EMPOWERING MULTIMODAL MULTIMEDIA INFORMATION RETRIEVAL THROUGH SEMANTIC DEEP LEARNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAEID SATTARI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

MARCH 2024

Approval of the thesis:

EMPOWERING MULTIMODAL MULTIMEDIA INFORMATION RETRIEVAL THROUGH SEMANTIC DEEP LEARNING

submitted by **SAEID SATTARI** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _______________

Prof. Dr. Halit Oğuztüzün
Supervisor, **Computer Engineering, METU** _______________

Prof. Dr. Adnan Yazıcı
Co-supervisor, **Computer Science, Nazarbayev University** _______________

Examining Committee Members:

Prof. Dr. Fazlı Can
Computer Engineering, Bilkent University _______________

Prof. Dr. Halit Oğuztüzün
Computer Engineering, METU _______________

Prof. Dr. İsmail Sengör Altıngövde
Computer Engineering, METU _______________

Prof. Dr. Sinan Kalkan
Computer Engineering, METU _______________

Prof. Dr. Murat Koyuncu
Information Systems Engineering, Atılım University _______________

Date:18.03.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: SAEID SATTARI

Signature :

ABSTRACT

EMPOWERING MULTIMODAL MULTIMEDIA INFORMATION RETRIEVAL THROUGH SEMANTIC DEEP LEARNING

SATTARI, SAEID
Ph.D., Department of Computer Engineering
Supervisor: Prof. Dr. Halit Oğuztüzün
Co-Supervisor: Prof. Dr. Adnan Yazıcı

March 2024, 86 pages

Multimedia data encompasses various modalities, including audio, visual, and text, necessitating the development of robust retrieval methods capable of harnessing these modalities to extract and retrieve semantic information from multimedia sources. This study presents a highly scalable and versatile end-to-end multimodal multimedia information retrieval framework. The core strength of this system lies in its capacity to learn semantic contexts within individual modalities and across different modalities, achieved through the utilization of deep neural models. These models are trained using combinations of queries and relevant shots obtained from query logs. One of the distinguishing features of this framework is its ability to create shot templates representing videos that have not been encountered previously. To enhance retrieval performance, the system employs clustering techniques to retrieve shots similar to these templates. An improved variant of fuzzy clustering with a modified loss function is applied to address the inherent uncertainty in multimodal concepts. Our approach goes beyond simple cluster-based ranking by incorporating Siamese networks for improved re-ranking, thereby enhancing retrieval precision. Additionally, a fusion method incorporating an OWA operator is introduced. This method employs various measures to aggregate ranked lists produced by multiple retrieval systems. The proposed approach leverages parallel processing and transfer learning to extract features from three distinct modalities, ensuring the adaptability and scalability of the framework. To assess its effectiveness, the system is rigorously evaluated through experiments conducted on six widely recognized multimodal datasets. Remarkably, our approach outperforms previous studies in the literature on five of these datasets. The experimental findings, substantiated by statistical tests, conclusively establish the effectiveness of the proposed approach in the field of multimodal multimedia information retrieval.

Keywords: Multimodal multimedia retrieval, Deep semantic learning, Adaptive fuzzy clustering, Information fusion, Siamese ranking, Ranked lists aggregation

ÖZ

SEMANTİK DERİN ÖĞRENME YOLUYLA MULTİMODAL MULTİMEDYA BİLGİ ERİŞİMİNİ GÜÇLENDİRME

SATTARI, SAEID
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Halit Oğuztüzün
Ortak Tez Yöneticisi: Prof. Dr. Adnan Yazıcı

Mart 2024 , 86 sayfa

Multimedya verileri, ses, görsel ve metin de dahil olmak üzere çeşitli yöntemleri kapsar ve bu yöntemler, multimedya kaynaklarından anlamsal bilgileri çıkarmak ve almak için bu yöntemlerden yararlanabilecek sağlam erişim yöntemlerinin geliştirilmesini gerektirir. Bu çalışma, oldukça ölçeklenebilir ve çok yönlü, uçtan uca çok modlu bir multimedya bilgi erişim çerçevesi sunmaktadır. Bu sistemin temel gücü, derin sinir modellerinin kullanımıyla elde edilen, bireysel yöntemler içindeki ve farklı yöntemler arasındaki anlamsal bağlamları öğrenme kapasitesinde yatmaktadır. Bu modeller, sorgu kombinasyonları ve sorgu günlüklerinden elde edilen ilgili çekimler kullanılarak eğitilir. Bu çerçevenin ayırt edici özelliklerinden biri, daha önce karşılaşılmamış videoları temsil eden çekim şablonları oluşturabilmesidir. Geri alma performansını artırmak amacıyla sistem, bu şablonlara benzer çekimleri almak için kümeleme tekniklerini kullanır. Çok modlu konseptlerdeki doğal belirsizliği gidermek için değiştirilmiş kayıp fonksiyonuyla bulanık kümelemenin geliştirilmiş bir çeşidi uygulanır. Yaklaşımımız, gelişmiş yeniden sıralama için Siyam ağlarını dahil ederek basit küme tabanlı sıralamanın ötesine geçiyor ve böylece erişim hassasiyetini artırıyor. Ayrıca OWA operatörünü içeren bir füzyon yöntemi tanıtılmıştır. Bu yöntem, birden fazla erişim sistemi tarafından üretilen sıralanmış listeleri bir araya getirmek için çeşitli önlemler kullanır. Önerilen yaklaşım, çerçevenin uyarlanabilirliğini ve ölçeklenebilirliğini sağlayarak üç farklı yöntemden özellikler çıkarmak için paralel işleme ve aktarım öğreniminden yararlanır. Etkinliğini değerlendirmek için sistem, yaygın olarak tanınan altı çok modlu veri kümesi üzerinde gerçekleştirilen deneyler aracılığıyla titizlikle değerlendirilir. Dikkat çekici bir şekilde, yaklaşımımız literatürde bu veri kümelerinin beşi üzerinde yapılan önceki çalışmalardan daha iyi performans göstermektedir. İstatistiksel testlerle desteklenen deneysel bulgular, önerilen yaklaşımın çok modlu multimedya bilgi erişimi alanında etkinliğini kesin olarak ortaya koymaktadır.

Anahtar Kelimeler: Çok modlu multimedya erişimi, Derin semantik öğrenme, Uyarlanabilir bulanık kümeleme, Bilgi birleştirme, Siamese listeleme, Sıralı listeler birleştirme

To My Family

ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my supervisor, Prof. Dr. Adnan Yazici, for his invaluable advice, continuous support, and patience during my Ph.D. study. His plentiful experience has encouraged me in all the time of my academic research and daily life. I thank all the jury members for their valuable comments and feedback. Finally, I would like to express my gratitude to my adorable wife, Moloud, and my supportive parents. Without their tremendous understanding and encouragement over the past few years, it would be impossible for me to complete my studies.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

OWA	Ordered Weighted Average
CCA	Canonical Correlation Analysis
KCCA	Kernel Canonical Correlation Analysis
DCCA	Deep Canonical Correlation Analysis
FCM	Fuzzy C-Means
CNN	Convolutional Neural Network
CBOW	Continuous Bag of Words
AAFCM	Alternative Adaptive Fuzzy C-Means
DNN	Deep Neural Network
NDCG	Normalized Discounted Cumulative Gain
CG	Cumulative Gain
DCG	Discounted Cumulative Gain

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Nowadays, storing and sharing a considerable amount of multimedia data has been boosted rapidly due to the advances in multimedia capturing devices. Consequently, there is a high demand for multimedia information retrieval systems that utilize multimedia content representation to search and retrieve relevant multimedia data. Generally, each multimedia data comprises three primary modalities: video, audio, and text. Incorporating complementary information from various modalities can enhance the accuracy of retrieved multimedia data. However, effectively utilizing multimodality for multimedia data retrieval is a prominent area of ongoing research [1]. Given that each modality can comprehend the semantics associated with other modalities, they can mutually complement each other in the search for multimedia content. Furthermore, leveraging the contextual relationships between modalities is crucial [2], as each modality can compensate for the weaknesses of the others.

When semantic concepts are annotated automatically for each modality, various concepts may overlap and correlate between modalities. Contextual relationships between semantic concepts are strongly dependent on the vocabulary used. However, in practical scenarios, data is not always complete with labels and full multimodal information, making applying these techniques challenging. An automatic semantic annotation may contain wrong labels and errors. These errors can appear in all modalities, but for most concepts, such as ‘car,‘ it is less likely that all three modalities will not detect it. Therefore, indirect information can be inferred using a combination of concepts from other modalities. Various studies have been proposed in multimedia information retrieval, but most focus on single or dual modalities.

Lately, neural network techniques have often been used in information retrieval [3]. Deep learning approaches are powerful since they facilitate revealing complex correlations and associations from multimedia data [4]. Therefore, they have been used to label and retrieve multimedia data using high-level semantic concepts. The study in [5] presents various aspects of deep learning in multimedia information retrieval. Most existing studies focus on automatic labeling, extracting semantic content, and classifying multimedia data using convolutional neural networks (CNN) [6, 7].

Traditional approaches exhibit efficacy in information retrieval (IR) yet encounter challenges when matching the query with multimodal multimedia data. The semantic match between the query and the multimedia data is necessary to help the user effectively obtain relevant information and diminish the semantic gap. To our knowledge, although there are several notable studies on the application of deep neural networks (DNN) to multimedia data [6, 7, 8], there are no studies on multimedia information retrieval using visual, audio, and text modalities and creating associations between queries and video shots for efficient retrieval of multimodal queries. When multiple retrieval engines operate in an IR system, they prepare separate lists of relevant documents for a submitted query. According to various studies, combining these lists improves overall accuracy [9]. When there are multiple retrieval systems or search engines, each has a weight that indicates its importance and effectiveness. Therefore, incorporating the degree of significance as a reasonable weight can improve the accuracy of the final ranked list. Although various rank-based and score-based studies and solutions [10] to aggregate multiple ranked lists were proposed, most do not address the significance of numerous retrieval engines.

In information retrieval systems, the assumption that highly related documents are associated with the same queries proves to be one of the most effective techniques. However, matching each document with every query is inefficient, as such an approach would consume unnecessary time. Instead, considering cluster-based retrieval as an effective retrieval system is more practical when relevant documents exhibit more remarkable similarity to non-relevant ones. Various methods have been proposed to reduce the number of document-query comparisons. However, recent studies have not given adequate attention to cluster-based retrieval in the context of multimodal multimedia retrieval research. Additionally, conventional clustering algorithms struggle to handle the challenges posed by multimedia data, where the nature and semantic context are often ambiguous and uncertain from the user's perspective.

Ranking in multimedia retrieval is a fundamental part of information retrieval systems [11] that offers several notable benefits that enhance the retrieval experience and the overall utility of multimedia retrieval systems [12]. Machine learning models have played a key role in shaping the ranking landscape in multimedia retrieval [13]. Approaches containing deep learning, neural networks, and Siamese networks have emerged as challenging tools for modeling the complex relationships between multimedia items and user queries. These models can capture both semantic and perceptual content, allowing for a more detailed understanding of content relevance.

1.2 Proposed Methods and Models

Given the problems above, challenges, and the latest achievements in multimedia information retrieval, this thesis proposes a system that leverages deep neural network models to learn contextual semantics within each modality and among the different modalities in multimedia data. After extracting semantic content from three modalities, deep learning techniques are applied to represent them as deep word embedding vectors. After that, two deep generative networks are introduced to create a query-level model (template) by fusing diverse semantic contexts from the three modalities. Subsequently, we use the generated template to retrieve multimedia data (videos) similar to a submitted query from fuzzy clustered spaces. We also propose a method based on the Ordered Weighted Averaging (OWA) [14] operator to fuse ranked lists from various retrieval engines. Within our information fusion approach, we consider multiple elements such as document ranking, relevance scores, and the importance weights assigned to various retrieval systems. Additionally, we leverage Siamese networks that improve retrieval performance by re-ranking the results.

1.3 The Research Objectives

The research objective of this thesis is to develop and present a comprehensive approach for multimodal multimedia information retrieval, an essential aspect of searching and querying multimedia data. The focus is on leveraging deep learning models and historical logs to create Shot Templates for previously unseen videos, enabling the extraction of contextual semantics across different media data modalities. The study also addresses challenges such as noisy labels and missing modalities. The primary aim is to enhance retrieval performance by using clustering techniques to identify similar shots that match patterns and employing fuzzy clustering to handle multimodal concept uncertainty. Another goal is introducing a fusion method based on an ordered weighted average (OWA) operator, effectively combining ranked lists from multiple retrieval engines. The approach utilizes parallel processing and transfer learning to extract features from three modalities to ensure scalability and adaptability to new multimedia datasets. The effectiveness and effectiveness of this approach are validated through experiments on six popular multimodal datasets, demonstrating its proficiency in ranking retrieval results for complex queries. The thesis's research objective is further supported by conducting statistical tests, which provide additional evidence of the approach's effectiveness in multimodal multimedia information retrieval. Ultimately, the goal is to contribute to advancing effective and accurate retrieval techniques for diverse multimedia data, facilitating better access and understanding of complex information for users.

1.4 Theoretical and Practical Implications of the Thesis

Our research has significant theoretical implications in the domain of multimodal multimedia information retrieval. By developing deep neural network models, we have advanced the understanding of how contextual semantics can be learned within and among different modalities in multimedia data. This contributes to the broader field of deep learning and provides insights into how neural networks can effectively handle multimodal information. Furthermore, integrating fuzzy clustering and novel deep learning models in our proposed system has improved multimedia information retrieval effectiveness and overall performance. This has theoretical implications for developing more accurate and sophisticated retrieval techniques to handle uncertainties in multimodal concepts and noisy data.

On the practical side, our research offers several valuable contributions to multimedia information retrieval. The proposed scalable and flexible multimedia feature extraction and multimodal information retrieval system presents a suitable solution for effectively handling large-scale multimedia datasets. By leveraging parallel processing and adapting the Apache Spark engine, our approach addresses the challenges of dealing with vast amounts of multimedia data, making it practical for real-world applications. Additionally, introducing the Ordered Weighted Averaging (OWA)-based fusion approach has practical implications for enhancing retrieval accuracy. This approach considers multiple factors from various retrieval systems, leading to more precise and relevant retrieval results for users. Overall, our research's practical implications lead to more effective and accurate retrieval of relevant multimedia data across different modalities. This will significantly benefit users by facilitating easier access and analysis of multimedia content, ultimately improving the usability and effectiveness of multimedia retrieval applications.

1.5 Contributions and Novelties

The contributions of this study can be specified as follows:

- Our study introduces an innovative end-to-end method for extracting multimedia features specifically designed to meet the scalability requirements of multimodal information retrieval. This method's distinctive feature is the use of parallel processing through the Apache Spark engine, significantly boosting the effectiveness in processing large datasets.
- We present a unique approach to learning contextual semantics. Utilizing query logs and deep learning models, we have developed a concept called "template shot," which integrates ideas from multiple modalities. This method effectively discerns contextual and semantic links both within and across different modalities, greatly facilitating the search for contextual connections.

- In our study, we have integrated various clustering techniques with custom-designed similarity functions into our system architecture, improving its effectiveness. A notable aspect of this integration is our advanced fuzzy clustering technique, which effectively addresses the uncertainty inherent in video data, thereby enhancing the performance of multimedia information retrieval.
- We enhance the retrieval accuracy by developing a ranking technique alongside evidence that employing a Siamese network equipped with Triplet loss substantially improves precision.
- The study introduces a novel fusion method that employs the Ordered Weighted Average (OWA) operator. This method distinctively merges document rankings, relevance scores, and performance evaluations from various retrieval systems, thereby improving the precision and accuracy of retrieval.
- Our study makes a notable contribution by rigorously evaluating the effectiveness of our approach across six renowned multimodal datasets. Impressively, our method outperforms existing studies in this domain on five datasets, demonstrating performance improvements ranging from 1.5% to 10.1% over the best results previously reported in those respective studies.

1.6 The Outline of the Thesis

The rest of the thesis is organized as follows. Chapter 2 explains previous studies related to our research. Chapter 3 describes preliminary materials and notations. Chapter 4 illustrates our proposed approach in detail. Chapter 5 offers the experimental results, the dataset used in our experiments, and the evaluation of the performance results. Finally, we conclude our study and point out some potential future paths in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

Multimodal multimedia information retrieval is an extensive research area [15, 16, 17, 18, 19]. The rapid progress in deep multimodal learning has recently opened new avenues for enhancing multimedia information retrieval [20]. The remarkable potential of deep multimodal learning has been demonstrated in improving the performance of retrieval systems by enhancing their ability to comprehend and process diverse types of data and modalities encountered in the retrieval process [4, 21, 22]. A scalable deep multimodal learning (SDML) retrieval method is introduced in [23], which defines a common subspace where the difference between classes is maximized, and the similarity within classes is minimized. Their proposed method is effective and efficient in multimodal learning and outperforms existing methods. Recently, cross-modal fusion has been extensively investigated and applied in various domains [24, 25, 26, 27], yielding highly promising outcomes.

The multimodal semantic autoencoder for cross-modal retrieval (MMSAE) presented in [28] introduces a two-phase learning technique for transforming multimodal data into low-dimensional embeddings in a way that maintains feature and semantic details. The usefulness of the proposed study is illustrated through experiments on benchmark datasets. In a similar survey [29], the multimodal semantic analysis problem is addressed by leveraging a regularized semantic autoencoder. Another initiative in this direction focuses on exploring image-text embedding techniques to enhance the effectiveness of multimodal retrieval [30]. This study [31] focuses on multimodal data analytics, specifically cross-data research. The research aims to outline an Intelligent Cross-Data Analysis and Retrieval.

An intelligent multimodal multimedia information is presented in [32], which annotates and indexes the semantic content of videos using visual, auditory, and textual modalities. The approach in [33] addresses the multimodal retrieval problem by exploiting semantic relevancy within each modality (intra-modal) and among the modalities (inter-modal). This work adopts a famous Canonical Correlation Analysis (CCA) method [34], maximizing the multimodal correlation to learn a shared space. They use the correlation between terms and modalities to establish connections between the different modalities by analyzing co-occurrence information.

