, titles, descriptions, and categories using ten (11) keywords. These key words are **አንቂንግር** (Motivational Speeches), **ምግብ** (Food), **ስፖርት** (Sport), **አካል ብቃት** (Physical Exercise), **አርት** (Art), **ቴክኖሎጂ** (Technology), **ልጆች** (Kids), **ኮንስትራክሽን** (Construction),**መዝናኛ** (Entertainment), **ጤና** (Health), **ዉበት** (Beauty). About 5500 video titles and descriptions were collected which is 500 from each category as shown in figure 3.2 to make a fair distribution over the data.

The advertisement data were collected using the YouTube API to scrape content from various YouTube channels operated by advertising and promotion companies. This approach ensured that a comprehensive and diverse set of advertisements were collected, encompassing multiple industries and promotional styles. By focusing on channels specifically managed by advertising firms, we ensured the inclusion of high-quality, professional advertisement videos. This dataset forms a robust foundation for our analysis, enabling the development of Advertisement recommendation algorithms that accurately reflect current trends and practices in digital advertising.

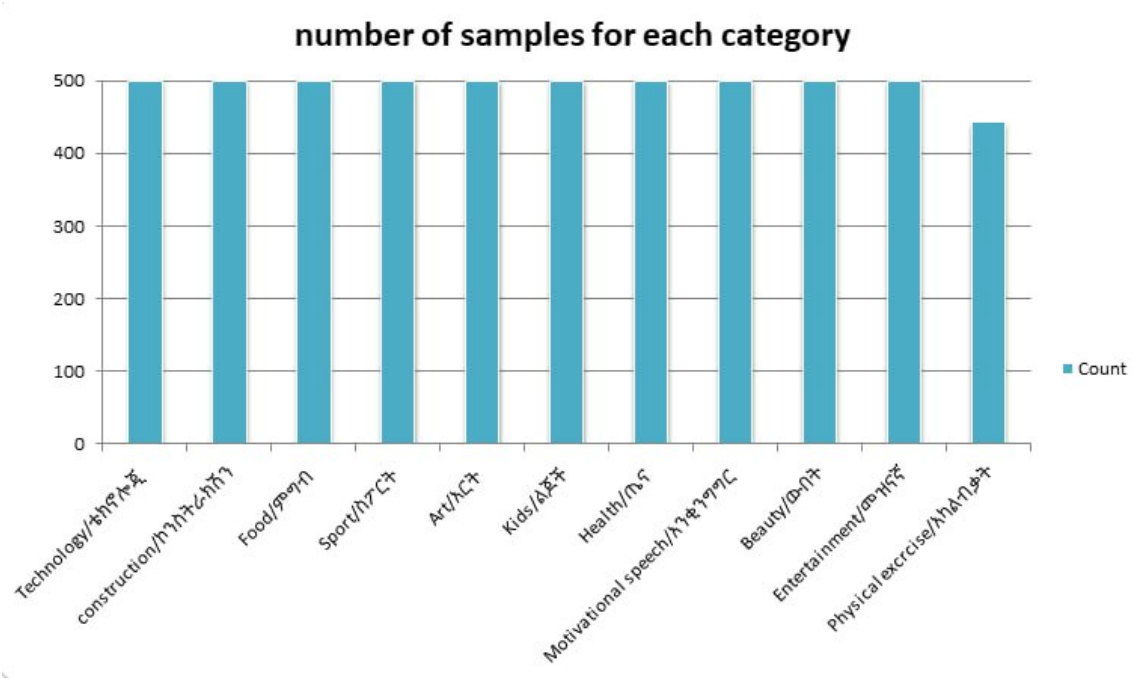

Figure 3.2: YouTube videos data distribution in each category

3.2.2 Data cleaning

Cleaning and filtering datasets are critical steps in the development of recommendation systems. Ensuring that the data is high-quality and relevant significantly enhances the accuracy and personalization of recommendations for users. This process involves removing duplicates, handling missing values, and eliminating irrelevant information, which collectively improve the dataset's integrity. By focusing on clean and well-filtered data, recommendation algorithms can better understand user preferences and behaviors, leading to more precise and tailored recommendations that enhance user satisfaction and engagement.

Columns in the datasets that contained missing values were removed since incomplete data can significantly impair the model's ability to learn meaningful patterns from the context. Columns containing video titles with alphabetic characters would not provide useful information for the recommendation task. These types of video titles are likely irrelevant or even misleading for the model. Therefore strategic decisions were made to remove these columns from the dataset entirely.

3.2.3 Text Preprocessing

Once data are filtered, the texts have to be cleaned and prepared for the model. Different approaches used for text processing are discussed below.

- **Normalization**

  Amharic is written in Geez alphabets called Fidel or (ፊደል). Some of these characters (Fidels) can be written in different ways, even though they sound the same. for example, the "Fidel" ሀ (ha) can have more than four representations (such as ሃ, ሐ, ሓ, ኀ, ኃ, and so on). This was more important in the past, as different fidels could change the meaning. But nowadays, people often write these characters however they want, especially online on social media including YouTube. This causes problems for NLP because the same word might have different representations. For example, the word for "eye" could be written as (ዐይን) or ((አይን)) but mean the same thing [20]. Therefore normalization was used to process the data to achieve a better recommendation system over language.

- **Unwanted feature removal**

  YouTube videos consistently feature both titles and descriptions. The title provides a concise summary of the video's content, helping viewers understand what to expect. The description offers additional details, which may include a more in-depth summary, relevant links, timestamps, and other pertinent information. These elements are crucial for both viewer engagement and the platform's search and recommendation algorithms, as they help categorize and surface videos to the appropriate audience.

  Emojis and Hashtags are used to make YouTube titles attractive to viewers. Such kinds of symbols were removed for better performance of the model. Numerical features in video titles and descriptions were also removed.

Unnamed: 0	Video Id	Title	Description	Category
4208	rqjK9K-9V64	ጤናማ ያልሆነን የብልት ፈሳሽ እንዴት እንለያለን	ጤናማ ያልሆነ የብልት ፈሳሽ እንዴት እንለያለን	ጤና/health
4210	TZtviPkXPU0	ዳኑ የአጥንት ህክምና ማእከል የምርቃት በአል	ዳኑ የአጥንት ህክምና ማእከል የምርቃት በአል	ጤና/health
4211	4fwk3oZxUJc	ለጉልበት ህመምም ፍቱን መፍትሄዎች	መላ በማድረግ ደንበኞችን ይሁኑ ለማወቅ የሚፈልጉት ርእስ	ጤና/health
4212	e0nD9GKpnWU	ወደ ህክምና መደድ አለብን	ጤና ይስጥልኝ ውድ የቻናሌ ቤተሰቦች ሰላማችሁ በያላችሁበት ይሆን እየተመኘሁ ሁሌም ይዤር በምመጣው ሀሳብ	ጤና/health
4218	UDUUZf3L9JY	በኢትዮጵያ ውስጥ ጥርስ ለማስነቀል ለማስተክል ለማስር ለማስሞላት አጠቃላይ ህክምና ለማግኘት ለጠየቃችሁት ጥያቄ	ሰላም ለእናንተ ይሁን በሚ ዘላለም ቦጋለ ይባላል የምኖረው በዳላስ ቴክሳስ ግዛት ውስጥ ሲሆን በአሁኑ	ጤና/health
4221	pNg3kDlxanA	ህክምና ክካድሪያዊ አሰራርና ስምሪት ነፃ ወጥድል ዶር ዮሀንስ ጫላ የአዲስ አበባ ጤና ቢሮ ሀላፊ ክፍል ሶስት	ህክምና ክካድሪያዊ አሰራርና ስምሪት ነፃ ወጥድል ዶር ዮሀንስ ጫላ የአዲስ አበባ ጤና ቢሮ ሀላፊ	ጤና/health
4223	KnuNJnZkHiY	አደገኛ አጥንት ነጂ የሆኑ የምግብ አይነቶችተጠንቀቁ	አደገኛ አጥንት ነጂ የሆኑ የምግብ አይነቶችተጠንቀቁ	ጤና/health
4226	Mah6rmLkex8	ህክምና የሚፈልጉ የእርግዝና አደገኛ ምልክቶች	በእርግዝና ወቅት ምን ምልክቶች አፋጣኝ የህክምና ክትትል ያስፈልጋቸዋል የደም መፍሰስ በእርግዝና ወቅት	ጤና/health
4227	j8x78BjXedw	የመጀመሪያ ደረጃ ህክምና	የመጀመሪያ እርዳታ የመጀመሪያ ህክምና እርዳታ አንድ ሰው አደጋ ከደረሰበት በኋላ ምን ልናደርግለት	ጤና/health
4228	wY4hr7GBED4	የ ብጉር ጠባሳ ህክምና	ብጉር ጥሎ የሚያልፋቸው ችግሮች አንዱ ጠባሳ ነው ይህንን ጠባሳ ለማከም የምንጠቀምበትን ህክምና በዚ ቪድዮ	ጤና/health