Some kernel methods, such as Kernel CCA (KCCA), are proposed to cultivate non-linear models in [35]. Another study in [36] discusses Multi-View CCA (MCCA), which learns non-linear relations between two multidimensional random variables (modalities) by leveraging the kernel trick. In the context of multi-view data representation, Distance-Based Kernel Canonical Correlation Analysis is presented in [37]. The high-level idea of this study is to non-linearly map diverse datasets into a joint subspace, emphasizing the importance of non-linear complementary information between views within the common subspace. Finally, this study [38] proposed an approach that learns a joint space by maximizing the inter-class differences and minimizing intra-class changes.

Recent works have widely used DNN to migrate multimodal data into a joint space [39, 40]. Deep Canonical Correlation Analysis (DCCA) suggested in [41] learns non-linear conversions so that the final data demonstrations have strong relationships. This study [42] proposed a scalable multi-label CCA (sml-CCA), a system to concurrently combine the correlation between semantics and the correlation between features for retrieving data from multimodal sources. Researchers in this study [43] proposed a multimodal coordinated network (MCCN), which is composed of two parts: a Multimodal Coordinated Embedding (MCE) and a Multimodal Contrastive Clustering (MCC) part for retrieving from extensive cross-modal data.

Cross-modal generative adversarial networks (CM-GANs) [39] is presented to tackle modeling data from distinct modalities in a shared distribution environment. Deep Multimodal Transfer Learning (DMTL) [44] introduced a method to map learning from labeled sources to enhance the results for new and unlabeled categories. They create multiple neural networks specialized in different modalities to develop a joint semantic space across them. This study [45] proposed an approach using the multimodal adversarial network (MAN) technique that migrates multimodal data to a joint space where a uniform metric can calculate similarities between modalities. This involves utilizing several generators for specific modalities, a discriminator, and a multimodal discriminant analysis loss. Following related studies in the literature, to our knowledge, the latent relationships of more than two modalities with noisy labels and missing modalities have not been sufficiently investigated.

Clustering has been utilized successfully in IR domain tasks [46, 47]. An effort in this direction is a thorough review, variants analysis, and advantages of clustering algorithms in big data application [48]. Another study we adopt in this work is understanding the competency between distance-based and density-based clustering approaches in multimedia information retrieval [49]. The survey in [50] proposes a Locality Sensitive Clustering (LSC) approach for efficient clustering and retrieval of multimedia data in high-dimensional space. Their experimental results show that their method is faster than K-means in the bag of words samples. In [51], researchers present a fuzzy cluster-based model that groups associated synonyms into separate clusters, which assists in extracting the semantic meaning of a query. Their results illustrate the effectiveness of the model. The study in [52] proposes a retrieval system that employs feature extraction, clustering, and machine learning methods. They demonstrate through the use of fuzzy C-means that the retrieval performance is both fast and reliable. In [53], authors present a video summarization and retrieval system using K-means clustering and employing high and low-level features [48]. A particular instance of this idea includes clustering incomplete multi-view data, which is addressed in [54]. In contrast to many existing methods that primarily use the Euclidean distance metric for clustering, our approach adopts the Cosine similarity measure. This choice is particularly effective for high-level multimedia information retrieval, offering enhanced similarity assessment in computational efficiency, an essential factor for handling high-dimensional spaces like those in our application. Additionally, we incorporate a fuzzy clustering algorithm to address the often-overlooked uncertainties in multimedia data found in these studies. Our method offers a more flexible and refined solution, adeptly tackling the prevalent challenges of uncertainty in large-scale information retrieval scenarios.

The order in which the results are significantly presented in IR systems impacts the user experience. Traditional ranking algorithms, such as TF-IDF and BM25, have been widely used. However, learning a ranking function [55] has emerged as a powerful paradigm for improving the quality of ranked results. Our observations reveal a general trend in the literature where re-ranking methods are frequently employed to enhance search effectiveness. Consequently, re-ranking has become a critical post-processing procedure within information retrieval [56, 57, 58]. Typically, there are three different categories of ranking approaches: Pointwise approaches [59], pairwise approaches [60], and Listwise approaches [61]. Accordingly, there are popular methods [62] employed in IR, such as RankNet, LambdaMART, RankSVM, RankBoost, and ListNet. Deep learning approaches have had a profound impact on the field of ranking in various applications, especially in tasks where the goal is to determine the order or relevance of items [63, 64, 65]. These approaches can handle multimodal data and learn joint representations, which can be valuable for ranking tasks involving diverse content types, such as multimedia retrieval. As discussed earlier, Siamese networks are a type of deep learning architecture commonly used for ranking tasks. They learn embeddings emphasizing similarity or dissimilarity between items, making them well-suited for information retrieval. Ranking with Siamese networks is an effective approach for tasks where the goal is to order items based on their similarity or relevance [66]. They are designed to learn embeddings that encode the intrinsic relationships between pairs of items, ensuring that similar items are placed closer in the embedding space while dissimilar ones are pushed farther apart [67, 68].

CHAPTER 3

PRELIMINARIES

Some materials and notions are needed to follow up on the main ideas in this research easily. We describe these notions in this section.

3.1 Shot Boundary Detection

Identifying shot boundaries or transitions is a crucial aspect of video content analysis. Shot boundary detection refers to the process of automatically locating the transitions between shots in a digital video, which allows for the temporal segmentation of the video. This process breaks a video down into primary time units known as shots. A shot comprises consecutive frames recorded by a single camera, representing a coherent and uninterrupted activity in both temporal and spatial domains. This step is crucial for automated indexing and content-based video retrieval applications, facilitating access to extensive video archives. There have been numerous studies on shot boundary detection, and our research utilizes the TransNet V2 neural network model [69], which achieves state-of-the-art results on established benchmarks.

3.2 Keyframe Extraction

Selecting salient frames from a video shot is known as keyframe extraction. In this process, a frame that differs from each other and encodes the highest possible information out of all the similar and consecutive frames in a group is denoted. These keyframes give an overview of the video shot content. Key-frames provide more flexibility in video content analysis and adopt them without fully processing all frames in a video shot. Various studies have been proposed to extract candidate frames. In our study, we utilize Katana [70], which automates the task of video keyframe extraction.

3.3 Transfer Learning

EfficientNet [71] is a convolutional neural network (CNN) used to extract image embedding vectors and image classification. Various versions of the EfficientNet model, such as EfficientNetB7, the latest version, reached the highest performance. EfficientNet models outperform previous models on many datasets while using fewer resources. We adopt the EfficientNet model to extract features by excluding the final dense layer from the model. As a result, the resulting model creates a 512-dimensional feature vector for an input image size of 224x224.

YAMNet [72] is a pre-trained Convolutional Neural Network that processes an audio waveform to predict 521 different audio events. It can also generate 1024-D feature embeddings for transfer learning. These embeddings can then be fed into another shallow network to predict new audio events, allowing for the rapid development of specialized audio classifiers. The model works by dividing the audio signal into frames and processing them in batches. It takes in a waveform as single-channel 16 kHz samples, frames it into windows of 0.96 seconds with a 0.48-second hop, and produces embeddings from these frames. The scores generated can be used to identify audio events by aggregating the per-class scores across the frames.

3.4 Word Embedding

Distributed representation, or distributed encoding, represents data using features or attributes where each feature contributes to the overall representation. In distributed representation, data is not stored in isolated, independent features but rather in the collective patterns and relationships among these features. In a distributed representation, different aspects or attributes of the data are interrelated and contribute jointly to the representation, allowing for more efficient and flexible information processing. This can be particularly useful in neural network-based learning tasks, where the relationships between data elements are complex and interconnected. One of the most famous examples of distributed representation is the word embeddings created using techniques like Word2Vec [73] and ELMo [74]. In these embeddings, each word is represented as a vector in a high-dimensional space, and the relationships between words are captured by the distances and directions between these vectors, enabling models to understand semantic and contextual relationships between words.

Word2Vec is built on representing words as dense, continuous vectors in a high-dimensional space. This contrasts traditional methods that use sparse one-hot encoding to represent words. Word2Vec exploits the distributional hypothesis, which asserts that words with similar meanings often appear in similar contexts. It relies on the local co-occurrence patterns of words in a large corpus of text to learn their vector representations. The two main architectures of Word2Vec are Continuous Bag of Words (CBOW) and Skip-gram. CBOW aims to predict a target word given its surrounding context words, while Skip-gram does the opposite, predicting the context words based on a target word. Both architectures leverage neural networks, specifically shallow feedforward neural networks, for training. The training process involves iteratively adjusting the word vectors using stochastic gradient descent to minimize the negative log-likelihood of the model. Word2Vec uses a negative sampling technique to sample non-context words, making the training process computationally efficient and scalable to large datasets. Unlike traditional autoencoders that learn by reconstructing the input, Word2Vec trains the words based on their relationships to other words within the input lexicon. The resulting word embeddings produced by Word2Vec possess exciting properties. Words with similar meanings are clustered in the vector space, allowing for semantic relationships to be accurately captured. Given sample data and context, Word2Vec can effectively predict a word's meaning based on its past usage

ELMo (Embeddings from Language Models) is a groundbreaking deep contextualized word representation technique. Unlike traditional word embeddings that rely on fixed, static representations, ELMo generates dynamic, context-dependent embeddings by leveraging the power of language models. It is based on a deep bidirectional language model pre-trained on a large corpus of unlabeled text data. This pre-training process enables ELMo to capture complex patterns and contextual information. ELMo's unique feature lies in its ability to generate word embeddings sensitive to the context in which the words appear. It considers the surrounding words and captures their impact on the target word's meaning. This contextual adaptability makes ELMo highly effective for various tasks. The architecture of ELMo comprises multiple layers of bi-directional Long Short-Term Memory (LSTM) networks, which work in tandem to capture both forward and backward context information. These LSTM layers create a rich representation of words by encoding the local and global contextual relationships. It is usually used as a pre-trained layer, and fine-tuning is performed on the task-specific dataset. The final output is a set of contextualized word embeddings, providing deep insights into the underlying semantics of the concepts. ELMo's exceptional performance on various benchmark datasets and its ability to capture complex semantic concepts have made it an excellent choice for researchers. Its contextual embeddings have significantly improved the accuracy and robustness across various applications. In this study, we employ these two techniques to transform one-hot vectors into real-valued vectors, which are subsequently used as inputs for neural networks.

3.5 Cluster-Based Retrieval

In a collection of documents, those that are related tend to be relevant to the same query. To utilize these relationships, one method is to group the documents into clusters. This way, documents in the same cluster have similar relevance to query results. In information retrieval, clustering can improve efficiency and cost-effectiveness in retrieving information or identifying document connections. The core idea behind this approach is that clusters offer a more comprehensive representation of document content, allowing for faster retrieval by comparing the higher-level patterns represented by the clusters rather than making exhaustive comparisons on individual document vectors. In this study, we employ K-means [75] and DBSCAN [76] for hard clustering.

Traditional clustering algorithms struggle to handle the complexities of real-world data, which can be ambiguous and uncertain. For example, the content of a video shot may belong to multiple categories. Fuzzy clustering solves these challenges as it can effectively deal with uncertainty. The fuzzy clustering approach creates clusters with flexible boundaries, allowing an object to belong to multiple clusters with varying membership levels. The degree of membership reflects the strength of an object's association with a specific cluster. There are several fuzzy clustering techniques. Because of its inherently ambiguous nature, fuzzy clustering is more appropriate for handling multimedia data than conventional clustering methods. Our study utilizes Fuzzy C-means (FCM) [77] and Alternative Adaptive Fuzzy C-Means (AAFCM) [78] methods for soft clustering.

3.6 Fuzzy C-Means

Clustering analysis seeks to group different objects into distinct classes, to place the most similar objects into the same class. Fuzzy C-Means (FCM) is a soft clustering technique that allows individual data points to belong to two or more clusters. It operates by minimizing this objective function:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \, d^2(x_i, c_j) \quad \text{where } \; 1 \leqslant m < \infty \tag{3.1}$$

Where m is the weighting exponent, a real number greater than one, that defines the fuzziness level, u_{ij} is the membership degree of x_i to j^{th} cluster. Subsequently, x_i presents the i^{th} d dimensional data point, c_j denotes the d dimensional center (mean vector) of the cluster, c is clusters counts, and d represents any metric that calculates the similarity between a data point and the center of a cluster (centroid). The suggested values of m are 1.5 to 2.5, and 2.0 is the favored selection. The optimization of the objective function through an iterative process results in fuzzy partitioning, which involves updating both the membership matrix u_{ij} and the cluster centers c_j.

$$u_{ij} = 1 \Big/ \sum_{k=1}^{c} \left[\frac{d^2(x_i, c_j)}{d^2(x_i, c_k)} \right]^{\frac{2}{m-1}} \quad \text{and} \quad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \, x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{3.2}$$

Depending on the application domain, different objective functions can be used. In multimedia retrieval, the similarity measure for two multimedia shots (documents) represented as vectors is mainly related to the Cosine similarity used in the high dimensional positive space. After trying two objective functions, we observe that the Cosine similarity objective function outperforms the least square error. Therefore, we use the Cosine similarity objective function instead of the least square error objective function. So, the objective function becomes:

$$J_m = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m} \left(1 - \cos(x_i, c_j)\right)^2 \quad \text{where} \;\; 1 \leqslant m < \infty \tag{3.3}$$

The algorithm, which is based on a finite set of data points, returns a list of c centroids represented by C and a membership matrix denoted by U, as shown below.

- $C = c_j \quad \text{where} \quad j = 1, \ldots, c$
- $U = u_{ij} \;\; \text{where} \quad i = 1, \ldots, n \;\; \& \;\; j = 1, \ldots, c$

where u_{ij} is a real value in the range of [0,1], which represents the membership level of the data point x_i to the cluster j with c_j as a centroid vector.

3.7 Alternative Adaptive Fuzzy C-Means

Alternative Adaptive Fuzzy C-Means (AAFCM) [78] is a traditional FCM clustering algorithm variant. It involves adapting the membership function and updating the cluster centers iteratively to improve the effectiveness of the clustering process. It uses an alternative distance vector presented in Equation 3.4. This variant aims to handle data with inherent uncertainty and overlapping membership more effectively by adjusting the membership degrees of data points across multiple clusters. This approach is robust to noise and outliers, and clusters of unequal sizes. Besides, an adaptive clustering procedure enables the classification of subsequent data without reprocessing the entire dataset. It helps address the challenges posed by noisy data or situations where data points exhibit varying degrees of similarity to different clusters. The algorithm is formalized as follows.

$$d_A(x, y) = 1 - \exp(-\beta d^2(x, y)) \tag{3.4}$$

where $d^2(x - y)$ is square Euclidean distance and β is a positive constant defined in Equation 3.5.

$$\beta = \left[\sum_{i=1}^{n} d^2(x_i, \bar{x}) \Big/ n \right]^{-1} \quad \text{where} \quad \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{3.5}$$

It has demonstrated that AAFCM minimizes the following objective function with similar parameters outlined in Equation 3.6.

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \left[1 - e^{-\beta d^2(x_i, c_j)}\right] \quad \text{where} \quad 1 \leqslant m < \infty \tag{3.6}$$

3.8 Ranked Lists Aggregating

When multiple retrieval systems (engines) perform searches within the same data repository, they return different ranked lists of documents. Typically, a data fusion algorithm merges two or more ranked lists into a unified list, thereby improving overall effectiveness compared to utilizing individual systems for data fusion [79, 80].

The performance of retrieval systems varies since there are multiple factors like query complexity, relevance, and context. These factors collectively determine the performance of retrieval systems and require careful consideration. As a result, we address this issue by learning the weight of each retrieval system that signifies its importance and effectiveness. Leveraging this significance via learning proper weights yields an improved aggregated ranked list. Likewise, a fusion of the ranked lists refers to scenarios where a query is submitted to multiple retrieval systems having access to the same document collection. The output lists from various systems are merged into a single list during fusion. One of the main advantages of employing fusion on the ranked list is that the fused list is often more relevant compared to individual retrieval systems [81]. Methods for fusing multiple retrieval results are categorized into different approaches. The following techniques are some of the well-established techniques utilized in our study. Typically, these approaches are applied without incorporating the significance weights of individual retrieval systems as parameters.

1. Score-based

   - CombMIN: Minimum relevance score
   - CombMAX: Maximum relevance score
   - CombSUM: Aggregation of relevance scores
   - CombMNZ: CombSUM × number of non-zero scores

    - CombANZ: CombSUM / number of non-zero scores

2. Rank-based
    - Borda Count
    - Condorcet

The first three methods in the score-based approach use basic operators to find rank scores. CombMNZ [82] represents the combined similarity for the overlapped documents, multiplied by the number of the appearance of results with a non-zero similarity score. CombANZ is like the previous method but uses division instead of multiplication.

$$CombMNZ(d) = \mid \{\forall i \mid d \in Rank_i\} \mid \times \sum_i s_i(d) \tag{3.7}$$

Borda Count [83] denotes a voting method based on candidates' positions in the rank-based approach. It assigns a weight based on the positions at which a candidate appears within the ranked lists of voters. The document receives a weight linked to its inverse position in the ranked list. Condorcet [83] is a voting algorithm based on the majoritarian approach that selects the best candidate in an election. It uses a pairwise comparison of two results $r(s_1) > r(s_2)$, then for each pair (s_1, s_2), it compares the number of times s_1 beats s_2. Finally, the best candidate is identified through pairwise comparison.