Figure 3.3: Sample example from the preprocessed YouTube videos dataset

3.2.4 Data annotation method

To evaluate the model, data were annotated through a carefully structured process. Samples were randomly selected from the entire YouTube video dataset and presented to users along with clustered advertisements. The task was to choose the advertisement that was most contextually related to a given video title and description. For this purpose Telegram chat platform was utilized to facilitate the process, creating a group of annotators and using Telegram polls to collect their choices. Detailed guidelines were provided within the Telegram group to ensure that annotators could perform their tasks effectively and consistently.

For each video title and description, a list of advertisement choices was provided. Since these Ads are around 500, it makes it difficult for the annotators to choose. Therefore clustering mechanisms were used to classify the Ads into related listed, section 4.2.1 discusses in brief the experiment conducted in clustering Ads data. Annotators were instructed to select only three Ads that they believed were the best match in context. The Ads that received the most votes were then chosen as recommendations for that specific video.

3.2.5 K-Means Clustering

K-means clustering is a widely used unsupervised machine learning algorithm that partitions data into K clusters based on their similarities. The algorithm works by iteratively reassigning each data point to the cluster with the closest mean until the points no longer change clusters [8]. The algorithm is particularly effective in identifying patterns and structures within large datasets, making it a valuable tool in various applications such as recommendation systems, data mining, and scientific paper analysis. Recent studies have explored the application of k-means clustering in various domains, including the analysis of scientific paper abstracts and the development of recommendation systems. This process is particularly useful for identifying patterns and structures in large datasets, such as identifying customer segments in marketing or grouping similar products in e-commerce [34].

The effectiveness of k-means clustering in various applications stems from its ability to identify dense regions in the data and group similar data points together. The algorithm is particularly useful in handling large datasets and can be combined with other techniques such as deep learning and natural language processing to improve its performance. Recent advancements in k-means clustering have also focused on improving its scalability and robustness, making it a versatile tool for a wide range of applications [35].

K-Means Clustering is an unsupervised learning algorithm used to partition data into K-distinct clusters. The algorithm operates through the following steps: it begins by initializing K centroids randomly and then assigns each data point to the nearest centroid. Subsequently, the centroids are recalculated based on the assigned points. These steps are repeated iteratively until the centroids stabilize. However, the algorithm has some drawbacks, including the need to specify the number of clusters (K) in advance, sensitivity to the initial placement of centroids, and the assumption that clusters have a spherical distribution. Despite these limitations, K-Means remains a popular and effective method for clustering tasks.

3.2.6 Elbow Methods (Clustering)

The Elbow Method is a widely adopted technique for determining the optimal number of clusters in k-means clustering [36]. This method involves plotting the within-cluster sum of squares (within-cluster sum of square (WCSS)) against the number of clusters (k) and identifying the ”elbow” point on the resulting graph. The WCSS measures the total variance within each cluster, and as the number of clusters increases, the WCSS decreases because smaller clusters are more homogeneous.

In practice, the Elbow Method works by first computing k-means clustering for different values of k, such as from 1 to 10. For each value of k, the WCSS is calculated, which represents the sum of the squared distances between each data point and the centroid of its assigned cluster. These WCSS values are then plotted against the corresponding number of clusters.

The key step in the Elbow Method is identifying the ”elbow” point on the plot. This point signifies a value of k after which the rate of decrease in WCSS slows down significantly. It is called the ”elbow” because the plot typically resembles an arm, and the optimal number of clusters is at the ”bend” or ”elbow” of the arm. The rationale behind this method is that while increasing the number of clusters will naturally reduce the WCSS, beyond a certain point, the marginal gain in reducing WCSS diminishes [37]. This ”elbow” point indicates that adding more clusters beyond this number yields minimal improvement in terms of variance reduction.

By using the Elbow Method, practitioners can effectively balance the complexity of the model with its performance. This ensures that the clustering solution is both efficient and effective, avoiding the pitfalls of overfitting or underfitting the data. The Elbow Method thus provides a practical approach to identifying the most appropriate number of clusters, enhancing the overall robustness and interpretability of the clustering results.

3.2.7 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical statistic used in text mining and information retrieval to reflect the importance of a word in a document relative to a collection of documents or corpus [38].

It combines two measures: Term Frequency (TF), which quantifies how often a word appears in a document, and Inverse Document Frequency (IDF), which downweights words that appear frequently across many documents, thereby highlighting more distinctive terms. By converting textual data into numerical vectors, TF-IDF enables the effective comparison and clustering of documents, making it a foundational technique for various applications such as search engines, recommendation systems, and natural language processing tasks. In this research this method have been used to cluster the Advertisement dataset and the result is comperes with the embedding models discussed in section 3.3.

3.3 Embedding Models

Embeddings for words or sentences provide a powerful method for representing language in natural language processing tasks. These embeddings are typically expressed as real-valued vectors within a vector space. In this space, semantically similar words or sentences are represented by vectors that are close to each other [39]. This proximity in the vector space reflects the similarity in meaning, making embeddings a crucial tool for understanding and processing language.

- **Word embeddings**: Word embeddings are numerical representations of words where words with similar meanings have similar vector representations. This technique, a cornerstone of natural language processing (NLP), encodes relationships between words by transforming them into numerical vectors. The dimensionality of these vectors is significantly smaller than the size of the vocabulary, ensuring efficient and meaningful representations. A prominent example of a word embedding model is Word2Vec, which effectively captures semantic relationships between words [40].

  Word2Vec: is a group of models that are used to produce word embeddings, which are dense vector representations of words in a continuous vector space. The main advantage of word2vec is its ability to capture semantic meanings and relationships between words based on their contextual usage in large text corpora [41].

  Word2vec comes in two primary architectures: Continuous Bag of Words (CBOW) and Skip-gram. The CBOW model predicts a target word based on its surrounding context words, while the Skip-gram model predicts the surrounding context words given a target word. These models use neural networks to learn word associations from a large corpus of text efficiently. The resulting word vectors can be used in various downstream tasks, such as word analogy solving, sentiment analysis, and machine translation [42].

  fast-text: Unlike traditional word embedding models like word2vec, FastText enhances its capability by considering subword information, making it particularly effective for handling rare and out-of-vocabulary words [43]. This approach involves breaking down words into character n-grams and generating word vectors from these subwords, resulting in more nuanced and flexible word embeddings. FastText can also be used for document embeddings by averaging the vectors of words within a document, offering a simple yet powerful method for capturing the semantic essence of the text. FastText is known for its speed and scalability, making it a popular choice for large-scale natural language processing tasks.