3.9 OWA Operator

The Ordered Weighted Averaging (OWA) of degree d is a non-linear aggregation operator with this mapping: $[0,1]^d \to [0,1]$ and a weight vector $W = [w_1, w_2, w_3, \ldots, w_d]$ such that $\sum_{j=1}^{d} w_j = 1$ and $w_j \in [0,1]$. This operator is defined as follows.

$$OWA(a_1, a_2, \ldots, a_d) = \sum_{j=1}^{d} w_j b_j \quad \text{where} \quad b_j \text{ is } j^{th} \text{ largest } a_i \tag{3.8}$$

where $A = [a_1, a_2, \ldots, a_d]$ is the argument vector to be aggregated. In our study, the weights used for aggregation are produced by a learning mechanism. A comprehensive framework to fuse results based on OWA is presented in [84]. To consider retrieval systems' weights, we interpret the meta-search task as a Multi-Expert Decision Making (MEDM) problem and solve this task using the OWA operator. From the decision-making standpoint, each retrieval system is considered an expert.

3.10 Siamese Network

The Siamese network [85] is a neural network architecture designed for tasks involving similarity or distance measurements between inputs. They are often used for image similarity, verification, and ranking tasks. Siamese networks consist of two or more identical subnetworks that share weights and are used to process pairs of inputs. These networks are commonly used in applications that aim to determine whether two inputs are similar or dissimilar. Siamese networks are used for pairwise ranking tasks, where we have a set of items and want to decide on their relative order of preference or relevance. Siamese networks can learn to map items into an embedding space where their representations reflect their characteristics.

A Siamese network comprises two parallel branches (subnetworks) that share weights. Each branch processes one of the input items. The network learns to produce embeddings for each input so that similar inputs have embeddings closer in the embedding space, while dissimilar inputs are farther apart. During training, the network is fed pairs of inputs along with their labels indicating whether they are similar or dissimilar. The loss function is designed to minimize the distance between embeddings of similar pairs and maximize the distance between embeddings of dissimilar pairs. The learned embeddings can be used to calculate distances or similarities between items. Items with smaller distances in the embedding space are considered more similar, and those with larger distances are considered less similar.

The mathematical definition of a Siamese network involves the following components:

1. **Input:** The Siamese network takes pairs of input items, such as images or text representations, as input. Let x_1 and x_2 denote the input items.

2. **Subnetworks:** The network architecture includes two identical subnetworks

(often implemented as convolutional neural networks (CNNs) for image data or recurrent neural networks (RNNs) for sequential data), which process each input item independently. Let $f(x_1)$ and $f(x_2)$ represent the outputs (embeddings) of the two subnetworks for the input items x_1 and x_2, respectively.

3. **Feature Extraction Layers:** Each subnetwork typically consists of multiple layers for feature extraction. For image data, these layers may include convolutional layers and pooling layers for spatial abstraction. For sequential data like text, recurrent layers or attention mechanisms may be used for capturing sequential dependencies.

4. **Embeddings:** The output of each subnetwork is an embedding vector representing the input item in a high-dimensional feature space. These embeddings are learned during the training process and capture the important characteristics of the input data.

5. **Similarity Metric:** After obtaining the embeddings $f(x_1)$ and $f(x_2)$, a similarity metric is applied to measure the similarity between the two embeddings. Standard similarity metrics include Euclidean distance, cosine similarity, or other distance functions.

6. **Loss Function:** The Siamese network is trained using a loss function that compares the similarity between the embeddings of pairs of input items with their ground truth labels (indicating whether the items are similar or dissimilar). Contrastive or triplet loss are commonly used loss functions for training Siamese networks.

7. **Training Strategy:** During training, pairs of input items, along with their labels, are fed into the network. The network parameters (weights) are updated using backpropagation and gradient descent to minimize the loss function, effectively learning to generate embeddings that place similar items closer together in the embedding space while pushing dissimilar items farther apart.

In Siamese networks, the choice of loss function plays an essential role in training the network to learn embeddings that effectively measure similarity or dissimilarity between pairs of inputs. The goal is to minimize the loss when processing similar pairs of inputs and maximize it for dissimilar pairs. Contrastive loss and triplet loss are particularly common due to their effectiveness in learning embeddings representing similar relationships. In this study, we utilized both of these loss functions.

3.10.1 Contrastive Loss

Contrastive loss is a function commonly used in Siamese networks, a type of neural network architecture designed for tasks related to similarity learning. It encourages similar pairs of items to be closer to each other in the embedding space while pushing dissimilar pairs farther apart. In the context of Siamese networks, contrastive loss is typically used to train the network by comparing pairs of inputs and adjusting the model's parameters to minimize the loss. The loss function penalizes the model when the distance between embeddings of similar items is large and encourages the distance between embeddings of dissimilar items to exceed a certain margin. Mathematically, the contrastive loss function can be defined as follows:

$$L(a, b, y) = (1 - y)\frac{1}{2}\text{distance}(a, b)^2 + y\frac{1}{2}\text{max}(0, margin - \text{distance}(a, b))^2 \quad (3.9)$$

In this formula:

- a and b are the embeddings of the two input items.
- y is the label indicating whether the pair is similar $(y = 1)$ or dissimilar $(y = 0)$.
- $distance(a, b)$ represents the distance metric (e.g., Euclidean distance or cosine similarity) between the embeddings a and b.
- $margin$ is a hyperparameter that specifies the desired minimum separation between similar and dissimilar pairs.

The contrastive loss function consists of two terms:

1. The first term penalizes the model when the distance between embeddings of similar items $(y = 1)$ is large.
2. The second term penalizes the model when the distance between embeddings of dissimilar items $(y = 0)$ is small, but only if this distance falls below the

margin. This term contributes zero to the loss if the distance exceeds the margin.

By minimizing this contrastive loss, the model learns to map similar items closer together in the embedding space while pushing dissimilar items farther apart, effectively learning to discriminate between them.

3.10.2 Triplet Loss

Triplet loss is a loss function used in machine learning for tasks related to similarity learning, often in the context of tasks like image retrieval, face recognition, or recommendation systems. The goal of triplet loss is to learn embeddings (vector representations) of items so that similar items are closer to each other in the embedding space while dissimilar items are farther apart. In triplet loss, training data is organized into triplets, each consisting of an anchor item, a positive item, and a negative item. The anchor item is a reference point, the positive item is similar to the anchor, and the negative item is dissimilar. The loss function encourages the network to reduce the distance between the anchor and positive items while increasing the distance between the anchor and negative items by a specified margin.

Formally, let a, p, and n denote the embeddings of the anchor, positive, and negative items, respectively. Triplet loss is computed for each triplet as follows:

$$\text{Triplet Loss} = \max(0, \text{distance}(a, p) - \text{distance}(a, n) + \text{margin}) \tag{3.10}$$

where:

- a represents the embedding of the anchor item.
- b represents the embedding of the positive item (similar to the anchor).
- n represents the embedding of the negative item (dissimilar to the anchor).
- $distance(a, b)$ represents the distance metric (e.g., Euclidean distance or cosine similarity) between the embeddings a and b.

- $margin$ is a hyperparameter that specifies the minimum desired difference between the distances of positive and negative pairs. It prevents the network from pushing similar items too close together or dissimilar items too far apart.

The loss function penalizes the model if the distance between the anchor and the positive item is less than the distance between the anchor and the negative item by an amount less than the margin. Otherwise, if the difference exceeds the margin, the loss is zero. This encourages the embeddings to be closer for similar items and farther apart for dissimilar items, with a desired separation specified by the margin. By optimizing the network with triplet loss, it learns to map similar items closer together and dissimilar items farther apart in the embedding space, thus improving its ability to capture similarities and differences between items.

CHAPTER 4

METHODOLOGY

We propose a scalable and flexible end-to-end architecture designed for multimodal and multimedia information retrieval. The proposed architecture comprises four primary modules: video processing, semantic content annotation, template vector generation, and a multimodal and multimedia information retrieval system with ranking method and fusion technique. The overall architecture is illustrated in Fig. 4.1. We also implement five algorithms associated with these components. Algorithm 1 and Algorithm 2 demonstrate the working logic of template vector generation. Algorithm 3 explains how similar vectors (shots) are retrieved from the clustered space. Algorithm 4 and Algorithm 5 are linked to the fusion component. In Algorithm 4, we illustrate the process of calculating importance scores for multiple retrieval systems, and Algorithm 5 presents the OWA-based fusion strategy.

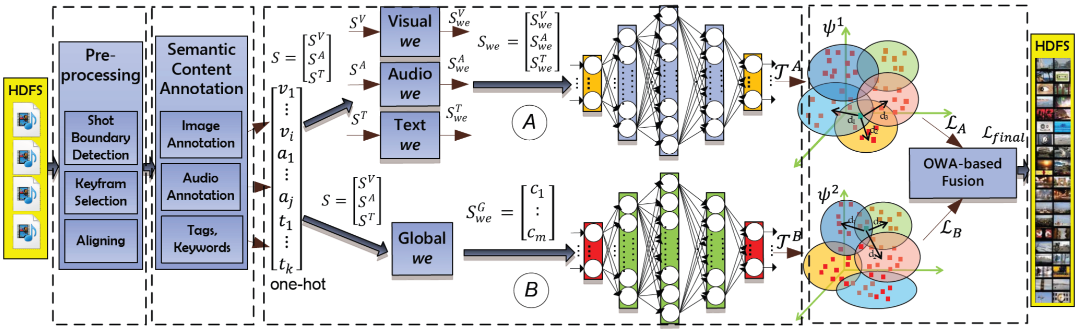

HDFS
Pre-processing
Shot Boundary Detection
Keyfram Selection
Aligning
Semantic Content Annotation
Image Annotation
Audio Annotation
Tags, Keywords
$S = \begin{bmatrix} S^V \\ S^A \\ S^T \end{bmatrix}$
$\begin{bmatrix} v_1 \\ \vdots \\ v_i \\ a_1 \\ \vdots \\ a_j \\ t_1 \\ \vdots \\ t_k \end{bmatrix}$
one-hot
S^V
S^A
S^T
Visual we
Audio we
Text we
S^V_{we}
S^A_{we}
S^T_{we}
$S_{we} = \begin{bmatrix} S^V_{we} \\ S^A_{we} \\ S^T_{we} \end{bmatrix}$
A
$\mathcal{T}^A$
$S = \begin{bmatrix} S^V \\ S^A \\ S^T \end{bmatrix}$
Global we
$S^G_{we} = \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix}$
B
$\mathcal{T}^B$
ψ^1
ψ^2
$\mathcal{L}_A$
$\mathcal{L}_{final}$
$\mathcal{L}_B$
OWA-based Fusion
HDFS

Figure 4.1: The overall architectural design of the proposed approach.

4.1 Video Processing

In this module, the processing of each multimedia data involves four key steps. Initially, each multimedia file's video, audio, and text parts are separated and subsequently transcoded and stored in distinct files. Identifying shot boundaries or transitions is a crucial aspect of video content analysis. This procedure aims to automatically locate the transitions between shots in a digital video, which allows for the temporal segmentation of the video. This process breaks a video down into primary time units known as shots. A shot comprises consecutive frames recorded by a single camera, representing a coherent and uninterrupted activity in both temporal and spatial domains. This step is crucial for content-based video retrieval applications, facilitating access to large video archives. Numerous studies are focusing on shot boundary detection. The proposed study utilizes the TransNet-V2 neural network model [69], which achieves state-of-the-art results on established benchmarks.

Keyframe extraction is selecting individual or sets of frames that effectively represent a shot. This process aims to retain the most salient features of the shot while filtering out redundant frames. During this process, frames that capture the maximum information from among all the similar and consecutive frames are identified and labeled. These keyframes give an overview of the video shot content. Keyframes provide more flexibility in video content analysis since they facilitate learning the visual concepts without fully processing all frames in the shot. Various studies are proposed to extract candidate keyframes. In our study, we utilize Katana [70], which automatically discovers which frames are concise representations of the shot. The audio is split into segments in the final steps aligned with a shot's start and stop time.

4.2 Semantic Content Annotation

In the current architecture, the semantic content extraction module automatically annotates the semantic concepts of multimedia data using three distinct modalities. EfficientNet [71] is a convolutional neural network used to extract image embedding vectors and image classification. Various versions of the EfficientNet model, such as EfficientNetB7, the latest version, reached the highest performance. EfficientNet models outperform previous models on many datasets while using fewer resources. We adopt and fine-tune the EfficientNet model to extract features from keyframes by excluding the final dense layer in the model. Therefore, the model creates a 512-dimensional feature vector for an input image size of 224x224. By employing a Softmax layer, we labeled all keyframes with multiple classes followed by their scores.

YAMNet [72] is a pre-trained Convolutional Neural Network that processes an audio waveform to predict 521 different audio events. It can also generate 1024-D feature embeddings for transfer learning. These embeddings can then be fed into another shallow network to predict new audio events, allowing for the rapid development of specialized audio classifiers. The model works by dividing the audio signal into frames and processing them in batches. It takes in a waveform as single-channel 16 kHz samples, frames it into windows of 0.96 seconds with a 0.48-second hop, and produces embeddings from these frames. The scores generated can be used to identify audio events by aggregating the per-class scores across the shot. In the proposed study, the acoustic classifications that aligned with the shot's length are evaluated as audio concepts.

Within the proposed system, subtitles, tags, and textual content associated with a video represent the textual modality. Furthermore, the textual semantic extraction module incorporates a method for identifying named entities such as individuals, locations, and organizations within the video's tags and subtitles. This study employs a rule-based hybrid named entity recognizer that learns from annotated data [32]. Consequently, the extracted named entities, tags, and specific keywords are considered part of the textual modality. The concepts identified by the semantic context extraction module are encoded as one-hot vectors representing a multimedia shot, as depicted in Fig. 4.2.

4.3 Distributed Representation

In various studies [33, 34], one-hot modeling is adapted to present visual, audio, and textual modalities as a vector. Each index in this vector is a decimal in the [0,1] range, indicating the weight of that particular concept. The one-hot vector representation is

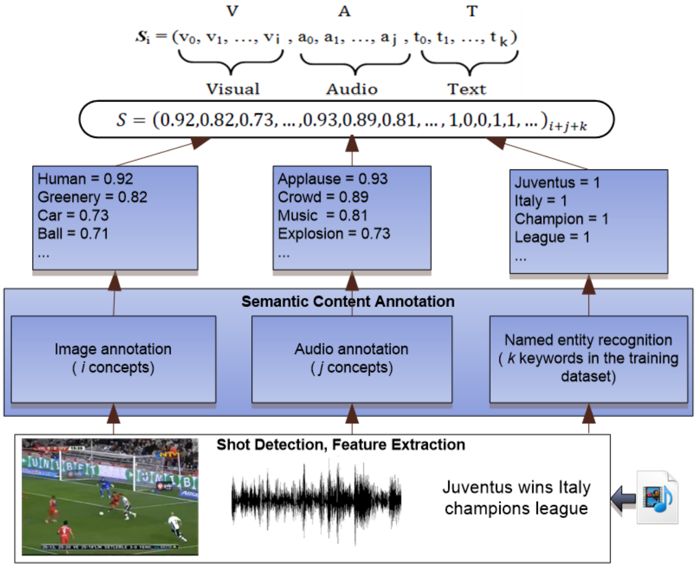

V
A
T
$S_i = (v_0, v_1, \ldots, v_i, a_0, a_1, \ldots, a_j, t_0, t_1, \ldots, t_k)$
Visual
Audio
Text
$S = (0.92, 0.82, 0.73, \ldots, 0.93, 0.89, 0.81, \ldots, 1, 0, 0, 1, 1, \ldots)_{i+j+k}$
Human = 0.92
Greenery = 0.82
Car = 0.73
Ball = 0.71
...
Applause = 0.93
Crowd = 0.89
Music = 0.81
Explosion = 0.73
...
Juventus = 1
Italy = 1
Champion = 1
League = 1
...
Semantic Content Annotation
Image annotation
(i concepts)
Audio annotation
(j concepts)
Named entity recognition
(k keywords in the training dataset)
Shot Detection, Feature Extraction
Juventus wins Italy champions league

Figure 4.2: Automatic semantic content detection.

human-readable but sparse and inefficient for neural networks. Since we use DNNs to train our model, we should convert one-hot representation into enhanced numerical vectors. In this study, two different embedding models are leveraged to transform one-hot vectors into real-valued vectors, subsequently used as inputs for neural networks. Word2Vec [73] is an unsupervised model. The basic idea behind word2vec is to use neural networks to learn a high-dimensional representation of words based on their context, i.e., the words that appear near them in a corpus of text. These embeddings can be used as input to other models, such as text classification, information retrieval, and machine learning. With sufficient data and context, word2vec can accurately infer a word's meaning based on its past usage.

ELMo (Embedding from Language Model) is a deep contextualized word representation technique that has achieved state-of-the-art word embedding performance [74]. It represents each word in a given text as a high-dimensional vector incorporating information from the surrounding context, allowing for a more nuanced and comprehensive understanding of its meaning. In addition, The model can represent the unknown or out-of-vocabulary words into vectors as the ELMo is character-based. ELMo embeddings are computed from a deep bidirectional language model trained on a large corpus of text data. These embeddings have been shown to perform well in various tasks, such as information retrieval, named entity recognition, and text classification.

This study uses word2vec and ELMo methods as a global word embedding model trained across all modalities. The study adopts these methods for generating word embeddings from one-hot vectors. We introduce two approaches for transforming one-hot vectors into the word embedding in the proposed framework by employing these methods. In the first approach, distinct word embedding models are trained for each modality, resulting in vectors that locally represent concepts within a specific modality. This approach, denoted "Intra-modal", enables the model to capture contextual relationships between concepts within the same modality.

Conversely, the second approach involves training a global word embedding model that simultaneously leverages information from all modalities. Referred to as "Inter-modal," this model exploits the correlations between modalities for conceptual association. By considering the contextual relations among various concepts from different modalities, this approach aims to benefit from the shared context influencing every concept across modalities. Consequently, it effectively captures semantic relations among concepts from diverse modalities. Following the training of word embedding models, each concept is represented by a d-dimensional vector, with the parameter d determining the dimensionality of the vectors. This parameter can be defined heuristically using best practices or fine-tuned through cross-validation, ensuring an optimal representation of concepts in the vector space. We depict both approaches explained above in Fig. 4.3.