- **Document embeddings**: Research has extended the techniques used for creating word embeddings to develop new methods for representing longer texts, such as sentences or articles. These methods fall under the general term ”document embeddings.” A widely used document-embedding model is Doc2Vec [44] which is an extension of Word2Vec that learns document embeddings by training a model to predict words in a context, considering an additional document vector, which captures the meaning of entire documents in a single, comprehensive vector.

  Doc2Vec: is an extension of the word2vec model developed by Mikolov et al.[40] to create vector representations for larger blocks of text, such as sentences, paragraphs, or entire documents. Unlike word2vec, which generates embeddings for individual words, Doc2Vec generates embeddings that capture the semantic meaning of entire documents. It does this by introducing an additional vector representing the document, which is trained alongside word vectors.

  During training, the document vector is concatenated or averaged with word vectors to predict the next word in a context, allowing the model to learn a fixed-length feature representation for each document. This approach enables applications such as document classification, clustering, and information retrieval by providing meaningful numerical representations of text, capturing context and semantic relationships effectively [45].

- **Transformer-based embeddings**: Embeddings produced by models like BERT and GPT, leverage the self-attention mechanism to generate contextually rich representations of text. These models are pre-trained on vast corpora of text, enabling them to understand and capture complex linguistic features and dependencies [46]. BERT processes text bi-directionally, considering the context from both directions to produce highly nuanced embeddings as discussed in section 3.3.2, while Generative Pre-trained Transformer (GPT) generates embeddings through unidirectional processing. The resulting embeddings are highly effective for a range of NLP tasks, as they capture the complex relationships between words within their specific contexts [47].

Transformer-based embeddings were chosen for this approach because, for Amharic data, pre-trained BERT models achieve superior semantic embeddings compared to traditional methods such as word2vec or fastText [20]. Using the Natural Language Inference (NLI) dataset BERT has shown a better performance in understanding sentence semantic similarity.

3.3.1 Transformer

In traditional language tasks such as translation and recommendation, the sequence-to-sequence approach has been predominantly used, with recurrent neural networks (Recurrent Neural Networks (RNNs)) serving as the primary architecture [48]. In this method, each word in the sentence is first converted into a word vector, which is then processed by an encoder to generate a hidden state. This process is repeated for each subsequent word, with the hidden state from the previous word also fed into the encoder. For the first word, the hidden state is initialized to zero. Once the entire sentence is processed, the hidden state from the last word is passed to the decoder, which begins generating the translation. The decoder outputs the translation for the first word, and then uses this translated word and the updated hidden state as inputs to generate the second translated word.

A key challenge in this approach is ensuring that the hidden state effectively captures and retains the memory of all input words and their order, enabling accurate translation at each step. Attention mechanisms address this by allowing the decoder to focus on specific hidden states that correspond to the relevant words in the input sentence. For instance, if the second word in the input sentence is being translated, the attention mechanism ensures that the decoder pays particular attention to the hidden state generated by that word. This targeted focus improves the accuracy and coherence of the translation by ensuring that the decoder has precise information about the word currently being translated.

A Transformer is a deep learning model architecture that has significantly impacted natural language processing (NLP) and other fields since its introduction in 2017 by Vaswani et al [49] in the paper ”Attention is All You Need.” Unlike traditional recurrent neural networks (RNNs), Transformers use a self-attention mechanism to process sequential data, allowing the model to weigh the importance of different words in a sentence independently of their position. This mechanism enables the model to capture complex relationships and dependencies within the data. To address the lack of inherent sequence understanding, positional encoding is added to the input embeddings, providing the model with positional context.

The multi-head attention mechanism, another key feature, allows the model to focus on various parts of the input simultaneously, capturing different types of relationships. Each layer in the Transformer consists of a multi-head self-attention mechanism and a feed-forward neural network, enhanced with layer normalization and residual connections for improved training stability and performance. The architecture is divided into an encoder, which processes the input sequence into continuous representations, and a decoder, which generates the output sequence using these representations as context.

Transformers have found applications in numerous domains. In NLP, they are used for language translation, text summarization, question answering, and sentiment analysis. They have also been adapted for computer vision tasks, such as image classification and object detection, through models like Vision Transformers Vision Transformers (ViTs). Additionally, Transformers are utilized in speech processing for recognition and synthesis, and in recommender systems to enhance user experience by understanding preferences through natural language processing.

There are three distinct attention blocks as shown in Figure 3.4 that is one for the input sentence, one for the target sentence, and a third that integrates information from both. This third block, known as the multi-head attention block, facilitates the mapping between different languages in translation tasks by generating multiple attention vectors for each word and averaging them to capture intricate relationships. This multi-head approach enriches the model's ability to encode complex word relationships, making it highly effective in tasks like translation.

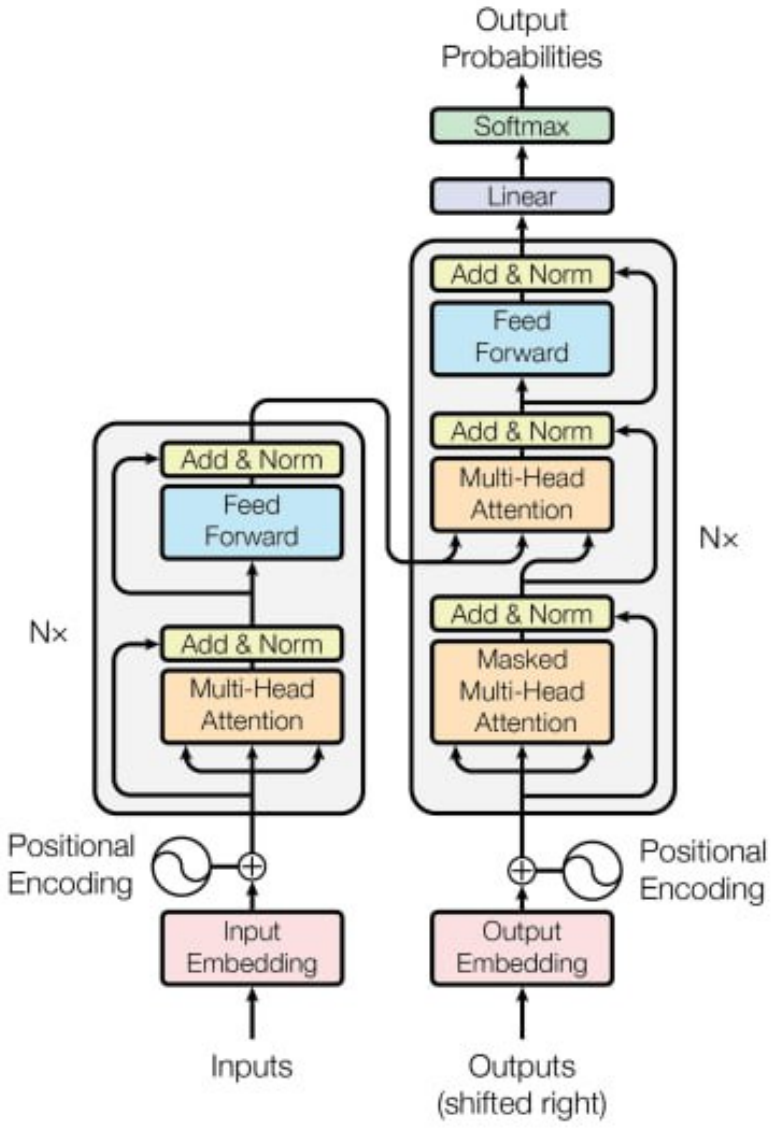

Figure 3.4: Transformer architecture [49]

3.3.2 BERT

BERT is a pre-trained language model developed by Google in 2018 [50]. It has been widely used in various NLP tasks due to its ability to improve performance on many tasks. BERT is a robust encoder for the transformer model that can understand the broader context and deliver state-of-the-art performance across various NLP tasks. It is pre-trained on a large corpus of text, which allows it to learn general language understanding and contextualized representations of words. This pre-training is done using a masked language modeling task and a next sentence prediction task [51]. BERT uses a multi-layer bidirectional transformer encoder. This architecture allows it to process input sequences in both forward and backward directions, which helps in capturing contextual relationships between words. It has been applied to various NLP tasks, including question answering, sentiment analysis, named entity recognition, and text classification. BERT can be fine-tuned for specific tasks, which allows it to adapt to different domains and tasks [52].