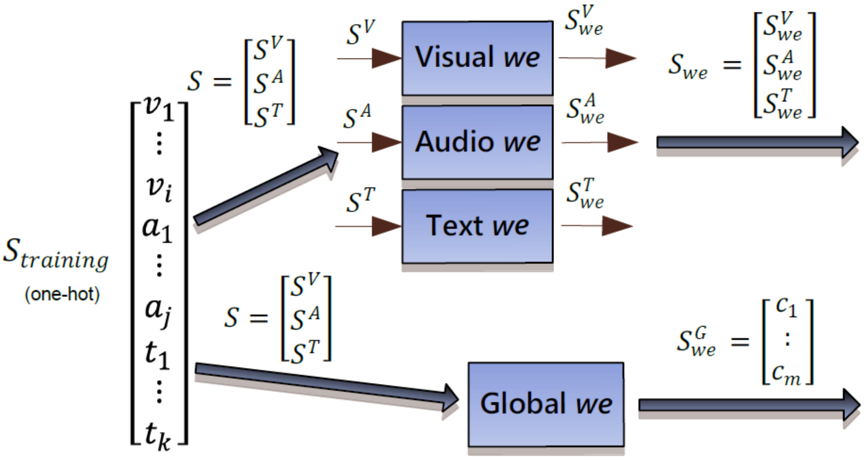


Figure 4.3: Word embedding learning.

After learning the word embedding (we) presentation, we display modalities with $visual_{we}(v) = [\ldots]_{1\times p}$, $audio_{we}(a) = [\ldots]_{1\times q}$ and $text_{we}(t) = [\ldots]_{1\times r}$ notations. To present a shot, word-embedded vectors of concepts are averaged over the number of concepts as:

$$S^V_{we} = \sum_{k=i}^{m} visual\_we(t_k) \Big/ m \tag{4.1a}$$

$$S^A_{we} = \sum_{k=i}^{n} audio\_we(t_k) \Big/ n \tag{4.1b}$$

$$S^T_{we} = \sum_{k=i}^{o} text\_we(t_k) \Big/ o \tag{4.1c}$$

where t_k denotes the terms (concepts) contained in the shot. Accordingly, m, n, and o are the number of concepts with scores greater than zero in each modality. Furthermore, the final shot vector is obtained by merging the word embedding representation of modalities as:

$$S_{we} = [S^V_{we}, S^A_{we}, S^T_{we}]_{(1\times(p+q+r))} \tag{4.2}$$

So, if we present a sample shot with a one-hot vector as:

$$Shot = [0,0,1,0,1,1,1,1,0,1,0,\ldots,1,0] \Longrightarrow S = (S^V, S^A, S^T) \tag{4.3}$$

and considering the number of vocabularies in our dataset, visual modality is represented in the word embedding models with:

$$S^V = [0,0,1,\ldots,0]_{\,1\times i} \Longrightarrow S^V_{we} = \sum_{k=i}^{m'} visual_{we}(t_k) \Big/ m' \tag{4.4}$$

In the same way, we convert queries to numerical vectors according to the terms (concepts) in the query. If a one-hot vector represents a query:

$$(Q^V, Q^A, Q^T) \Longrightarrow Q = [0,1,1,0,1,0,1,0,1,0,\ldots,1,1]_{1\times(i+j+k)} \tag{4.5}$$

Then, for a sample query, each modality is described by the following equations so that m', n', and o' represent the number of query terms in each modality.

$$Q^V = [0,0,1,\ldots,0]_{1\times i} \Longrightarrow Q^V_{we} = \sum_{k=i}^{m'} visual_{we}(t_k) \Big/ m' \tag{4.6a}$$

$$Q^A = [0,0,1,\ldots,0]_{1\times j} \Longrightarrow Q^A_{we} = \sum_{k=i}^{n'} audio_{we}(t_k) \Big/ n' \tag{4.6b}$$

$$Q^T = [0,0,1,\ldots,0]_{1\times k} \Longrightarrow Q^T_{we} = \sum_{k=i}^{o'} text_{we}(t_k) \Big/ o' \tag{4.6c}$$

and the final query vector representation is as follows:

$$Q_{we} = [Q_{we}^{V}, Q_{we}^{A}, Q_{we}^{T}]_{1\times(p+q+r)} \tag{4.7}$$

In the global word embedding model, we leverage all modalities together and at the same time. In this case, every query and every shot are transformed directly into numerical vectors without being separated into specific modalities:

$$Q_{we}^{G} = \sum_{k=0}^{x} Global\_we(t_k) \Big/ n \tag{4.8}$$

where t_k denotes the query terms (keywords) within the query, and n is the total number of these terms. In this case, we aim to benefit from the contextual relations among various concepts from different modalities. We employ this approach since the context of all modalities affects every concept. As a result, semantic relations among concepts from various modalities are captured. In the latter case, every query and every shot are transformed directly into numerical vectors without being separated into specific modalities.

4.4 Template Vector Generation

Deep Neural Networks can uncover complex correlations and associations within input data. We use historical log data in the training dataset to output the queries and their corresponding shots $(Q, S)_{training}$ to define a model that generates any potential shot for a given query. Consequently, we use these data to train a deep neural network. All queries and shots inside the training dataset are transformed into word-embedded forms using learned models. In the network training step, an input for DNN is a query, and the output layer is a shot related to the query. There are multiple shots as a result of each query in the training set. Therefore, all of these shots should be fed into DNN sequentially. A DNN is trained using combinations of all queries and shots. We aim to learn a template vector ($\mathcal{T}$) by utilizing different combinations of queries and logs of their result sets from the training dataset. Fig. 4.4 presents how we train models.

We train a deep neural network that generates a representative template for a hypothetical shot according to the given query. In addition to the query/shots pairs in the

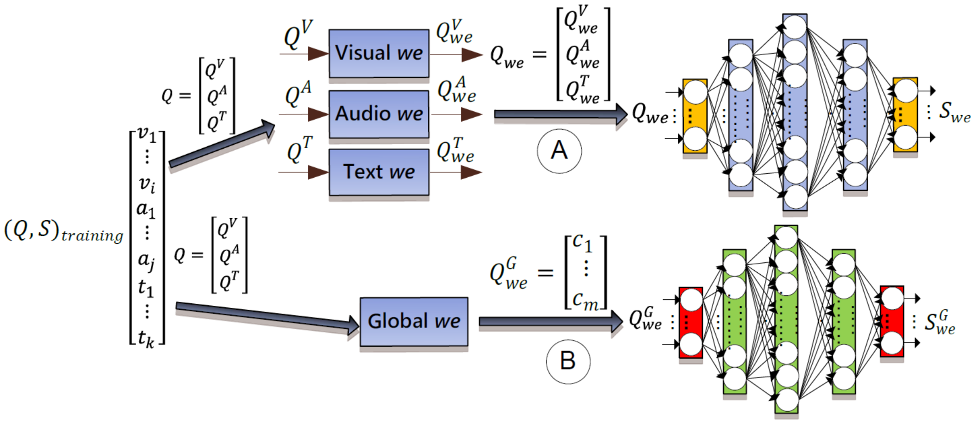


Figure 4.4: Models training steps.

training set, different query and result pairs are prepared for training the network according to the following procedures. There are queries that a shot can be a candidate result for each shot. One candidate is the query that contains all the concepts of the shot based on the ground truth (the representation of the query is the same as the representation of the shot). This specific shot is also expected to be a candidate result for any subset of this query. Consider an arbitrary shot S_1 from the ground truth data with the following concepts:

S_1=(football_player, football_ball, greenery, applause, crowd, Juventus, Italy, league)
If we compose sample queries with any combination of these concepts:
q_1=(football_player, football_ball, greenery, applause, crowd, Juventus, Italy, Euro league)
q_2=(football_player, greenery, crowd, Juventus)
q_3=(greenery, crowd, Juventus, Euro league)
q_4=(Juventus, league)

When these queries are submitted to a retrieval system, it is expected that S_1 to be in the list of result sets. We compose a set of queries for each shot in the ground truth by randomly selecting different concepts from that shot. For each shot in the training data, combinations of 1, 5, 10, and 15 concepts are randomly chosen to generate 1-term, 5-term, 10-term, and 15-term queries. This process is repeated five times to prepare more data for training. Then, we prepare a training dataset with query and shot pairs used to train a deep neural network model. Generating more training examples improves the model performance by providing more extensive data and preventing overfitting.

Each query is segmented into three parts to provide inputs for the DNN, comprising concepts of a single modality. Similarly, the same segmentation is applied to the shot to divide each into three parts. For each part, an associated word-embedding model is used to convert the concepts into real value vectors by averaging vectors' concepts over the number of concepts. After creating a vector for each part, these vectors are concatenated side-by-side to create a single query vector and a single shot vector. Now, these vectors are used to train the deep network model. The query vectors are fed into the input layer, and the shot vectors are considered outputs to train the network.

The DNN model in this study is similar to a variational autoencoder. However, instead of training against the same output through reconstruction, it trains query vectors against matching documents (shots) by minimizing the Cosine loss function [86]. We tried the Least Square Error (LSE) and Cosine loss and observed that the Cosine loss outperformed the LSE loss function. Therefore, we use Cosine loss. This model attempts to find the matching document for a given query based on the contexts learned with enough data, usage, and contexts. The network architecture we use in this research is elaborated on in the experimental section.

In the test stage, queries and shots in the test set are similarly represented as vectors using the learned word embedding models. A hypothetical vector (template) is generated in the output for any test query as an input. This vector represents a possible result when the test query is fed into the trained deep neural network. Suppose we create such a vector model; if this vector depicts the semantic meaning of terms in the query and the correlation between the terms, we can use this model to discover similar vectors in the space of the shots. To achieve this, we can retrieve the most similar shots by utilizing the Cosine similarities between the model vector and shot vectors.

After we trained the network, we tested the networks as follows. For any test query, we convert it to a word embedding model and then generate an output vector for that. We refer to the output vector as a shot template. This template is a vector representation of a potential shot resulting from the submitted test query. Suppose we represent all shots in the word embedding space. In that case, we search for vectors more similar to the template vector (according to the Cosine similarity) while retrieving the results.

For both approaches in word embedding models, separate networks are trained (A and B). The sole variation in their design lies in the size of each layer's neurons and the input configuration. A general overview of the testing steps is shown in Fig. 4.5.

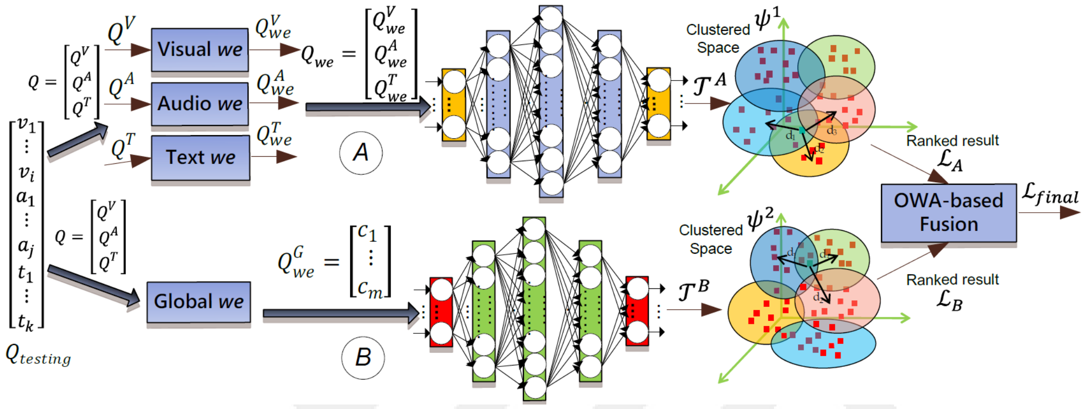


Figure 4.5: Overall testing steps.

To retrieve results for the following sample query:

$$Q = (Q^V, Q^A, Q^T) \Longrightarrow Q = [0, 0, 1, 0, 1, 1, 1, 0, 1, 0, \dots, 1, 0]_{1\times(i+j+k)} \tag{4.9}$$

According to Equation 4.6, vectors are calculated according to the learned models. The result is the following vector which is used as an input to the neural network that generates the template.

$$Q_{we} = [Q^V_{we}, Q^A_{we}, Q^T_{we}]_{1\times(p+q+r)} \tag{4.10}$$

Then, the generated template is used to retrieve results from the vector space (video shot space). Two different clustering methods are used while retrieving results from video shot space. In the next section, more details about these clustering approaches are given. Algorithms 1 and 2 present two retrieval systems (A and B) depicted in Fig. 4.5. In Algorithm 1, the retrieval system A is presented. In this algorithm, modalities are presented in separate word embedding models. Correspondingly, the retrieval system B is shown in Algorithm 2, where the modalities are combined and presented in the global (G) word embedding model.

Algorithm 1 Retrieval system A using multiple embedding models

```
Input: $Q = (Q^V, Q^A, Q^T)$                                   ▷ Input query
output: $l^A$                                                  ▷ Ranked result
Initialize:
    $template^A \leftarrow 0$        ▷ $template^A$: Template vector from
    retrieval system $A$
    $l^A \leftarrow \phi$
for i = 1 to m do
    for $t_i \in Q^V$ do                                       ▷ $t_i$: Query terms
        $Q_{we}^V \leftarrow \sum_{i=1}^{m} visual_{we}(t_i)/m$      ▷ $Q_{we}^V$: Visual modality word
embedding
    end for
end for
for j = 1 to n do
    for $t_j \in Q^A$ do
        $Q_{we}^A \leftarrow \sum_{j=1}^{n} audio_{we}(t_j)/n$       ▷ $Q_{we}^A$: Audio modality word
embedding
    end for
end for
for k = 1 to o do
    for $t_k \in Q^T$ do
        $Q_{we}^T \leftarrow \sum_{k=1}^{o} visual_{we}(t_k)/o$      ▷ $Q_{we}^T$: Visual modality word
embedding
    end for
end for
$Q_{we} \leftarrow [Q_{we}^V, Q_{we}^A, Q_{we}^T]$
$template^A \leftarrow DNN^A(Q_{we})$                   ▷ $DNN^A$ Deep neural network
$l^A \leftarrow \mathcal{R}^A(template^A)$ ▷ $\mathcal{R}^A$: See cluster-based retrieval in Algorithm 3
return $l^A$
```

Algorithm 2 Retrieval system B using a global embedding model

```
Input: $Q = (Q^V, Q^A, Q^T)$                                   ▷ Input query
output: $l^B$                                                  ▷ Ranked result
Initialize:
    $template^B \leftarrow 0$    ▷ $template^B$: Template vector from retrieval
    system B
    $l^B \leftarrow \phi$
for i = 1 to n do
    for $t_i \in Q$ do                                         ▷ $t_i$: Query terms
        $Q^G_{we} \leftarrow \sum_{i=1}^{n} global_{we}(t_i)/n$    ▷ $Q^G_{we}$: Global word embedding
    end for
end for
$template^B \leftarrow DNN^B(Q^G_{we})$                        ▷ $DNN^B$ Deep neural network
$l^B \leftarrow \mathcal{R}^B(template^B)$      ▷ $\mathcal{R}^B$: See cluster-based retrieval in Algorithm 3
return $l^B$
```

4.5 Cluster-Based Retrieval

There are various advantages to using clusters in multimedia data retrieval. Based on the clustering approach of the whole multimedia collection, clustering can utilize the cluster hypothesis to improve search results. When a query matches a document (video shot), we return similar shots in the cluster containing the matched one. Typically, two types of clusters are used in cluster-based retrieval. Each document (i.e., a video shot) belongs to a single cluster (i.e., a video collection) in hard clustering approaches. Alternatively, a document can belong to multiple clusters (video collections) in soft clustering approaches. We demonstrated that the soft clustering approach is more suitable for multimedia content retrieval applications. This idea is beneficial if search terms have different senses within various modalities. If we cluster videos so that similar videos appear together, inspecting a few consistent groups is often simpler than many individual videos. Within smaller clusters, we can fully compute similarities and rank videos in a standard way. Finding the nearest cluster is much faster since fewer clusters exist than videos.

In the K-means approach, we use Cosine similarity as an objective function instead of the Euclidean distance in the cluster-based retrieval. Spherical K-means is this algorithm's name, a prevalent clustering technique for high-dimensional text data. After clustering documents' space to n clusters (appropriate n is found using cluster validation), a vector, which is the output of the DNN, is compared to the clusters' centroid, and the vector is assigned to a cluster. Then, in this cluster, the top 20 vectors similar to the current vector are fetched (using Cosine similarity) and ranked according to the similarity degrees. This process is applied to the two retrieval systems presented in Fig. 4.1. Finally, two ranked lists are fused, and the top 10 results are used in the evaluation. In this work, the number of clusters (n), determined by the cluster validation, is 15 when using a global word2vec and 13 for multiple word2vec models.

Traditional clustering methods face obstacles when handling natural data that is frequently ambiguous and uncertain. For instance, the content of a video shot can be associated with different domains. Fuzzy clustering methods can manage such situations more effectively than conventional clustering algorithms. In both approaches, we compare the template vector, which is the output of the DNN model, with all centroids in C according to the Cosine similarity, then appoint the template vector to the nearest cluster. Within the cluster, the template vector is compared with all elements members of the cluster by employing Cosine similarity. Then the most similar data point is selected. Next, this data point is associated with three clusters according to the maximum membership degree. Then, the top 20 vectors are selected from each cluster according to its membership degree. In each cluster, results are ranked by membership degree, and finally, the top 20 are chosen from the list. Like the first approach (using K-means), the results of two retrieval systems are fed into the fusion module to get a final ranked list as an output of a query.

Algorithm 3 outlines the overall process, which includes the steps for retrieving a similar vector from clustered instances. In this study, we set the number of clusters c_j using a validation approach. Based on the results presented in Chapter 5, cluster-based retrieval significantly improves the overall performance of multimedia retrieval.