Historically, language models could only read input text sequentially either left-to-right or right-to-left as discussed in section 3.3.1 but couldn't do both at the same time. BERT is different because it's designed to read in both directions at once. The introduction of transformer models enabled this capability, which is known as bidirectionality. Using bidirectionality, BERT is pre-trained on two different but related NLP tasks; these are masked language modeling Masked Language Modeling (MLM) and next sentence prediction Next Sentence Prediction (NSP).

The paper [50] introduces two sizes of BERT models. BERT Base and BERT Large, with the latter achieving state-of-the-art results. BERT, essentially a stack of Transformers, differs in the number of encoder blocks it contains. BERT Base has 12 encoder blocks, each with a feed-forward network of 768 nodes and 12 attention heads. In contrast, BERT Large has 24 encoder blocks, each with a feed-forward network of 1024 nodes and 16 attention heads. For comparison, the original Transformer model has 6 encoder layers, 512 hidden units, and 8 attention heads. Similar to the Transformer, BERT processes an input sequence of words, applying self-attention in each layer to understand the relationships between words and passing the outputs to the next layer.

BERT utilizes a sophisticated type of word embedding called Embeddings from Language Models (ELMo) [53]. Unlike traditional embeddings, ELMo considers the entire sentence before assigning an embedding to a word, capturing valuable contextual information. This is crucial for understanding words with multiple meanings, like ”right,” which varies depending on context. ELMo achieves this by using a bi-directional Long Short-Term Memory (LSTM), trained to predict the next word in a sentence based on the preceding words. This contextual approach allows BERT to excel in natural language processing tasks by understanding the nuanced meanings of words within their specific contexts. The bi-directionality of a model is important to understand the meaning of a language. The example below illustrates this [54].

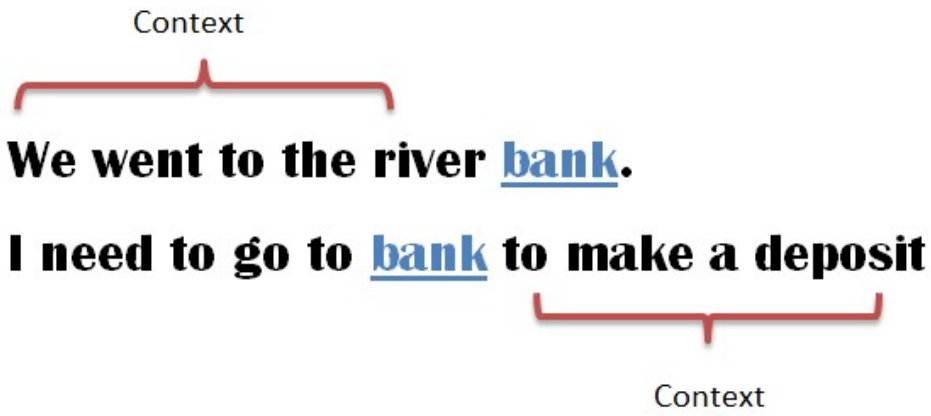

Figure 3.5: Showcase that BERT captures both the left and right context [54]

The left and right contexts are both captured by BERT. Predicting the nature of the word ”bank” using only the left or right context would result in an error in at least one of the two given examples. To address this issue, both the left and right contexts are considered before making a prediction, which is precisely what BERT does.

3.4 Sentence Embedding

Sentence embedding is a fundamental technique in NLP that transforms sentences into dense vector representations, called embedding, which encapsulates their semantic meaning. Unlike traditional bag-of-words models that lose the context, sentence embedding preserves the relationships between words within a sentence. This makes them highly valuable for a range of NLP tasks, including text classification, sentiment analysis, information retrieval, and machine translation. By capturing the essence of sentences in a compact form, sentence embedding facilitates more sophisticated and context-aware text processing.

3.4.1 Sentence Bidirectional Encoder Representations from Transformers

Sentence-BERT (SBERT), introduced in 2019, adapts the BERT architecture by employing siamese and triplet network structures to create semantically meaningful sentence embeddings. significantly enhances efficiency, finding the most similar sentence embeddings among 10,000 sentences in just 5 seconds, compared to BERT's 65 hours. Additionally, SBERT is about 9% faster than the InferSent model and approximately 55% faster than the Universal Sentence Encoder (USE) which is a pre-trained neural network model that generates high-quality sentence embeddings for a wide range of natural language processing NLP tasks on a Graphics Processing Unit (GPU), both of which are leading models for generating sentence embeddings [55]. SBERT introduces the Siamese network concept meaning that each time two sentences are passed independently through the same BERT model. Before discussing SBERT architecture, let us refer to a subtle note on Siamese networks:

SBERT achieves this efficiency by adding a pooling layer to BERT's output, producing fixed-size embedding vectors. The BERT model is then fine-tuned using a siamese network architecture, which updates the weights to ensure that the resulting embeddings are semantically meaningful. These embeddings can be compared using cosine similarity, allowing for accurate measurement of sentence similarity.

3.4.2 Pooling layer

The process of transforming a sequence of embeddings into a single sentence embedding is known as "pooling". Essentially, this involves compressing detailed token-level representations into a fixed-length vector intended to encapsulate the meaning of the entire sequence. Sentence embeddings are compressions. This inherent compression is a key reason why sentence embeddings may struggle to capture fine details in lengthy texts the compression process can be too loss to maintain such granularity.

Text that consists of 200 tokens and uses a language model that produces 512-dimensional embeddings. The resulting embeddings will be 200 distinct 512-dimensional arrays. The sentence embedding will, by contrast, represent the information in the sequence in only one single 512-dimensional array [56].

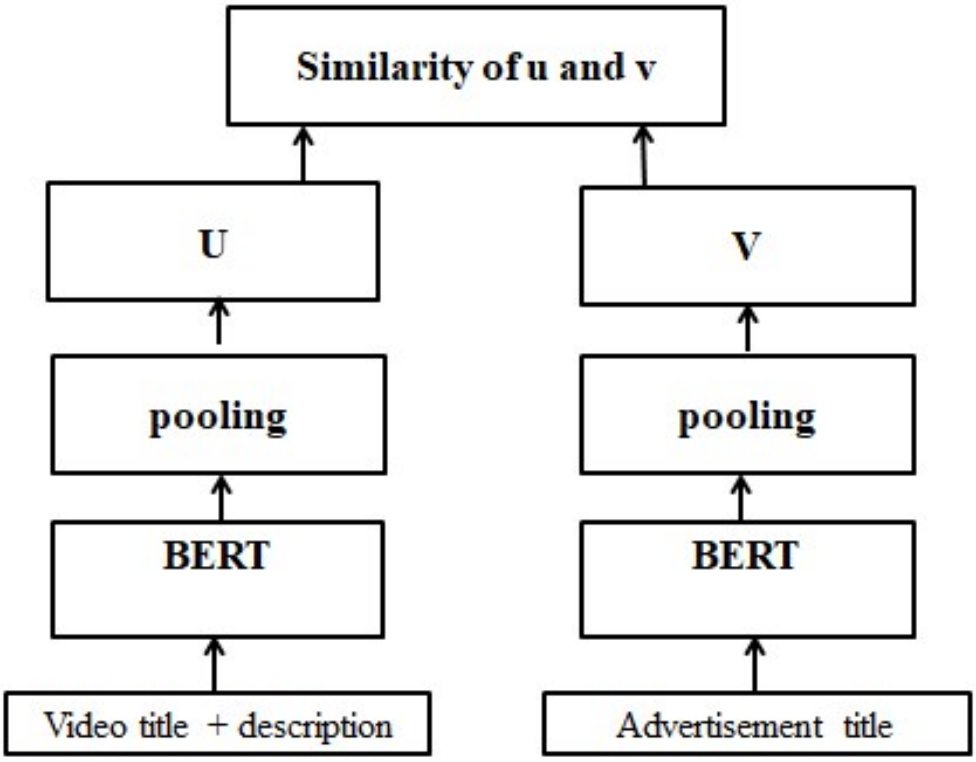

Figure 3.6: SBERT architecture to compute similarity scores modified for the research methodology

The pooling layer is essential for converting the token embeddings generated by the BERT model into a single, fixed-size vector representation for each sentence. When a sentence is processed by BERT, it produces embeddings for each token. The pooling layer then applies one of several methods to these token embeddings:

- **Mean pooling:** to obtain a fixed-length representation (embedding) for the entire sentence, mean pooling calculates the average of all word embeddings in the sentence. In other word it averages the embeddings of all tokens.
- **Max pooling:** takes the maximum value from each dimension across all token embeddings.
- **CLS pooling:** uses the embedding of the special [CLS] token added at the beginning of the sentence or token.

These methods condense the rich contextual information from the token embeddings into a single vector that effectively represents the entire sentence, making it suitable for tasks such as semantic similarity, sentence classification, and clustering [56].

3.5 Distance Similarity Measurement

Distance similarity metrics are essential tools for quantifying how similar two entities are based on their separation in high-dimensional space. These metrics are vital in numerous applications, including NLP. Similarity measurements often involve comparing high-dimensional vector representations of words or sentences. These vector embeddings capture the semantic meaning of the text, allowing better comparisons.There are different kinds of distance similarity measurement the most well known are discussed below.

- **Euclidean distance:** is a line between two points in the Euclidean space. The Euclidean distance between the vector representations of two text documents serves as a proxy for their semantic similarity here documents with smaller Euclidean distances are considered more similar in content, while documents with larger distances are less alike [57].

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

p, q = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

- **Cosine Similarity:** Cosine similarity is a metric used to measure the similarity between two non-zero vectors. The cosine similarity between two vectors is calculated by taking the dot product of the two vectors and dividing it by the product of their Euclidean lengths. Mathematically, this can be expressed as:

$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$

Where A and B are the two vectors, and θ is the angle between them as shown in figure 3.7.

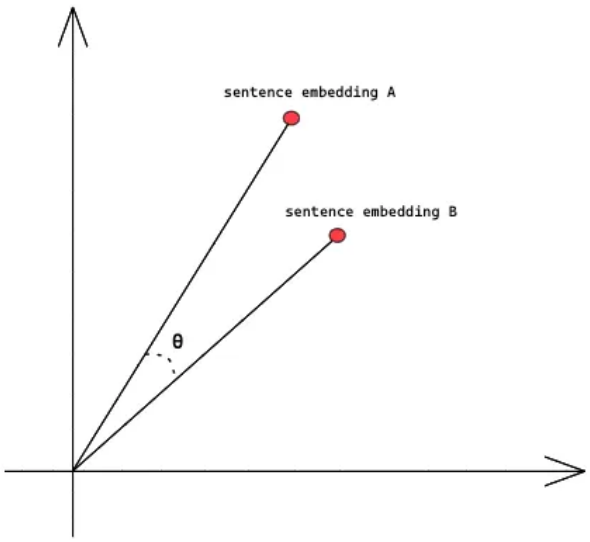

Figure 3.7: Angle between two vectors A and B [57]

The resulting cosine similarity value will range from -1 to 1. A value of 1 indicates that the two vectors have the same direction, 0 indicates they are orthogonal (perpendicular), and -1 indicates they are in opposite directions.

Cosine similarity is a widely used metric in many applications, such as information retrieval, recommendation systems, and text analysis, where quantifying the similarity between high-dimensional vectors is important.

Cosine similarity focuses on the orientation or direction of the vectors, ignoring any differences in their magnitude or scale. This means that two vectors can have a high cosine similarity even if they have very different lengths or magnitudes. For two vectors to have their cosine similarity calculated, they must exist in the same inner product space [58]. This ensures that the vectors can be multiplied using the inner product operation, which produces a scalar value.

Cosine similarity is a powerful tool for measuring similarity in recommendation systems and due to its ability to handle sparse, high-dimensional data. It focuses on the direction of vectors rather than their magnitude. This makes it a preferred method for this research. For the proposed method Cosine similarity is chosen since sentence transformers are specifically designed with cosine similarity in mind.

Chapter 4

Result and Discussion

In this chapter test data preparation, experimental setups, and experiment scenarios used to evaluate the proposed approach with the obtained results and evaluations are discussed.

4.1 Experimental Setup

All modules within the developed recommendation system have been implemented using Python in Google Colab, a versatile cloud-based service offered by Google. Models were retrieved from Hugging Face. Hugging Face is a widely-used online platform that offers open-source machine learning models, serving thousands of organizations globally [59]. In this thesis, the Hugging Face Transformers library was used to generate embeddings for Amharic sentences using sentence transformer. This library provides an extensive collection of pre-trained Transformer-based models, enabling efficient and effective implementation of advanced machine learning techniques.

4.2 Test data preparation

As discussed in section 3.2, advertisement data were collected from the YouTube pages of various advertisement and promotion companies. This data scraped directly from the platform, was initially uncategorized unlike the YouTube video data, all kinds of Advertisement data were collected. Assigning 500 pieces of Ads data for annotators to choose from for each YouTube video was challenging. To address this issue, clustering techniques were implemented to organize the data into manageable groups, making the selection process more efficient and effective for annotators.

4.2.1 Advertsement Data Clustering

The elbow method suggested the optimal number of clusters for the Amharic advertisement data. Elbow Method as discussed in section 3.2.6 used to determine the optimal number of clusters by plotting the sum of squared errors sum of squared errors (SSE) against the number of clusters and looking for an ”elbow” point where the rate of decrease sharply slows. As mentioned in Section 3.2.5, KMeans clustering, which is well-known for unsupervised clustering, was utilized in this analysis. Given that the dataset comprises textual data, various text embedding models were compared to cluster the data based on their contextual meaning. As discussed in section 3.3 Tf-Idf, word2vec, fast text, and BERT language models were used as text embedding models. This combination ensured more accurate and meaningful data categorization.