Algorithm 3 $\mathcal{R}$: Getting similar vectors from the clustered space

```
Input: $T$: Template vector, $C$: Clusters Centroids list, $U$: Fuzzy membership matrix
output: $l^c$                                                  ▷ Ranked result
Initialize:
    $l^c \leftarrow \phi$
    $l^{tmp} \leftarrow \phi$
    $C^l[] \leftarrow 0$
    $k \leftarrow 10$                                          ▷ $k$: Top $k$ result
for j = 1 to m do
    for $c_j \in C$ do                                         ▷ $c_j$: The centroid of a cluster
        $c_r \leftarrow x_i$   where,   $Cosine(c_j, T)$ is maximum
    end for
end for
for i = 1 to n do
    for $x_i \in c_j$ do
        $x_r \leftarrow x_i$   where,   $Cosine(x_i, T)$ is maximum      ▷ $x_r$: Any member of a cluster
    end for
end for
$C^l[] \leftarrow c_j$ according to top-3 $s$ where $s_j \in u_{rj}(j = 1, \ldots, c)$   ▷ $u_{ij}$: Any element of the matrix $U$
for $c_j \in C^l$ do
    $l^{tmp} \leftarrow l^{tmp} \cup$ top-20($x_m$)   where,   $Cosine(c_j, x_m)$ is maximum
end for
$l^c \leftarrow \{sort(l^{tmp})$   then,   fetch top-$k$($x_m$) result$\}$
return $l^c$
```

4.6 Ranked Lists Fusion

Previous fusion approaches described in Section 3 only considered ranking or relevance scores of documents while fusing ranked lists retrieved from distinct retrieval systems (engines). In this study, we proposed a novel method that treats fusion as a meta-search problem aimed at aggregating ranked lists from diverse retrieval (search) systems. In the proposed (OWA)-based fusion method, we consider multiple parameters, including document ranking, relevance scores, and the importance weights assigned to multiple retrieval systems. Each pathway denoted as A and B in Fig. 4.1 is considered as an independent retrieval system.

Given a set of separate ranked lists of video shots provided by each DNN, we first consider the performance scores of each video shot with respect to a retrieval system. Performance scores can be computed based on relevance scores computed as Cosine similarity by each retrieval system or the rank of each video shot in the ranked lists. The performance score of the video shots evaluated by each retrieval system is defined based on the ranks of the retrieved videos in the lists and their corresponding relevance scores. Since the relevance scores calculated by different retrieval systems are generally incomparable, we first apply the following normalization scores [87] on each list.

$$w^l(i) = \frac{s^l(i) - \min_{(j \in l)} s^l(j)}{\max_{(j \in l)} s^l(j) - \min_{(j \in l)} s^l(j)} \tag{4.11}$$

Suppose $s^l(i)$ denotes a real-valued relevance score assigned to video shot i in the ranked list l, then $w^l(i)$ indicates the normalized weight of video shot i in ranked list l and is calculated with Equation 4.11. In this study, we generate weights through a learning mechanism. We can deduce W with respect to the weight of normalized importance of the multimedia data retrieval systems. Algorithm 4 shows how the weights of retrieval systems are computed. Fagin et al. [88] proposed several techniques to compare two top-k ranked lists. Kendall's tau is a well-known method used for comparing two lists. This technique imposes a cost, denoted as $S(i, j) = 1$, for each pair (i, j) of distinct items where i ranked before j in one combination, and j is ranked before i in the other combination.

Since the ranked lists obtained from retrieval engines are not a permutation of all video shots, some videos may not be listed in the top-k results associated with some retrieval engines. This study uses a modified version of Kendall's tau proposed in [89]. This variant of Kendall's tau is appropriate when we are interested in comparing the top-k in one list to the top-k in another so that each pair i, j of different items are not necessarily present in the top-k results of one or both lists. The score of one (1) is attained when the top-k items of both lists are identical and in the same order, while the score of zero (0) is obtained when the top-k items of the two ranked lists have no common elements.

We use queries and the ordered list of the results provided by the ground truth for training data. To find a weight of importance for a retrieval system, we compare the ordered list obtained from each retrieval system with the ranked list provided by the ground truth data. We compute the average Kendall's tau score for each retrieval engine for all training data in the ground truth (query/video shot pairs). For a finite number of retrieval systems (d), we obtain the normalized weight w_i for the retrieval system i with:

$$w_i = \frac{\tau^i}{\sum_{k=1}^{d} \tau^k} \tag{4.12}$$

where τ^i denotes the averaged Kendall's tau score of the i^{th} retrieval system and $\sum_{i=1}^{d} w_i = 1$.

To exploit both the rank $r^m(j)$ and relevance score $s^m(j)$ of the video shot j, which is obtained from the retrieval system m, we define a local score for each video shot with respect to the retrieval system m in the top-k ranked result. We denote this score with H_j^m and calculate it as:

$$H_j^m = \begin{cases} \frac{s^m(j)}{r^m(j)}, & \text{if } j \leqslant k. \\ 0, & \text{if } j > k. \end{cases} \tag{4.13}$$

The score assigned to a video shot decreases locally as its ranking moves down in the list. To determine a global score for a video shot j with respect to all retrieval systems (m), we define an OWA operator G for video shot j within m retrieval systems as follows:

$$G(d_j) = \sum_{m=1}^{d} w_m H_j^m \tag{4.14}$$

Algorithm 4 Normalized weight calculation

```
Input: $M$: Retrieval systems, $GT$: Training data
output: $W$ =$[w_1, w_2, \ldots, w_m]$        ▷ $W$: Weight vector of retrieval systems
Initialize:
    $W \leftarrow 0$
    $\tau_{tmp}[] \leftarrow 0$
for i = 1 to m do
    for $m_i \in M$ do                                   ▷ $c_j$: The centroid of a cluster
        for j = 1 to n do
            for $(q_j, l_j) \in GT$ do   ▷ $q_j$: A query from $GT$, $l_j$: List of the relevant
results for $q_j$
                $l'_{i,j} \leftarrow retrive^{m_i}(q_j)$        ▷ $retrive^{m_i}(q_j)$: See Algorithms 1 & 2
                $\tau_{tmp}[j] \leftarrow \tau(l'_{i,j}, l_j)$        ▷ $l'_{i,j}$: List of the results returned by $i^{th}$
retrieval system
            end for   ▷ $\tau(l_1, l_2)$: Similarity of two ordered list using Kendall's tau
        end for
        $\tau^i \leftarrow \frac{1}{n}\sum_{j=1}^{n} \tau_{tmp}[j]$              ▷ $\tau^i$: Average Kendall's tau Similarity
        $w_i \leftarrow \tau^i \big/ \sum_{k=1}^{m} \tau^k$                  ▷ $w_i$: Weight of $i^{th}$ retrieval system
        $W \leftarrow W \cup w_i$
        $\tau_{tmp}[] \leftarrow 0$
    end for
end for
return $W$
```

The operator G is applied to all retrieval systems' top-k video shots to get the final ranking. Then the videos are arranged according to the global scores in descending order. Algorithm 5 outlines the steps to get the final ranked results. Table 4.1 shows an example run for fusing two ranked lists where each list contains ten video shots. Each d is input for Algorithm 5 in this table. After computing the global scores for each video shot, the list displayed in Table 4.2 presents the final ranked results.

Algorithm 5 Calculating the final ranked list (l^{final}) using OWA fusion

```
Input: $L = [l^1, l^2, \ldots, l^m]$: Ranked lists, $W$: Importance score of retrieval systems
output: $l^{final}$                                   ▷ $l^{final}$: Final ranked list
Initialize:
    $H \leftarrow 0$
    $k \leftarrow 10$                                  ▷ $k$: Top $k$ result
for i = 1 to m do
    for $d_j \in l^i$ do                               ▷ $d$: Document (video shot) in the list
        $w^{l^i}(d_j)$                                 ▷ $w^{l^i}$: Relevance score normalization
        if $r^i(d_j) \leqslant k$ then
            $H^i_j \leftarrow \frac{s^i(j)}{r^i(j)}$    ▷ $s$: Relevance score of $d$, $r$: Rank of $d$
        else
            $H^i_j \leftarrow 0$                       ▷ $H$: local score of $d_j$ in $l^i$
        end if
    end for
end for
for $d_j \in l^i$ do
    $G(d_j) \leftarrow \sum_{u=1}^{m} w_u H^u_j$       ▷ $G(d_j)$: Global score of $d$
end for
$l^{final} \leftarrow sort(d_j)$ according to $G(d_j)$
return $l^{final}$
```

Table 4.1: The global score of video shots.

m_1	w_1= **0.59**									
d	d_3	d_7	d_8	d_{11}	d_{19}	d_{18}	d_9	d_2	d_{10}	d_{13}
r	1	2	3	4	5	6	7	8	9	10
s	0.81	0.80	0.72	0.7	0.69	0.68	0.65	0.64	0.6	0.59
w^l	1	0.95	0.59	0.5	0.45	0.41	0.27	0.23	0.05	0
H	1	0.475	0.197	0.125	0.09	0.068	0.038	0.029	0.006	0
$w_1 \times H$	0.59	0.28025	0.11623	0.07375	0.0531	0.04012	0.02242	0.01711	0.00354	0
m_2	w_2= **0.41**									
d	d_2	d_{19}	d_{18}	d_3	d_{11}	d_7	d_{21}	d_{12}	d_1	d_{10}
r	1	2	3	4	5	6	7	8	9	10
s	0.89	0.86	0.81	0.79	0.75	0.71	0.69	0.68	0.61	0.58
w^l	1	0.9	0.74	0.68	0.55	0.42	0.35	0.32	0.097	0
H	1	0.45	0.247	0.17	0.11	0.07	0.05	0.04	0.011	0
$w_2 \times H$	0.41	0.1845	0.10127	0.0697	0.0451	0.0287	0.0205	0.0164	0.00451	0

Table 4.2: Final results ranking.

d_j	d_1	d_2	d_3	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{18}	d_{19}	d_{21}
$G(d_j)$	0.005	0.427	0.660	0.309	0.116	0.022	0.004	0.119	0.016	0	0.141	0.238	0.021
d'_j	d_3	d_2	d_7	d_{19}	d_{18}	d_{11}	d_8	d_9	d_{21}	d_{12}	d_1	d_{10}	d_{13}
l^{final}	0.660	0.427	0.309	0.238	0.141	0.119	0.116	0.022	0.021	0.016	0.005	0.004	0

4.7 Ranking With Siamese Network

Ranking in multimedia information retrieval is an essential technique to enhance the effectiveness and precision of search results in multimedia content. The initial ranking produced by search algorithms may not always meet the user's needs, often requiring further refinement, and this is where re-ranking steps in. Re-ranking involves accurately re-evaluating and revising the order of retrieved multimedia items to align them with the query better. One of the primary objectives of re-ranking is to elevate the most relevant multimedia content to the top of the search results. This involves considering various factors, such as content similarity and contextual relevance. Machine learning models, like Siamese networks, can assess the similarity between multimedia items, ensuring they are closely aligned with the query.

Siamese networks are well-suited for ranking applications because they explicitly learn to measure similarity or dissimilarity between pairs of data points. These networks are versatile tools that can be adapted to various tasks where learning the similarity or dissimilarity between data pairs is crucial. Their ability to learn meaningful embeddings makes them valuable in tasks that require understanding the relationships between data points, like ranking. The choice of the loss function in Siamese networks depends on the specific application and the problem at hand. Contrastive loss and triplet loss are particularly common choices due to their effectiveness in learning embeddings that represent similarity. As a result, we employed these loss functions to re-rank the results.

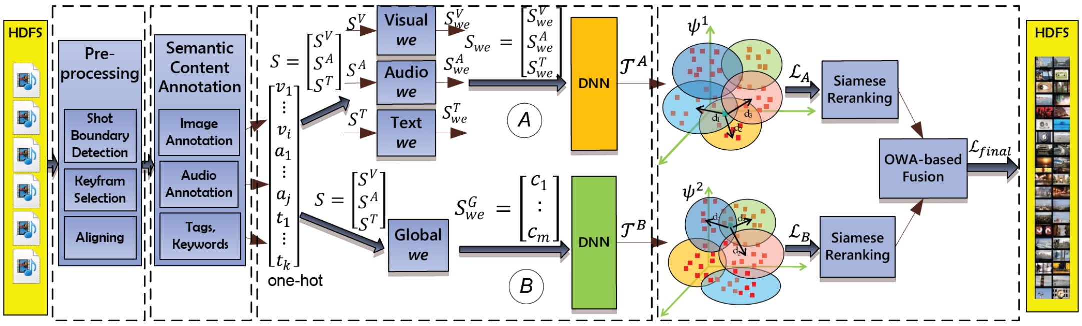


Figure 4.6: The overall architectural design of the proposed approach with the Siamese re-ranking layer.

This study ranks the results according to the sorted list returned by clustered space. In addition to this approach, we aim to leverage a supplementary ranking algorithm to re-rank the results through a learning mechanism. In this setup, first, we replace the clustered retrieval approach with a trained Siamese network. Secondly, we use both clustering-based and Siamese ranking methods. Therefore, the results pass through two ranking algorithms, and the final sorted list is fed to the fusion algorithm as depicted in Fig. 4.6. We employ queries and relevant items (positive samples) from the ground truth dataset to train the Siamese network. To prepare irrelevant items (negative samples), we select random items from the remaining non-relevant list. We choose a similar number of positive and negative samples for each query to provide balanced data. The query itself is used as an anchor in the training steps. Since Contrastive loss and Triplet loss outperform in ranking tasks, we leverage them in the Siamese network.

CHAPTER 5

EXPERIMENTS

In this section, we demonstrate the effectiveness of the proposed system for retrieving multimodal multimedia data. First, we introduce datasets and define performance metrics. Next, we describe the parameter selection methods and the network architecture. After that, we present the experimental results for different configurations using our approach. After that, we demonstrate the proposed system's effectiveness using some statistical tests. Ultimately, we evaluate our approach by comparing it with state-of-the-art studies.

5.1 Datasets

To verify the proposed method's effectiveness, we use three data sets: the NTV News Archive dataset [33] [90], a subset of the Flickr YFCC100m dataset [91], and the complete version of Flickr, which we referred to as YFCC100m*. Furthermore, we conduct experiments on six widely-used benchmark multimodal datasets: NUS-WIDE [92], XMediaNet [93], Flickr (YFCC100m) [91], Pascal [94], Wikipedia [95], and MS COCO [96]. Each dataset is randomly separated into three parts: 80% of the data is used for training, 10% for validation, and 10% for testing. We summarize some specifications of the smallest and largest datasets in Table 5.1.

5.2 Implementation

In this study, we use Deeplearning4j, TensorFlow, HDFS, and Spark engine. All training and testing are carried out in a big data cluster. Source codes are written in

Table 5.1: Datasets specifications.

Specifications	**NTV-News**	**YFCC100m**	**YFCC100m***
Number of videos	1500	7750	470479
Length (hour)	6	103	6081
Number of visual concepts	49	94	1261
Number of audial concepts	17	28	369
Number of textual concepts	100	151	1414
Total number of concepts	166	273	3034

Java and Python.

5.3 Evaluation Metrics

To assess the results, we employ two metrics. The Mean Average Precision (mAP) is a commonly used metric for evaluating the relevancy quality in a ranked list. Precision@k or P@k computes the relevant percent (%) in the top-k results for a given rank position. This computation ignores the video shots ranked lower than k and max-k. To calculate this metric:

1. Consider the rank position of each relevant video shots $d_1, d_2, \ldots, d_r$
2. Compute P@k for each $d_1, d_2, \ldots, d_r$
3. Average Precision (AP) is the average of P@k
4. Take the average of AP across multiple queries as follows.

$$mAP = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{n_j}\sum_{i=1}^{n_j}P(d_i) \tag{5.1}$$

Where n_j is the number of relevant video shots for query j, q specifies the number of queries, and $P(d_i)$ represents the precision at the i^{th} relevant video shot.

The second metric we employ to evaluate the quality of rankings is Normalized Discounted Cumulative Gain (NDCG). This metric is mainly used in information retrieval problems, such as measuring the effectiveness of the search engine by ranking the documents it displays according to their relevance to search keywords. To calculate this metric, we need to compute some intermediate values. Cumulative Gain (CG) is a middle value that does not include the result's position (rank) in the effectiveness of a result list. We calculate GC by summing up the relevance values of video shots in the result set.

$$GC = \sum_{pos=1}^{n} relevance_{pos} \tag{5.2}$$

Since GC does not reward relevant results that appear higher in the ranked list, we must discount results that appear in low ranks using Discounted Cumulative Gain (DCG). A common method for doing this is:

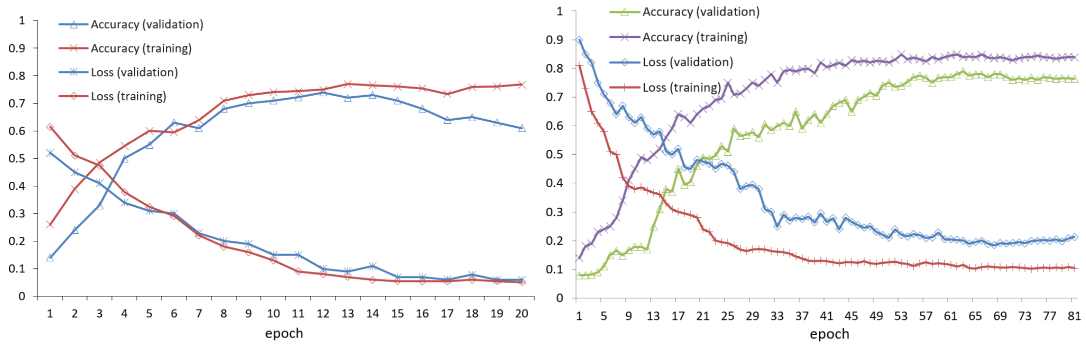
$$DCG = \sum_{pos=1}^{n} \frac{relevance_{pos}}{\ln(pos+1)} \tag{5.3}$$

The final stage is normalized for different queries to scale the results based on the best (ideal DCG or $iDCG$).

$$NDCG_{pos} = \frac{DCG_{pos}}{iDCG} \tag{5.4}$$

5.4 Network Architecture and Parameter Selection

The generative deep neural network architectures proposed in this study consist of 3 hidden layers. The number of units in the output layer is selected to match the output of the word embedding models. We use $ReLU$ as an activation function and the Cosine loss function as an objective function to minimize the dot product between the output vectors (document) and the ground truth vector (query). We start with 0.01 as a learning rate and use the Adaptive Moment Estimation (Adam) optimizer for weight updates and $L2$ regularization. Additionally, we leverage dropout to reduce overfitting. We use model accuracy and loss to get appropriate epoch numbers 12 for NTV-News and 68 for the YFCCM100M dataset. We deduce these values according to the observations presented in Fig. 5.1. Table 5.2 presents a comprehensive overview of the layers and parameters associated with the proposed models for retrieval systems A and B.