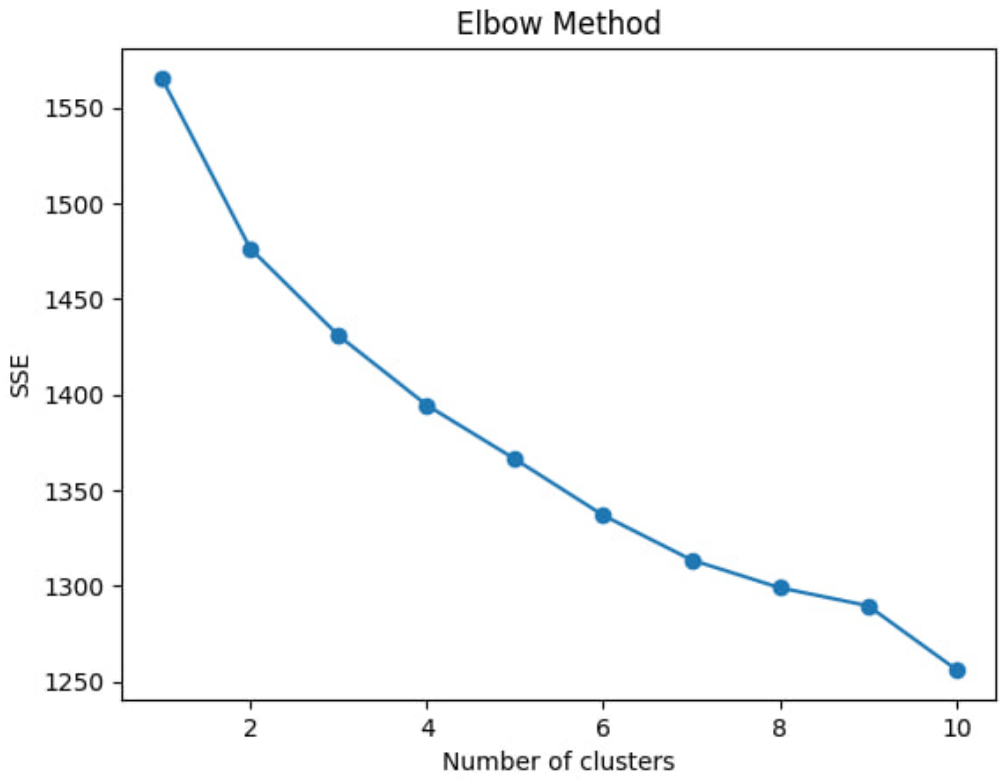

Figure 4.1: Elbow Method to decide the number of categories for Amharic advertisement

Comparing the clustering performance of different text embedding models (tf-idf, Word2Vec, FastText, and BERT) from the figure 4.2 for the Amharic advertisement data, with its balanced cluster distribution and as the data of the cluster from the model implies BERT advanced semantic understanding, BERT appears to be the most effective method for clustering the Amharic Advertisement dataset.

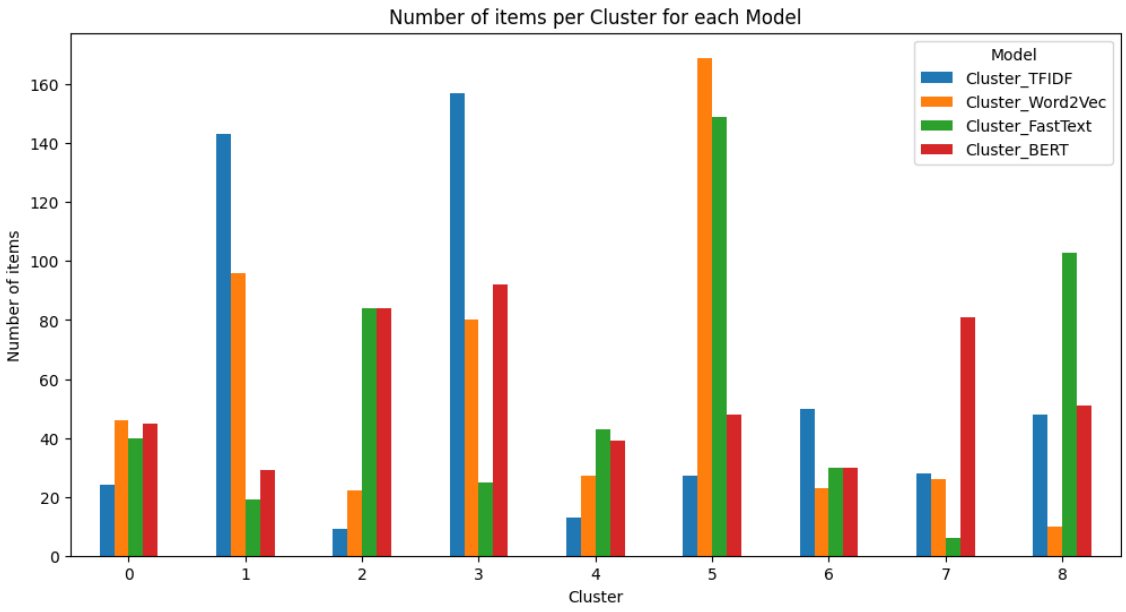

Figure 4.2: Comparison of language models in clustering Ads data

The Amharic advertisement data, clustered using BERT and KMeans clustering, were provided to annotators to create video-advertisement pairs as discussed in Section 3.2.4. Out of 4,500 processed YouTube video data (titles and descriptions), 1,200 entries were randomly selected for this task. These selected videos, along with the clustered advertisement data, were given to the annotators. Before presenting the clustered advertisements as choices, the categories of the YouTube video data were carefully considered. This ensured that the annotators could make better decisions, aligning the clustered advertisements with the relevant video categories, thus enhancing the accuracy and relevance of the video advertisement pairings.

4.3 Experimental Scenario

Two experiments are conducted to evaluate the proposed recommendation system. Based on their intended goal, those experiments are categorized into three:-

- **Experiment I:** Aimed to evaluate the performance of the proposed model in recommending relevant Ads.
- **Experiment II:** Aimed to evaluate the proposed model concerning user cold start.

4.3.1 Experiment I:Model Efficiency in recommendation

BERT models were used to make a contextual embedding for YouTube video titles and descriptions. The same model was used to embed Ads title, here three BERT models fine-tuned for the Amharic language were used to make the sentence embedding, and the models are found in Hugging Face. These models were fine-tuned for tasks like named entity recognition, sentiment analysis, and word question answering. In this experiment, the models shown in table 4.1 were used to embed the large texts from the dataset, the last row in the table is the proposed model for this thesis. Different threshold values were set to check how the embeddings made by the language models responded to the cosine similarity. Three threshold values were selected when the cosine similarity was greater than 0.7, 0.8, and 0.9.

Table 4.1 show models that are used for the embedding. The first model is Amharic sentence BERT this model is train on small data with a maximum sequence length of 512, the second one is multi-lingual Bidirectional Encoder Representations from Transformers (mBERT) bert model, this model is a BERT model that was pre-trained in multi-language including Amharic and the last one is the proposed sentence multi-lingual Bidirectional Encoder Representations from Transformers (SmBERT) model, here mean pooling have been applied with a maximum sequence length of 512 on the multi-lingual BERT to make a SBERT.

	Models
1	Amharic SBERT
2	bert-base-multilingual-cased-finetuned-Amharic(mBERT)
3	bert-base-multilingual-cased-finetuned-Amharic + pooling (SmBERT)

Table 4.1: BERT models fine-tuned for Amharic language

4.3.2 Experiment II:Model Efficiency in cold start

Once the model was developed, an experiment was conducted to evaluate its efficiency in a cold start scenario. In this context, ”cold start” refers to the challenge of making accurate recommendations for new users who have no prior interaction history with the system. To assess this, 50 recommendations generated by the model were selected for evaluation by new users who had not previously seen the videos or the advertisements.

These 50 recommendations were chosen to ensure a robust evaluation, distinct from the data set used for initial labeling, as discussed in Section 3.2.4. The new users provided feedback on these recommendations, allowing us to know how well the model performed without relying on prior user-specific data. This evaluation helps to understand the model's generalization capabilities and its effectiveness in providing relevant recommendations to users who are entirely new to the system.

4.4 Evaluation Metrics

The evaluation metric employed in this experiment is accuracy. While there are other metrics available for evaluating the performance of recommendation systems, accuracy was specifically chosen for this experiment due to its compatibility with the method employed. Accuracy provides a straightforward and comprehensive measure of how well the recommendation algorithm predicts advertisement data to the YouTube videos assessing the proportion of correctly recommended items against the total number of recommendation.

Given the nature of the datasets, which include clear and well-defined true labels for user preferences, accuracy serves as an effective metric to quantify the overall performance of the recommendation system.

4.4.1 Accuracy

Recommendation accuracy is the most basic metric for evaluating a recommendation algorithm, which measures the extent to which the recommendation algorithm can accurately predict the user's preference for the recommended product [60].

The accuracy of the model was calculated by determining how many of the recommendations generated by the system matched the YouTube video and advertisement pairs gained from the annotators. In this context, the top three advertisements were generated by the recommendation model for each video, and three annotated advertisements were provided for each video by the annotators. A correct recommendation was counted as one accurate or relevant recommendation if the recommendation model we proposed included an advertisement that matched one of the three advertisements recommendation-annotated. The accuracy was then computed by dividing the number of correct predictions by the total number of videos. This approach ensures that the system's ability to accurately predict relevant advertisements based on the annotated data is evaluated effectively.

4.4.2 Confidence Interval

A Confidence Interval (CI) is a range of values, derived from a data sample, that is likely to contain the true population parameter, accuracy in this paper case with a certain level of confidence [61]. It provides an estimate of the uncertainty or precision of the sample statistic. It provides an estimated range of values that, with a certain level of confidence (typically 95% or 99%), includes the parameter of interest. provide a range of plausible values for an unknown population parameter, rather than a single point estimate, which helps to understand the variability and uncertainty of the estimate. By reflecting sampling variability, confidence intervals acknowledge that different samples from the same population could yield different estimates, making conclusions more robust.

Taking confidence levels are 95%, the confidence interval for the accuracy of this model in solving cold start is calculated below.

$$\text{CI} = \bar{x} \pm Z \left(\frac{s}{\sqrt{n}} \right)$$

Where:

$\bar{x} =$ Sample mean

$s =$ Sample standard deviation

$n =$ Sample size

$Z =$ Z-score corresponding to the desired confidence level (1.96 for 95% confidence)

4.5 Results

4.5.1 Experiment I: Proposed model efficiency in recommendation

From the experiment conducted in evaluating the recommendation model, as discussed in section 4.3.1 there were three models used for embedding the textual data in both YouTube video textual data and advertisement textual data including the modified SmBERT fine-tuned for Amharic. As shown in the table 4.2, the model's accuracy in recommending relevant Ads shows a significant difference. The proposed model demonstrates much better performance compared to other models due to its efficiency in sentence embedding. As the sentence embedding similarity threshold value increases, the recommendation performance of all models decreases. This indicates that higher similarity thresholds require advertisement titles to be nearly identical in all textual features to the YouTube video titles and descriptions for accurate recommendations.

Models	Cosine Similarity > 0.7	Cosine Similarity > 0.8	Cosine Similarity > 0.9
AmSBERT	30.7%	5.3%	2.8%
mBERT(fine-tuned for Amharic),	40.8%	20.06%	7.5%
mBERT(fine-tuned for Amharic) + pooling (SmBERT)	70.2%	52%	28.02%

Table 4.2: Models performance for YouTube Amharic advertisements on different thresholds of cosine similarity

4.5.2 Experiment II: Proposed model efficiency in cold-start

As discussed in Section 4.3.2, evaluation Ads recommendations were selected randomly from the video advertisement pair generated from the proposed model and assigned to users who had not previously watched the Ads. This approach ensured that the evaluators were cold start users, meaning they had no prior exposure to the advertisements. These YouTube users were asked to watch the videos and determine whether the recommended Ads were relevant to the video content or not.

For this evaluation, approximately 50 videos were selected, each paired with three recommended Ads. These videos, along with their associated advertisements, were then distributed to 27 evaluators. Since the evaluation is made by questioning peoples, the confidence interval was calculated as discussed in section 4.4.2. Therefore Based on their feedback, the following conclusions were drawn:

Out of the 50 video advertisement pairs provided for 27 evaluators, on average about 34 video-recommendation pairs were selected as relevant recommendations, From this, in a cold start scenario, where evaluators had no prior exposure to the advertisements, approximately 68% of the video-advertisement pairs were considered relevant. This indicates that the recommendation system was relatively effective in selecting ads that were appropriate and related to the content of the videos by users with no previous interaction history.

With a 95% confidence interval, the mean of the collected data is approximately (29.18, 38.44). This indicates that there is 95% confidence that the true accuracy of the model in recommending relevant Ads to cold start users lies between 58% and 76%.

Chapter 5

Conclusion and Future work

5.1 Conclusion

This thesis proposed a solution to address the user cold start problem in YouTube Ads recommendation systems using language model embedding. To tackle this issue, approximately 4500 YouTube video titles and descriptions, along with 500 YouTube Ads titles, were collected and processed. Out of these, 1200 videos were selected and presented to annotators along with clustered advertisements. KMeans clustering with various word embedding models were employed to ensure that related advertisements were fairly distributed within the same category of YouTube videos.

Two experiments were conducted to evaluate the proposed methods. The first experiment assessed the efficiency of the system in recommending relevant Ads. The second experiment focused on evaluating the system's effectiveness in addressing the new user problem in YouTube Ads. To evaluate the model 50 recommendations that the system generates were provided to 27 users who have never seen the videos and the Ads before.

Based on the experiments conducted to test the efficiency of the system, the proposed system:

- Demonstrated an accuracy of 70% in recommending relevant Ads when compared to the annotated YouTube video and Ads pairs.
- Showed that at 95% confidence interval, 58% to 76% of the advertisements recommended for the YouTube videos were relevant by new users who had no prior exposure to the Ads.

Depending on these results, It can be concluded that incorporating mean pooling with mBERT allows for effective Amharic sentence embedding. This enhancement enables the system to comprehend sentences meaning. Which improves the relevance of Ads recommendation and helps to address the user cold start problem. Since this method doesn't need extensive data sharing from YouTube users we can conclude that our approach significantly enhances privacy.

5.2 Future Works and Recommendations

finally, The following gaps have been identified as potential directions for future research:

- In this research, the accuracy evaluation was conducted by taking a recommended Ads from the proposed system as matching if it appeared at least once in the annotated data. This method was used regardless of whether there were two or more matching Ads for the video. It would be beneficial to consider and evaluate scenarios where two or more similar advertisements exist between the system data and the annotated data, as this could potentially impact the results.
- In its current form, the developed model can be applied to solve the user cold start problem and recommend relevant Ads for various social media platforms, such as Twitter, TikTok and web pages.
- It would be worth using the proposed model for sentence similarity tasks with labeled datasets.

References

[1] Mehrdad Mollanoroozi. Review on recommender system and architecture. *Majlesi Journal of Telecommunication Devices*, 11(3):177–185, 2023.

[2] Deepjyoti Roy and Mala Dutta. A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1):59, 2022.

[3] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, and Mohamed Hasan Ibrahim. Recommender systems challenges and solutions survey. In *2019 international conference on innovative trends in computer engineering (ITCE)*, pages 149–155. IEEE, 2019.

[4] Sabrina Chowdhury. Evaluating cold-start in recommendation systems using a hybrid model based on factorization machines and sbert embeddings, 2022.

[5] F Bakhshandegan Moghaddam and Mehdi Elahi. Cold start solutions for recommendation systems. In *Big data recommender systems: Recent trends and advances*. IET, 2019.

[6] Bilin Shao, Xiaojun Li, and Genqing Bian. A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Expert Systems with Applications*, 165:113764, 2021.