(a)

(b)

Figure 5.1: The epoch vs model accuracy and loss on train and validation datasets using (a) NTV, (b) YFCC100M (Flickr).

Table 5.2: Layers and parameters of the proposed models.

Parameters	**Retrieval system** A	**Retrieval system** B
Input Layer size	3072	1024
Hidden unites	3	3
Hidden-1 size	6144	2048
Hidden-2 size	12288	4096
Hidden-3 size	6144	2048
Output Layer size	3072	1024
Activation function	ReLU	ReLU
Optimizer	Adam	Adam
Weights initialization	He-et-al	He-et-al
Lost function	Cosine loss	Cosine loss
Regularization	L2, Dropout	L2, Dropout
Dropout rate	0.2	0.2
Learning rate	0.01	0.01
Batch Size	128	128
Epochs	68	59

5.5 Analysis of Clusters

To determine an appropriate number of clusters for each configuration, we calculate mAP@10 across various cluster sizes. We identify the optimal number of clusters through cluster validation and Silhouette analysis, which results in the highest mAP. In the most effective setup that utilizes AAFCM, the number of clusters is 14, 129, 6231 for NTV-News, YFCC100m*, and YFCC100m datasets.

5.6 Analysis of Missing Modality

Supervised methods achieve the highest performance by leveraging semantic meaning among the numerous multimodal multimedia retrieval techniques. However, in practical scenarios, data is not always complete with labels and full multimodal information, making applying these techniques challenging. We encounter situations where specific modalities are missing or incomplete in real-world use cases. Researchers in [97] proposed a knowledge distillation framework for alleviating this issue. The approach leveraged additional information from all modalities while avoiding any noise associated with the extra data. Another study [98] proposed a cross-partial multi-view network to handle missing or incomplete modalities and views. To address the same issue, [99] proposed an iterative data augmentation approach for emotion recognition.

The architecture proposed in this research can address multimodal multimedia retrieval tasks, even when dealing with missing labels (concepts) and modalities. Our study employs two distinct embedding models to establish correlations among concepts within and across different modalities. To elaborate, the data in each modality is embedded into a shared feature space, and their correlations are maximized collectively. Even in cases where labels or concepts are absent in specific modalities, the proposed method can still maximize the correlation between the available data or labels. In specific scenarios, the absence of modalities can be addressed by incorporating information from other modalities through fusion. For instance, if the visual modality is missing, textual information can be leveraged to provide context and enhance the dataset. Our architecture employs two trained models to capture inter-modal and intra-modal correlations and a common subspace. This design enables adaptability to missing modalities by integrating complementary information from other modalities. Lastly, the clustering strategy in multimodal retrieval can be utilized for alleviating outliers. Nonetheless, our solution may generate uncertain concepts for the missing modalities in incomplete multiple modalities.

5.7 Results and Discussions

This study focuses on two main methods to test the effectiveness of the proposed architecture. First, we presented various techniques to improve the performance of the baseline study [33] and illustrated the results in Fig 5.2. Later, we chose the best setup from the first test method, ELMo+DNN+AAFCM+OWA, and carried out tests on various well-known multimodal datasets. After that, we compared the retrieval performance of our approach with different state-of-the-art studies and presented the results in Table 11. Additionally, we conducted a query performance test on various datasets and presented the results in Table 10. We select the top 10 results in all test setups and compute the mAP and $NDCG$ to evaluate the performance.

We use different configurations in the experimental tests. To represent a video as a vector, we denote one-hot encoding by ‘one-hot‘, single word embedding model (word2vec or ELMo) by ’Single w2v’, used in retrieval system B. When we train different word embedding models (word2vec or ELMo) for each modality, we denote this setup by ‘Multi w2v‘, used in retrieval system A. We utilize the proposed DNN model and Canonical Correlation Analysis (CCA) to learn contextual association among modalities. In the cluster-based retrieval, we denoted different approaches by K-means (KM), Fuzzy C-Means (FCM), DBSCAN, and AAFCM. This study utilizes different fusion techniques for A and B retrieval systems. We define these methods by ComboMax, Condorcet, and OWA. Consequently, when we employ a fusion approach in test setups, it implies that we use both ‘Single w2v‘ and ‘Multi w2v‘, which we refer to as ‘W2V‘. This study's baseline is the experimental result presented in [16]. This result is shown with the ‘CCA+FCM‘, which indicates an approach using Canonical Correlation Analysis and FCM. We show the experimental results for various test setups in Fig. 5.2.

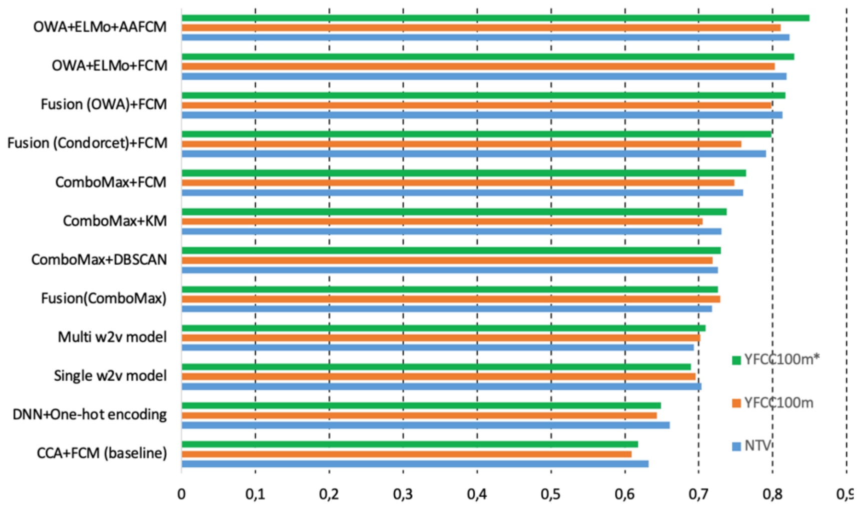


Figure 5.2: NDCG@10 for three datasets.

We initially prepared a test configuration with one-hot encoding to train deep neural networks. In this configuration, we train a single DNN so that the values of the input and output layers are one-hot vectors representing queries and shots. This configuration aims to investigate the impact of word embedding and one-hot encoding on the performance of the proposed system. The result of this configuration is displayed with the ‘DNN+one-hot encoding‘ label. We observe that the single w2v model outperforms the one-hot encoding approach. The results of the remaining test configurations are compared to the baseline. The revised configuration, labeled as ‘Multi w2v model‘, is associated with a setup where separate word embedding models are trained for each modality. A comparison with the initial ‘Single w2v model‘ configuration underscores the significance of employing distinct embedding models for each modality. Another configuration that fuses the result of two previous setups using ComboMAX is presented with ‘Fusion(ComboMax)‘. This configuration suggests that incorporating fusion enhances the overall mAP value in each scenario. In the subsequent configurations, we explore different clustering methods. Our findings indicate that FCM clustering not only raises the MAP value but also surpasses the performance of K-means clustering. Upon consideration of the Condorcet method and the OWA fusion approach proposed in this study, the latter demonstrates superior performance in scenarios denoted by ‘Fusion(Condorcet)+FCM‘ and ‘Fusion(OWA)+FCM‘. In the prior configurations, the word2vec (w2v) approach is employed as the word embedding method. However, in the scenario denoted by ‘OWA+ELMo+FCM‘, it is evident that substituting ELMo for w2v has a positive impact on the overall performance.

To assess the efficacy of the template generation layer, we exclude the DNN component and develop a new test setup. In this configuration, we utilize ELMo for word embedding, AAFCM as the clustering methodology, and OWA for fusion. This particular setup is denoted by "ELMO+AAFCM+OWA".

In conclusion, it is observed that employing AAFCM surpasses the performance of FCM. Specifically, the configuration that utilizes ‘OWA+ELMo+AAFCM‘ outperforms all other setup configurations.

In addition to the NDCG@10 metric, the precision-recall curve is plotted in Fig. 5.3 for the Flickr YFCC100m dataset. The area under the curve shows the performance of different configuration setups. The rightmost curve illustrates the optimal outcome achieved by employing ELMo word embedding, AAFCM clustering, and OWA fusion. These results are consistent with the NDCG@10 values reported in Fig. 5.2. Table 5.3 presents an ablation study by removing each component from the proposed architecture and showing the overall system performance. Eliminating specific components such as distributed word representation, clustering, and fusion aims to

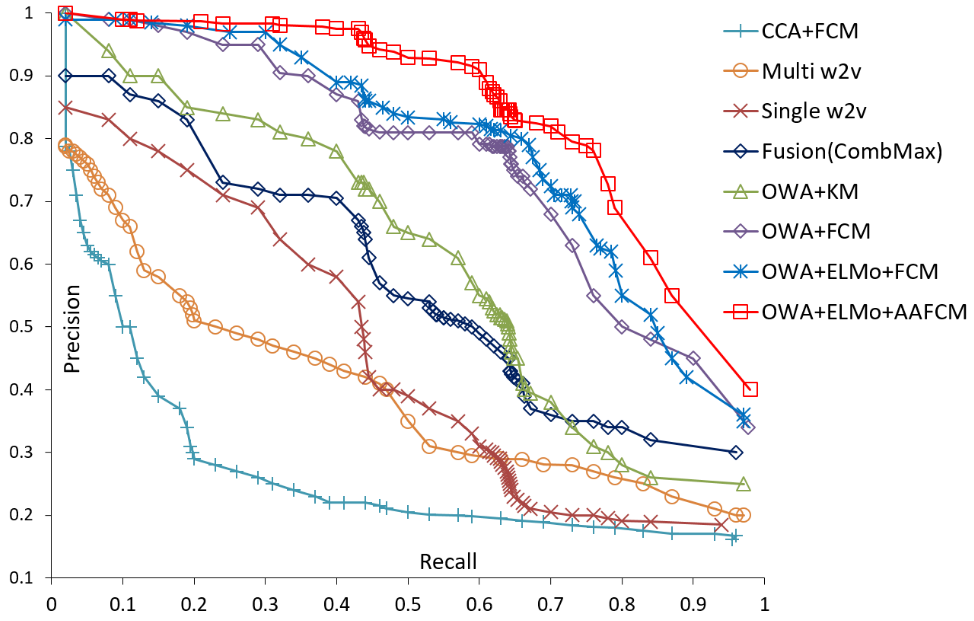


Figure 5.3: The Precision-Recall curve per test configuration on YFCC100m dataset. PR is maximized in the upper right corner.

understand each component's contribution to the overall system. Bold results are the best ones, indicating the ablated strategy is the most important. The current study leveraged various score-based and rank-based fusion methods to find the best strategy. Additionally, We propose an OWA-based method to fuse the ranked lists according to the performance scores of the retrieval systems. The retrieval performance of different ranked lists fusion methods is given in Table 5.4. We compute mAP and NDCG metrics for each fusion method. As presented in this table, the proposed OWA-based fusion obtains the best results compared to other approaches considering both metrics. Since the proposed method combines score-based, rank-based, and performance scores of the retrieval systems, it outperforms other techniques by 3.7%. The most similar works to this study are selected to compare the results with state-of-the-art. Precisely, we assess the effectiveness of the proposed approach by comparing it with the nine most similar methods. However, minor variations in architecture and test setups may exist. Table 5.5 presents various features each study supports. We consider the following features: Supported modalities, Fusion, Clustering, Scalability, Han-

Table 5.3: Ablation study on each component proposed in our work, with NDCG@10 employed as the evaluation metric.

Test Setup	**NTV-News**	**YFCC100m**	**YFCC100m***
CCA+FCM (**baseline**)	0.632	0.609	0.618
One-hot+DNN	0.661	0.643	0.649
Single w2v+DNN	0.704	0.696	0.689
Multi w2v+DNN	0.693	0.702	0.709
W2V+DNN+ComboMax	0.718	0.729	0.726
W2V+DNN+KM+ComboMax	0.731	0.705	0.738
W2V+DNN+DBSCAN+ComboMax	0.726	0.719	0.730
W2V+DNN+FCM+ComboMax	0.760	0.748	0.764
W2V+DNN+FCM+Condorcet	0.791	0.758	0.798
W2V+DNN+FCM+OWA	0.813	0.798	0.817
ELMo+DNN+FCM+OWA	0.819	0.803	0.829
ELMo+DNN+AAFCM+OWA	**0.823**	**0.811**	**0.850**

Table 5.4: MAP and NDCG @10 for different ranked list fusion (NTV-News).

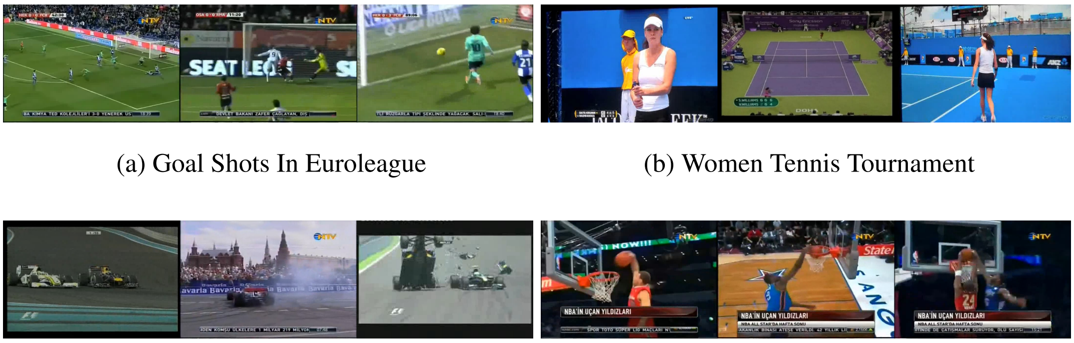
Test Setup	**mAP (%)**	**NDCG**
CombMAX	77.03	0.71
CombMIN	76.72	0.73
CombSUM	78.81	0.75
CombMNZ	80.24	0.78
CombANZ	78.12	0.75
Borda count	77.48	0.76
Condorcet	81.07	0.79
OWA	**84.11**	**0.81**

dling noisy modalities, Using big data technologies, and leveraging deep learning. The proposed study supports three modalities and utilizes all other features. We use well-known multimodal datasets in table 5.6 and compare the mAP@10 result with state-of-the-art studies. The best mAP scores are illustrated in **bold**, and the second best scores are presented with underline. Some studies have not conducted tests on specific datasets that We mark with the "×" symbol. The mAP values in Table 5.6 indicate that our approach outperforms other methods in most datasets. The proposed study (ELMo+DNN+AAFCM+OWA) gets the second and third largest mAP@10 in the remaining datasets. In this study, effectiveness is directly associated with the size of the datasets. Considering this fact, mAP@10 got weaker performance in Wikipedia and MS COCO datasets. We illustrate frames of retried videos for some search queries in Fig. 5.4.

This study ranks the results according to the sorted list returned by clustered space. In addition to this approach, we utilized an extra ranking algorithm to re-rank the results through a learning mechanism. In this setup, first, we replace the clustered retrieval approach with a Siamese network trained using Contrastive and Triplet loss. Secondly, we use clustering-based retrieval to rank the results and Siamese network methods to re-rank the initial list. Using four methods, we use various multimodal datasets, compare the mAP@10 value, and present the results in Table 5.7. The high-

Table 5.5: Qualitative features comparison. We use ✓ to denote that the specific study supports the selected feature. Unsupported features are marked with ×.

Methods	**Visual Modality**	**Audio Modality**	**Textual Modality**	**Fusion**	**Clustering**	**Scalability**	**Noisy Modalities**	**Big Data Tech.**	**Deep Models**	**Handling Uncertainty**	**Relevance Feedback**
CCA [33]	✓	✓	✓	×	✓	×	×	×	×	×	×
DCCA [41]	✓	✓	×	×	×	×	✓	×	✓	×	×
SDML [23]	✓	✓	✓	×	×	✓	×	✓	✓	×	×
MMSAE [28]	✓	×	✓	×	×	✓	×	×	✓	×	×
SML-CCA [42]	✓	×	✓	×	×	×	✓	×	✓	×	×
MCCN [43]	✓	×	✓	×	✓	×	×	×	✓	×	×
CM-GANs [39]	✓	×	✓	✓	×	✓	✓	×	✓	✓	×
DMTL [44]	✓	×	✓	✓	×	×	×	×	✓	×	×
MAN [45]	✓	✓	✓	✓	×	✓	✓	×	✓	×	✓
Our study	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×



(a) Goal Shots In Euroleague

(b) Women Tennis Tournament

(c) F1 Car Accident

(d) NBA Slam Dunk

Figure 5.4: Illustration of frames extracted from retried shots for specific search queries.

Table 5.6: Performance comparison with other studies. The best mAP scores are illustrated in **bold**, and the second best scores are presented with underline. The "×" symbol indicates that the study has not conducted any experiments on the selected dataset.

Methods	**NTV-News**	**NUS-WIDE**	**XMediaNe**	**Flickr**	**Pascal**	**Wikipedia**	**MS COCO**
CCA[33]	0.614	0.531	0.548	×	×	0.465	0.403
DCCA [41]	×	0.717	0.656	0.645	0.406	0.486	0.415
SDML [23]	×	0.697	0.609	×	0.680	0.505	**0.823**
MMSAE [28]	×	0.826	×	0.751	0.449	0.535	×
SML-CCA [42]	×	0.830	×	0.709	0.489	×	×
MCCN [43]	×	×	0.742	×	0.707	0.520	×
CM-GANs [39]	×	0.685	0.586	×	0.641	0.548	×
DMTL [44]	×	0.645	0.718	×	0.634	0.552	×
MAN [45]	×	×	0.571	×	0.696	0.586	×
Our study	**0.790**	**0.873**	**0.805**	**0.879**	**0.730**	**0.647**	0.739

est mAP score is shown in **bold**, and the lowest is presented with underline. The baseline method refers to ELMo+DNN+AAFCM+OWA in Table 5.3, which employs clustering-based ranking. We replaced the cluster-based retrieval with a Siamese network in the following two approaches. In each iteration, we leverage different loss functions for ranking the results. In the last approach, initial results are ranked using cluster-based retrieval. Subsequently, the initial outcomes undergo further refinement via a Siamese network utilizing the Triplet loss. This network aims to re-rank the list and generate input data suitable for the fusion algorithm. As we analyze the outcomes, it becomes evident that the most optimal performance is achieved by combining clustering-based ranking with a Siamese (Triplet loss) for re-ranking. Notably, this re-ranking process has led to a significant enhancement in overall precision. The results suggest that employing the Contrastive loss yields suboptimal performance.

Table 5.7: Retrieval performance comparison according to the ranking method.

	mAP@10 for different Datasets						
Methods	**NTV-News**	**NUS-WIDE**	**XMediaNet**	**Flickr**	**Pascal**	**Wikipedia**	**MS COCO**
AAFCM	0.773	0.841	0.781	0.862	0.724	0.545	0.730
Triplet	0.760	0.830	0.790	0.846	0.705	0.611	0.713
Contrastive	0.697	0.790	0.697	0.792	0.681	0.567	0.674
AAFCM+Contrastive	0.784	0.812	0.704	0.770	0.719	0.623	**0.741**
AAFCM+Triplet	**0.790**	**0.873**	**0.805**	**0.879**	**0.730**	**0.647**	0.739

5.8 Model Simplification

In multimedia information retrieval, the effectiveness of retrieval systems heavily relies on integrating various components and techniques to optimize performance. This study undertakes a comprehensive stepwise ablation analysis to simplify the model and systematically assess the impact of individual components on the effectiveness of a proposed multimodal multimedia retrieval architecture. The architecture proposed in this study presents an overall approach with multiple components. We aim to evaluate each component's contribution to the retrieval system's overall performance through experiments and evaluations. The baseline architecture is initially introduced, highlighting its foundational components. Subsequently, the stepwise ablation process is defined, wherein specific components are systematically removed or replaced to assess their individual impact on retrieval performance. These components include word embedding methods, clustering algorithms, fusion techniques, and template generation layers.

The study employs three distinct word embedding approaches: one-hot encoding, word2vec (w2v), and ELMo. Clustering is performed using K-means (KM), Fuzzy C-Means (FCM), DBSCAN, and Alternative Adaptive Fuzzy C-Means (AAFCM). Various fusion techniques are employed, including ComboMax, Condorcet, and OWA. When fusion approaches are utilized in test setups denoted by A and B in Fig. 4.1, it indicates the presence of two retrieval systems.

When employing the "Multi w2v" configuration, distinct embedding models are trained for each modality, denoted as the "Local" model. Conversely, when modalities are stacked together as a single vector, referred to as the "Single w2v" configuration, only one embedding model, known as the "Global" model, is trained. Moreover, configurations incorporating template generation are designated with DNN. The baseline study presented in [33] employs CCA for finding the correlation between two modalities and leverages FCM for retrieval and ranking. We denote this setup with "CCA+FCM"

In "One-hot+Cosine", we leverage One-hot encoding and use Cosine similarity to retrieve and rank the results. Fig. 5.5 presents the simplified diagram of this setup. In "One-hot+DNN+Cosine", we leverage one-hot encoding and use the template vector generation layer. Additionally, we employ Cosine similarity for ranking the results. Fig. 5.6 presents the simplified diagram of this setup. In "Single w2v+DNN+Cosine",

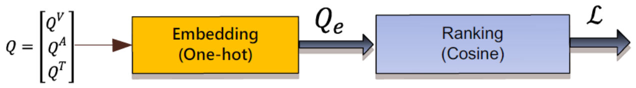


Figure 5.5: One-hot+Cosine.

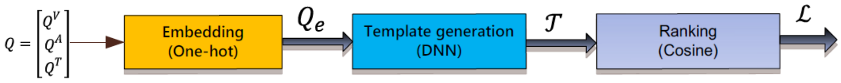


Figure 5.6: One-hot+DNN+Cosine.

we employ global w2v embedding and use the template vector generation layer. Additionally, we employ Cosine similarity for ranking the results. Fig. 5.7 presents the simplified diagram of this setup. In "Multi w2v+Cosine", we employ local w2v embedding and use the template vector generation layer. Additionally, we employ Cosine similarity for ranking the results. Fig. 5.8 presents the simplified diagram of this setup. In "W2V+DNN+ComboMax+Cosine", we employ w2v embedding and use the template vector generation layer. Additionally, we employ Cosine similarity for ranking and ComboMax to fuse the results. Fig. 5.9 presents the simplified diagram of this setup. In "W2V+DNN+KM+ComboMax", we employ w2v embedding and use the template vector generation layer. Additionally, we leverage KM clustering for retrieval and ranking. The comboMax technique is used to fuse the results. Fig. 5.10 presents the simplified diagram of this setup. In "W2V+DNN+DBSCAN+ComboMax", we employ w2v embedding and use the template vector generation layer. Additionally, we leverage DBSCAN clustering for retrieval and ranking. The comboMax technique is used to fuse the results. Fig. 5.11 presents the simplified diagram of this setup. In "W2V+DNN+FCM+ComboMax", we employ w2v embedding and use the template vector generation layer. Additionally, we leverage FCM clustering for retrieval and ranking. The comboMax technique is used to fuse the results. Fig. 5.12 presents the simplified diagram of this

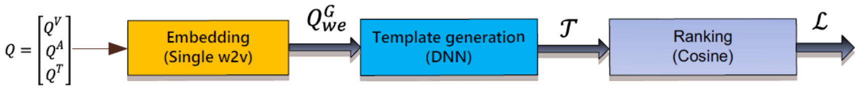


Figure 5.7: Single w2v+DNN+Cosine.

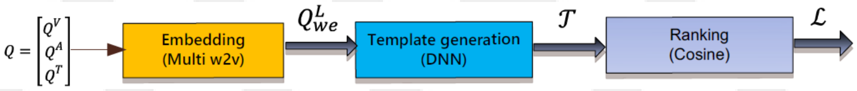


Figure 5.8: Multi w2v+Cosine.

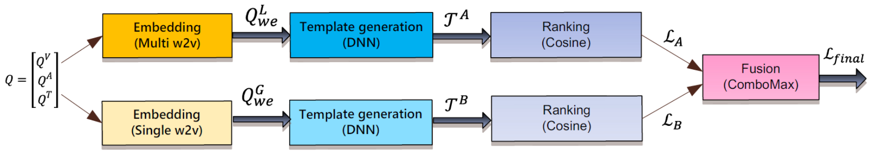


Figure 5.9: W2V+DNN+ComboMax+Cosine.

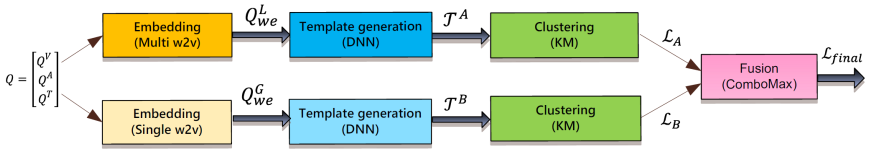


Figure 5.10: W2V+DNN+KM+ComboMax.

setup. In "W2V+DNN+FCM+Condorcet", we employ w2v embedding and use the template vector generation layer. Additionally, we leverage FCM clustering for retrieval and ranking. The Condorcet technique is used to fuse the results. Fig. 5.13 presents the simplified diagram of this setup. In "W2V+DNN+FCM+OWA", we employ w2v embedding and use the template vector generation layer. Additionally, we leverage FCM clustering for retrieval and ranking. The OWA technique is used to fuse the results. Fig. 5.14 presents the simplified diagram of this setup. In "ELMo+DNN+FCM+OWA", we employ ELMo embedding and use the template vector generation layer. Additionally, we leverage FCM clustering for retrieval and ranking. The OWA technique is used to fuse the results. Fig. 5.15 presents the simplified diagram of this setup. In "ELMo+DNN+AAFCM+OWA", we employ ELMo embedding and use the template vector generation layer. Additionally, we leverage AAFCM clustering for retrieval and ranking. The OWA technique is used to fuse the results. Fig. 5.16 presents the simplified diagram of this setup. In "ELMo+AAFCM+OWA", we employ ELMo embedding and leverage AAFCM clustering for retrieval and ranking. The OWA technique is used to fuse the results. Fig. 5.17 presents the simplified diagram of this setup. In "ELMo+DNN+Triplet+OWA", we employ ELMo embedding and use the template vector generation layer. Additionally, we leverage the Siamese network with Triplet loss for ranking the results. The OWA technique is used to fuse the results. Fig. 5.18 presents the simplified diagram of this setup. In "ELMo+DNN+Contrastive+OWA", we employ ELMo embedding and use the template vector generation layer. Additionally, we lever-

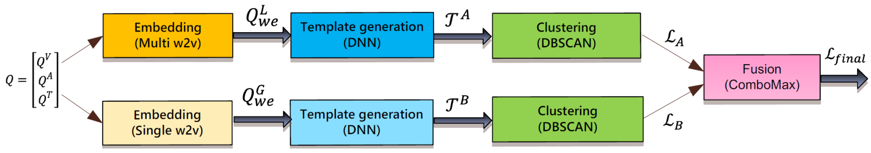


Figure 5.11: W2V+DNN+DBSCAN+ComboMax.

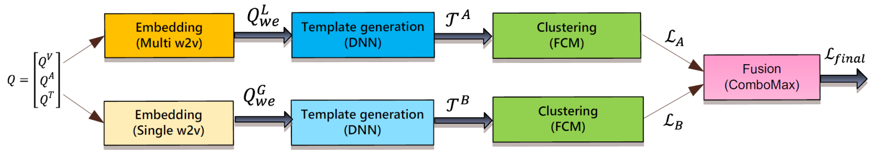


Figure 5.12: W2V+DNN+FCM+ComboMax.

age the Siamese network with Contrastive loss for ranking the results. The OWA technique is used to fuse the results. Fig. 5.19 presents the simplified diagram of this setup. In "ELMo+DNN+AAFCM+Contrastive+OWA", we employ ELMo embedding and use the template vector generation layer. For retrieving and initial ranking, AAFCM clustering is used. Additionally, we leverage the Siamese network with Contrastive loss for re-ranking the results. The OWA technique is used to fuse the results. Fig. 5.20 presents the simplified diagram of this setup. In "ELMo+DNN+AAFCM+Triplet+OWA", we employ ELMo embedding and use the template vector generation layer. For retrieving and initial ranking, AAFCM clustering is used. Additionally, we leverage the Siamese network with Triplet loss for re-ranking the results. The OWA technique is used to fuse the results. Fig. 5.21 presents the simplified diagram of this setup. The following table 5.8 outlines the results of the stepwise ablation study. Each step is compared with the preceding one, and the percentage change in NDCG@10 is indicated. An upward arrow denotes an increase, while a downward arrow signifies a decrease.

Table 5.8: Stepwise ablation study on each component using YFCC100m dataset. Each step is compared with the preceding one, indicating the percentage change in NDCG@10 value.

Test Setup	**NDCG@10**	**Impact(%)**
CCA+FCM (**baseline**)	0.618	-
One-hot+Cosine	0.503	18.6 ↓
One-hot+DNN+Cosine	0.649	5.0 ↑
Single w2v+DNN+Cosine	0.689	11.5 ↑
Multi w2v+DNN+Cosine	0.709	14.7 ↑
W2V+DNN+ComboMax	0.726	17.5 ↑
W2V+DNN+KM+ComboMax	0.738	19.4 ↑
W2V+DNN+DBSCAN+ComboMax	0.730	18.1 ↑
W2V+DNN+FCM+ComboMax	0.764	23.6 ↑
W2V+DNN+FCM+Condorcet	0.798	29.1 ↑
W2V+DNN+FCM+OWA	0.817	32.2 ↑
ELMo+DNN+FCM+OWA	0.829	34.1 ↑
ELMo+DNN+AAFCM+OWA	0.850	37.5 ↑
ELMo+AAFCM+OWA	0.683	10.5 ↑
ELMo+DNN+Triplet+OWA	0.812	31.4 ↑
ELMo+DNN+Contrastive+OWA	0.779	26.0 ↑
ELMo+DNN+AAFCM+Contrastive+OWA	0.857	38.7 ↑
ELMo+DNN+AAFCM+Triplet+OWA	0.863	39.6 ↑

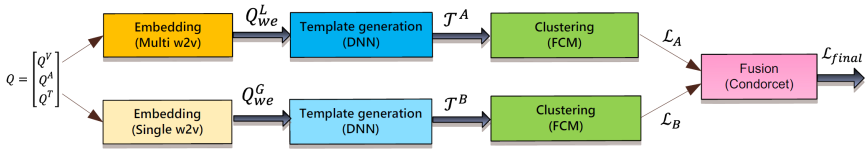


Figure 5.13: W2V+DNN+FCM+Condorcet.

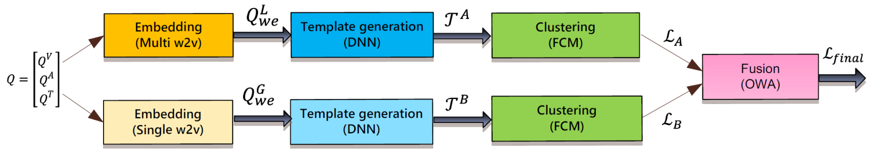


Figure 5.14: W2V+DNN+FCM+OWA.

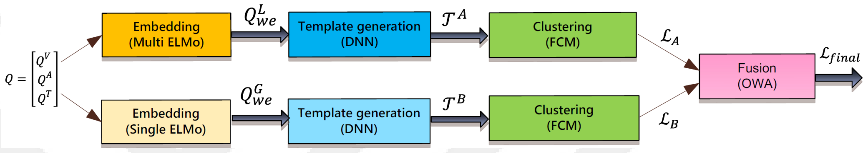


Figure 5.15: ELMo+DNN+FCM+OWA.

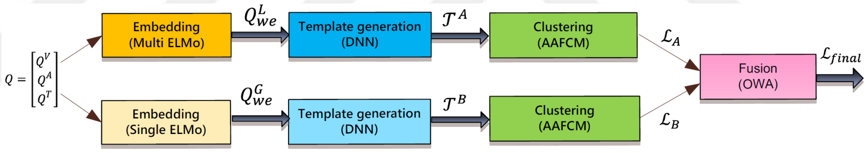


Figure 5.16: ELMo+DNN+AAFCM+OWA.

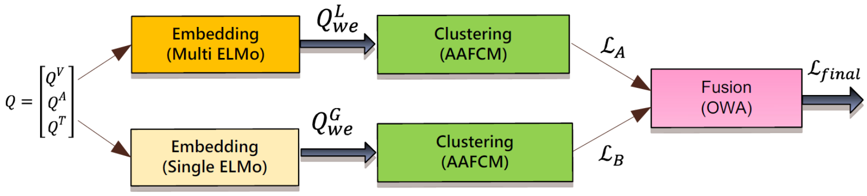


Figure 5.17: ELMo+AAFCM+OWA.

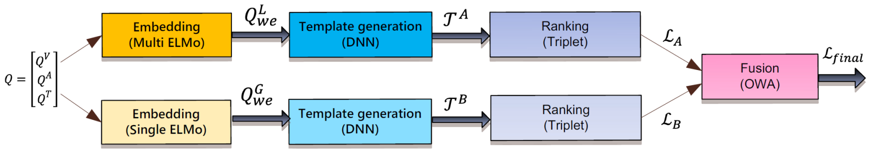


Figure 5.18: ELMo+DNN+Triplet+OWA.

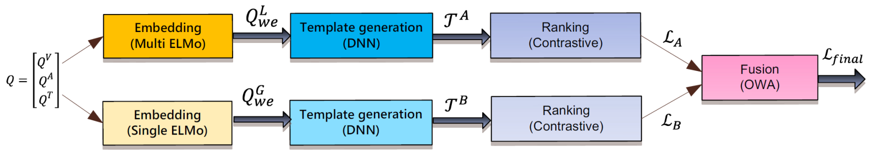


Figure 5.19: ELMo+DNN+Contrastive+OWA.

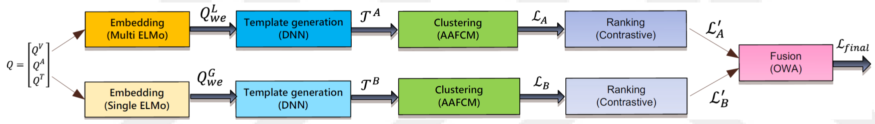


Figure 5.20: ELMo+DNN+AAFCM+Contrastive+OWA.

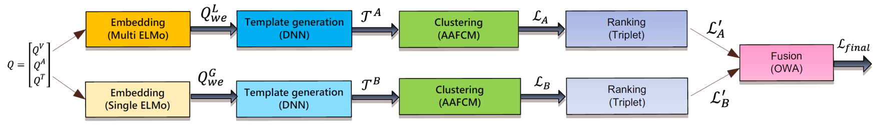


Figure 5.21: ELMo+DNN+AAFCM+Triplet+OWA.

5.9 Statistical Analysis

To check if the results in Table 5.3 are statistically significant, we use one-tailed t-value test statistics. We want to answer with significance tests whether a precision difference between two algorithms is statistically significant. We compare the test results with the baseline results in the first step. After computing the t-statistics, we find a corresponding p-value for this test. The results show that the p-value=0.007 is less than the significance level (α = 0.05), so the first test configuration results are significant compared to the previous results.