[7] Noa Tuval, Alain Hertz, and Tsvi Kuflik. Addressing the cold start problem in privacy preserving content-based recommender systems using hypercube graphs. *arXiv preprint arXiv:2310.09341*, 2023.

[8] Piyanuch Chaipornkaew and Thepparit Banditwattanawong. A recommendation model based on user behaviors on commercial websites using tf-idf, kmeans, and apriori algorithms. In *International Conference on Computing and Information Technology*, pages 55–65. Springer, 2021.

[9] Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, 11(1):141, 2022.

[10] Shah Khusro, Zafar Ali, and Irfan Ullah. Recommender systems: issues, challenges, and research opportunities. In *Information science and applications (ICISA) 2016*, pages 1179–1189. Springer, 2016.

[11] Miao Hu, Di Wu, Run Wu, Zhenkai Shi, Min Chen, and Yipeng Zhou. Rap: a lightweight privacy-preserving framework for recommender systems. *IEEE Transactions on Services Computing*, 15(5):2969–2981, 2021.

[12] Seyed Naini, Mohammed Shafia, and Negar Nazari. Examining different factors in effectiveness of advertisement. *Management Science Letters*, 2(3):811–818, 2012.

[13] Pooran Singh et al. A review paper on digital advertising. *International Journal of Innovative Research in Computer Science & Technology*, 9(6):73–76, 2021.

[14] LLC YouTube. Youtube. *Retrieved*, 27:2011, 2011.

[15] Duygu Firat. Youtube advertising value and its effects on purchase intention. *Journal of Global Business Insights*, 4(2):141–155, 2019.

[16] Nathakorn Srithong. Influences of youtube advertising on young adults: A social identity perspective. 2022.

[17] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*, 2015.

[18] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744, 2023.

[19] Séamus Lankford, Haithem Afli, and Andy Way. adaptmllm: Fine-tuning multilingual language models on low-resource languages with integrated llm playgrounds. *Information*, 14(12):638, 2023.

[20] Seid Muhie Yimam, Abinew Ali Ayele, Gopalakrishnan Venkatesh, Ibrahim Gashaw, and Chris Biemann. Introducing various semantic models for amharic: Experimentation and evaluation with multiple tasks and datasets. *Future Internet*, 13(11):275, 2021.

[21] Abinew Ali Ayele, Seid Muhie Yimam, Tadesse Destaw Belay, Tesfa Asfaw, and Chris Biemann. Exploring amharic hate speech data collection and classification approaches. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 49–59, 2023.

[22] Bemnet Meresa Hailu, Yaregal Assabie, and Yenewondim Biadgie Sinshaw. Semantic role labeling for amharic text using multiple embeddings and deep neural network. *IEEE Access*, 11:33274–33295, 2023.

[23] Lutz Edzard. Amharic. In *The Semitic Languages*, pages 202–226. Routledge, 2019.

[24] Benjamin Piper and Agatha J van Ginkel. Reading the script: How the scripts and writing systems of ethiopian languages relate to letter and word identification. *Writing Systems Research*, 9(1):36–59, 2017.

[25] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5382–5390, 2021.

[26] Junmei Feng, Zhaoqiang Xia, Xiaoyi Feng, and Jinye Peng. Rbpr: A hybrid model for the new user cold start problem in recommender systems. *Knowledge-Based Systems*, 214:106732, 2021.

[27] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. Dropoutnet: Addressing cold start in recommender systems. *Advances in neural information processing systems*, 30, 2017.

[28] Sang-Min Choi, Kiyoung Jang, Tae-Dong Lee, Abdallah Khreishah, and Wonjong Noh. Alleviating item-side cold-start problems in recommender systems using weak supervision. *IEEE Access*, 8:167747–167756, 2020.

[29] Joy jeevamoy and VG Renumol. An ontology-based hybrid e-learning content recommender system for alleviating the cold-start problem. *Education and Information Technologies*, 26:4993–5022, 2021.

[30] Usha Yadav, Neelam Duhan, and Komal Kumar Bhatia. Dealing with pure new user cold-start problem in recommendation system based on linked open data and social network features. *Mobile Information Systems*, 2020(1):8912065, 2020.

[31] Joanna Misztal-Radecka, Bipin Indurkhya, and Aleksander Smywiński-Pohl. Meta-user2vec model for addressing the user and item cold-start problem in recommender systems. *User Modeling and User-Adapted Interaction*, 31(2):261–286, 2021.

[32] 2023.

[33] Joseph Kready, Shishila Awung Shimray, Muhammad Nihal Hussain, and Nitin Agarwal. Youtube data collection using parallel processing. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1119–1122. IEEE, 2020.

[34] Wojciech Kwedlo. A clustering method combining differential evolution with the k-means algorithm. *Pattern Recognition Letters*, 32(12):1613–1621, 2011.

[35] Sanghamitra Bandyopadhyay and Ujjwal Maulik. An evolutionary technique based on k-means algorithm for optimal clustering in rn. *Information Sciences*, 146(1-4):221–237, 2002.

[36] Fan Liu and Yong Deng. Determine the number of unknown targets in open world based on elbow method. *IEEE Transactions on Fuzzy Systems*, 29(5):986–995, 2020.

[37] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

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

[39] Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Computing Surveys*, 56(6):1–42, 2024.

[40] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[41] Long Ma and Yanqing Zhang. Using word2vec to process big text data. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2895–2897. IEEE, 2015.

[42] Rafly Kurnia, Y Tangkuman, and A Girsang. Classification of user comment using word2vec and svm classifier. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(1):643–648, 2020.

[43] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[44] Mudasir Mohd, Rafiya Jan, and Muzaffar Shah. Text document summarization using word embedding. *Expert Systems with Applications*, 143:112958, 2020.

[45] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

[46] Usman Naseem, Imran Razzak, Katarzyna Musial, and Muhammad Imran. Transformer based deep intelligent contextual embedding for twitter sentiment analysis. *Future Generation Computer Systems*, 113:58–69, 2020.

[47] S Selva Birunda and R Kanniga Devi. A review on word embedding techniques for text classification. *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020*, pages 267–281, 2021.

[48] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[49] 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.

[50] 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.

[51] Jiajia Wang, Jimmy Xiangji Huang, Xinhui Tu, Junmei Wang, Angela Jennifer Huang, Md Tahmid Rahman Laskar, and Amran Bhuiyan. Utilizing bert for information retrieval: Survey, applications, resources, and challenges. *ACM Computing Surveys*, 56(7):1–33, 2024.

[52] Cameron Hashemi-Pour and Ben Lutkevich. Bert language model, 2024.

[53] Congcong Wang, Paul Nulty, and David Lillis. A comparative study on word embeddings in deep learning for text classification. In *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, pages 37–46, 2020.

[54] Mohdsanadzakirizvi@gmail.com Sanad. Bert: A comprehensive guide to the groundbreaking nlp framework, Sep 2019.

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

[56] Jan Lehečka, Jan Švec, Pavel Ircing, and Luboš Šmídl. Adjusting bert's pooling layer for large-scale multi-label text classification. In *International Conference on Text, Speech, and Dialogue*, pages 214–221. Springer, 2020.

[57] Mahmoud Harmouch. 17 types of similarity and dissimilarity measures used in data science., Mar 2021.

[58] Mathias Leys. The art of pooling embeddings □□ | ml6team, Jun 2022.

[59] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[60] Shuhao Jiang and Jinlong Song. Evaluation metrics for personalized recommendation systems. *Journal of Physics: Conference Series*, 1920, 2021.

[61] Avijit Hazra. Using the confidence interval confidently. *Journal of thoracic disease*, 9(10):4125, 2017.