Since we use different test configurations in this study, we also use another statistical test among the groups. We can technically perform a series of t-value tests based on our results. However, as the number of groups increases, we may end up with many comparisons of pairs. To find the best result among all the results obtained in our study, we must show that our impact is significant. Therefore, we use a one-way variance analysis to determine whether the results are statistically significant. The statistics show a strong p-value, and there are effective results among the results we get in this study. To determine the specific result, we use a post-hoc test. We use the well-known post-hoc test called Scheffe's test. The test results indicate that the configuration using AAFCM, ELMo, and the OWA fusion gives the best result among all the other alternatives.

5.10 Accuracy Improvement Factors Analysis

The improved accuracy in the proposed method can be attributed to several key factors:

1. Contextual Semantics Learning: The introduction of a novel approach to contextual semantics learning is a significant factor. The method captures rich contextual relationships by leveraging query logs and employing deep learning models to generate a "template shot" that integrates concepts from all modalities. This enables a more accurate and nuanced understanding of the semantic associations within and between different modalities, leading to improved ac-

curacy in retrieval.

2. Efficient Clustering Techniques: Integrating various clustering approaches, particularly fuzzy clustering, improves accuracy. Fuzzy clustering is adept at handling uncertainty in video data, allowing the system to better adapt to the inherent complexities of multimedia information. The more nuanced clustering helps refine the retrieval results and enhances accuracy.

3. Fusion Methodology: The proposed fusion method based on the Ordered Weighted Average (OWA) operator is crucial in enhancing accuracy. By intelligently combining ranked lists from multiple retrieval systems, the fusion method mitigates the limitations of individual systems. This collaborative approach results in more accurate and comprehensive retrieval outcomes.

4. Parallel Processing and Scalability: The use of parallel processing capabilities, facilitated by the Apache Spark engine, contributes to the scalability and efficiency of the method. This ensures that the system can handle large-scale multimedia datasets without compromising accuracy. The scalable nature of the architecture allows for the extraction of features from multiple modalities, contributing to improved accuracy.

5. Deep Learning Models: Incorporating deep learning models in generating shot templates enhances the system's ability to understand complex patterns and relationships within the data. The end-to-end deep learning approach enables the extraction of high-level features, leading to more accurate representations and, consequently, improved accuracy in retrieval.

6. Re-ranking Technique: The study integrates clustering-based and Siamese ranking methods, allowing the search results to undergo two ranking algorithms. This comprehensive approach enhances the precision and effectiveness of multimedia information retrieval by leveraging the strengths of both methods and mitigating their respective limitations.

7. Utilization of Siamese Networks: Siamese networks are leveraged to measure the similarity between multimedia items accurately. These networks are adaptable and capable of learning meaningful embeddings, which is essential for

understanding the relationships between data points. By employing Siamese networks in the re-ranking process, the study benefits from their effectiveness in assessing similarity, thereby enhancing the accuracy of search results.

8. Choice of Loss Functions: Contrastive and triplet loss functions are chosen for their effectiveness in learning embeddings accurately representing similarity. These loss functions are tailored to the requirements of ranking tasks, ensuring that the Siamese network can effectively measure similarity between multimedia items and produce accurate re-ranking results.

The method's improved accuracy results from a comprehensive approach encompassing contextual semantics learning, efficient clustering, intelligent fusion techniques, scalability through parallel processing, and utilizing deep learning models. Each of these elements contributes collaboratively to the overall accuracy of the proposed multimodal information retrieval system.

5.11 Implications and Limitations

This study significantly advances multimodal multimedia retrieval systems, highlighting the proposed system's scalability and adaptability for broad applications. Our proposed end-to-end retrieval framework offers a practical solution for extracting relevant information from diverse multimedia data sources. Its effectiveness in deciphering complex contextual relations across and within modalities demonstrates its capability to understand accurate and precise information. The study's focus on selecting optimal word embedding methods, particularly the superiority of ELMo, offers key insights for enhancing semantic interpretation in multimedia data. The superiority of the AAFCM clustering algorithm in query matching over other methods enhances retrieval performance. Integrating cluster-based and Siamese-based ranking methods into a two-stage approach enhances the effectiveness of the system's performance. An innovative OWA-based fusion method introduced for merging ranked lists further enriches the field. The system's resilience in handling noisy labels and flexibility in combining multiple retrieval systems showcase its robustness and adaptability. The system achieves scalability and responsiveness for real-world scenarios by utilizing transfer learning and parallel processing. Comprehensive testing on six popular multimodal datasets, backed by statistical validation, confirms the effectiveness of the proposed approach in ranking retrieval results for complex queries. Remarkably, our method outperforms five of the datasets compared to other studies. The study's findings hold practical significance, paving the way for future implementations in various fields.

While this study introduces a promising multimodal multimedia feature extraction and retrieval system with strengths in deep learning and clustering techniques, several limitations should be acknowledged. Firstly, the generalization of the proposed system to diverse application domains remains an area of exploration, as the effectiveness demonstrated in benchmark datasets might not necessarily translate seamlessly to other contexts. The reliance on deep neural networks and the choice of specific clustering algorithms might introduce biases, limiting the system's adaptability to some data characteristics. Additionally, the experimental validation, while extensive, primarily focuses on benchmark datasets, raising questions about the system's robustness in real-world dynamic scenarios. Furthermore, the dynamic nature of multimedia content, wherein context and relevance can evolve, adds a layer of complexity. Addressing this issue requires advancements in multimodal semantic understanding and the development of robust algorithms. Lastly, the adaptability to evolving multimedia data formats and the generalizability of the proposed transfer learning approach to diverse modalities warrant further investigation. Despite these limitations, the study is a strong foundation for future developments in multimodal multimedia retrieval systems, emphasizing the need for ongoing research and refinement in this dynamic field.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This study introduces a scalable and adaptable multimodal multimedia feature extraction and retrieval system. The system employs deep neural network models to generate pattern templates. Through experiments with multiple datasets, we showcase how our deep learning approach effectively learns intricate contextual relationships and correlations across different modalities and within each modality. Consequently, we can effectively create a template shot (as a model) for searching contextual associations. Furthermore, we emphasize the critical role of selecting an appropriate word embedding method, with ELMo outperforming word2vec and one-hot embedding techniques.

Additionally, we demonstrate the positive impact of clustering algorithms on retrieval performance in multimedia data applications. Specifically, we highlight the significance of the AAFCM fuzzy clustering algorithm within our proposed architecture, as it enhances query matching for multimedia data compared to approaches like FCM, DBSCAN, and K-means. We also introduce an OWA-based fusion approach to combine ranked lists generated by multiple retrieval systems. Our experiments illustrate that employing the OWA operator for list fusion, in conjunction with AAFCM, yields the most favorable outcomes among all configurations. Combining clustering-based ranking with a Triplet loss Siamese network for re-ranking also achieved optimal performance.

Moreover, our approach demonstrates robustness in handling noisy labels, underscoring the resilience of our framework. The flexibility of our system allows for the definition and integration of multiple retrieval systems within the architecture. Leveraging transfer learning to extract features from three modalities and parallel processing using the Spark engine for computations ensures that adapting to new datasets requires minimal effort, highlighting the scalability and dynamism of our work.

In the experiments, We validate these findings by conducting experiments with small and large datasets and subjecting the results to statistical significance tests. We evaluate the performance of our method using six widely used benchmark datasets and compare it against nine state-of-the-art approaches. The results demonstrate the effectiveness of our proposed method in retrieving multimodal multimedia queries.

Future research directions could focus on (1) Developing adaptive learning systems that update and refine their models with new data, feedback, or changes in context, leveraging online, active, and reinforcement learning to ensure continuous improvement. (2) Customizing the framework for sectors like healthcare, education, security, or entertainment, addressing their specific needs and challenges. (3) Enhancing retrieval systems to support semantic search and content creation based on user queries, incorporating NLP to interpret complex requests and deliver content that matches user intent.

REFERENCES

[1] S. Jabeen, X. Li, M. S. Amin, O. Bourahla, S. Li, and A. Jabbar, "A review on methods and applications in multimodal deep learning," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 2s, pp. 1–41, 2023.

[2] Z. Liu and W.-S. Zheng, "Learning multimodal relationship interaction for visual relationship detection," *Pattern Recognition*, vol. 132, p. 108848, 2022.

[3] K. D. Onal, Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, *et al.*, "Neural information retrieval: At the end of the early years," *Information Retrieval Journal*, vol. 21, no. 2, pp. 111–182, 2018.

[4] S. Mai, Y. Sun, Y. Zeng, and H. Hu, "Excavating multimodal correlation for representation learning," *Information Fusion*, vol. 91, pp. 542–555, 2023.

[5] D. Sujatha, M. Subramaniam, and C. R. Rene Robin, "A new design of multimedia big data retrieval enabled by deep feature learning and adaptive semantic similarity function," *Multimedia Systems*, vol. 28, no. 3, pp. 1039–1058, 2022.

[6] W. Zhou, Z. Xia, P. Dou, T. Su, and H. Hu, "Aligning image semantics and label concepts for image multi-label classification," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 2, pp. 1–23, 2023.

[7] X. Deng, S. Feng, G. Lyu, T. Wang, and C. Lang, "Beyond word embeddings: Heterogeneous prior knowledge driven multi-label image classification," *IEEE Transactions on Multimedia*, 2022.

[8] C. V. Gysel, M. De Rijke, and E. Kanoulas, "Neural vector spaces for unsupervised information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 36, no. 4, pp. 1–25, 2018.

[9] G.-M. Park, H.-I. Hyun, and H.-Y. Kwon, “Multimodal learning model based on video–audio–chat feature fusion for detecting e-sports highlights,” *Applied Soft Computing*, vol. 126, p. 109285, 2022.

[10] N. Boehmer, R. Bredereck, and D. Peters, “Rank aggregation using scoring rules,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 5515–5523, 2023.

[11] T. Qin, T.-Y. Liu, J. Xu, and H. Li, “Letor: A benchmark collection for research on learning to rank for information retrieval,” *Information Retrieval*, vol. 13, pp. 346–374, 2010.

[12] T. Mei, Y. Rui, S. Li, and Q. Tian, “Multimedia search reranking: A literature survey,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–38, 2014.

[13] A. Usta, I. S. Altingovde, R. Ozcan, and Ö. Ulusoy, “Learning to rank for educational search engines,” *IEEE Transactions on Learning Technologies*, vol. 14, no. 2, pp. 211–225, 2021.

[14] S. Marrara, G. Pasi, and M. Viviani, “Aggregation operators in information retrieval,” *Fuzzy Sets and Systems*, vol. 324, pp. 3–19, 2017.

[15] B. Ionescu, H. Müller, A. M. Drăgulinescu, A. Popescu, A. Idrissi-Yaghir, A. García Seco de Herrera, A. Andrei, A. Stan, A. M. Storås, A. B. Abacha, *et al.*, “Imageclef 2023 highlight: Multimedia retrieval in medical, social media and content recommendation applications,” in *European Conference on Information Retrieval*, pp. 557–567, Springer, 2023.

[16] L. Zhu, C. Zheng, W. Guan, J. Li, Y. Yang, and H. T. Shen, “Multi-modal hashing for efficient multimedia retrieval: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[17] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, and A. Hussain, “Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions,” *Information Fusion*, 2022.

[18] W. Ma, Q. Chen, T. Zhou, S. Zhao, and Z. Cai, “Using multimodal contrastive knowledge distillation for video-text retrieval,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[19] A. K. Mallick and S. Mukhopadhyay, “Video retrieval framework based on color co-occurrence feature of adaptive low rank extracted keyframes and graph pattern matching,” *Information Processing & Management*, vol. 59, no. 2, p. 102870, 2022.

[20] W. Ji, Y. Wei, Z. Zheng, H. Fei, and T.-s. Chua, “Deep multimodal learning for information retrieval,” in *ACM International Conference on Multimedia*, 2023.

[21] L. V. B. Beltrán, J. C. Caicedo, N. Journet, M. Coustaty, F. Lecellier, and A. Doucet, “Deep multimodal learning for cross-modal retrieval: One model for all tasks,” *Pattern Recognition Letters*, vol. 146, pp. 38–45, 2021.

[22] L. Ying, G. Yingying, F. Jie, F. Jiulun, H. Yu, and L. Jiming, “Survey of research on deep learning image-text cross-modal retrieval.,” *Journal of Frontiers of Computer Science & Technology*, vol. 16, no. 3, 2022.

[23] P. Hu, L. Zhen, D. Peng, and P. Liu, “Scalable deep multimodal learning for cross-modal retrieval,” in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pp. 635–644, 2019.

[24] L. Yang, J.-C. Na, and J. Yu, “Cross-modal multitask transformer for end-to-end multimodal aspect-based sentiment analysis,” *Information Processing & Management*, vol. 59, no. 5, p. 103038, 2022.

[25] C. Yu, Y. Ma, L. An, and G. Li, “Bcmf: A bidirectional cross-modal fusion model for fake news detection,” *Information Processing & Management*, vol. 59, no. 5, p. 103063, 2022.

[26] X. Chen, H. Xie, Z. Li, G. Cheng, M. Leng, and F. L. Wang, “Information fusion and artificial intelligence for smart healthcare: a bibliometric study,” *Information Processing & Management*, vol. 60, no. 1, p. 103113, 2023.

[27] S. Wang, H. Zhao, Y. Wang, J. Huang, and K. Li, “Cross-modal image–text search via efficient discrete class alignment hashing,” *Information Processing & Management*, vol. 59, no. 3, p. 102886, 2022.

[28] Y. Wu, S. Wang, and Q. Huang, “Multi-modal semantic autoencoder for cross-modal retrieval,” *Neurocomputing*, vol. 331, pp. 165–175, 2019.

[29] S. Malik and P. Bansal, “Multimodal semantic analysis with regularized semantic autoencoder,” *Journal of Intelligent & Fuzzy Systems*, vol. 42, no. 2, pp. 909–917, 2022.

[30] D. Feng, X. He, and Y. Peng, “Mkvse: Multimodal knowledge enhanced visual-semantic embedding for image-text retrieval,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 5, pp. 1–21, 2023.

[31] G. Habault, M.-S. Dao, M. A. Riegler, D. T. D. Nguyen, Y. Nakashima, and C. Gurrin, “Icdar’23: Intelligent cross-data analysis and retrieval,” in *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*, pp. 674–675, 2023.

[32] A. Yazici, M. Koyuncu, T. Yilmaz, S. Sattari, M. Sert, and E. Gulen, “An intelligent multimedia information system for multimodal content extraction and querying,” *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 2225–2260, 2018.

[33] S. Sattari and A. Yazici, “Multimodal query-level fusion for efficient multimedia information retrieval,” *International Journal of Intelligent Systems*, vol. 33, no. 10, pp. 2019–2037, 2018.

[34] S. Sattari and A. Yazici, “Multimedia information retrieval using fuzzy cluster-based model learning,” in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–6, IEEE, 2017.

[35] E. Ullah and R. Arora, “Generalization bounds for kernel canonical correlation analysis,” *Transactions on Machine Learning Research*, 2022.

[36] S. Guo, L. Song, R. Xie, L. Li, and S. Liu, “Multiview nonlinear discriminant structure learning for emotion recognition,” *Knowledge-Based Systems*, vol. 258, p. 110042, 2022.

[37] S. Gupta, U. Thakar, and S. Tokekar, “Similarity distance-based kernel canonical correlation analysis for multiview data representation,” in *Proceedings of International Conference on Data Science and Applications: ICDSA 2021, Volume 2*, pp. 649–660, Springer, 2022.

[38] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, “Multi-view discriminant analysis,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 188–194, 2015.

[39] Y. Peng and J. Qi, “Cm-gans: Cross-modal generative adversarial networks for common representation learning,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 1, pp. 1–24, 2019.

[40] B. Li and J. Zhao, “Visual-audio correspondence and its effect on video tipping: Evidence from bilibili vlogs,” *Information Processing & Management*, vol. 60, no. 3, p. 103347, 2023.

[41] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, “Deep canonical correlation analysis,” in *International conference on machine learning*, pp. 1247–1255, PMLR, 2013.

[42] X. Shu and G. Zhao, “Scalable multi-label canonical correlation analysis for cross-modal retrieval,” *Pattern Recognition*, vol. 115, p. 107905, 2021.

[43] Z. Zeng, Y. Sun, and W. Mao, “Mccn: Multimodal coordinated clustering network for large-scale cross-modal retrieval,” in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 5427–5435, 2021.

[44] L. Zhen, P. Hu, X. Peng, R. S. M. Goh, and J. T. Zhou, “Deep multimodal transfer learning for cross-modal retrieval,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 798–810, 2020.

[45] P. Hu, D. Peng, X. Wang, and Y. Xiang, “Multimodal adversarial network for cross-modal retrieval,” *Knowledge-Based Systems*, vol. 180, pp. 38–50, 2019.

[46] M. Eminagaoglu, “A new similarity measure for vector space models in text classification and information retrieval,” *Journal of Information Science*, vol. 48, no. 4, pp. 463–476, 2022.

[47] X. Zhao, F. Nie, R. Wang, and X. Li, "Improving projected fuzzy k-means clustering via robust learning," *Neurocomputing*, vol. 491, pp. 34–43, 2022.

[48] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, 2022.

[49] I. Sadat and S. Keshid, "A clustering study for the optimization of emotional information retrieval systems: Dbscan vs k-means," in *2022 International Conference on Computer Communications and Intelligent Systems (I3CIS)*, pp. 67–71, IEEE, 2022.

[50] J. Yue, W. Zhang, H. Hu, and Z. Shi, "Efficient locality sensitive clustering in multimedia retrieval," in *2013 IEEE 16th International Conference on Computational Science and Engineering*, pp. 403–408, IEEE, 2013.

[51] D. Mahapatra, C. Maharana, S. P. Panda, J. P. Mohanty, A. Talib, and A. Mangaraj, "A fuzzy-cluster based semantic information retrieval system," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 675–678, IEEE, 2020.

[52] L. R. Nair, K. Subramaniam, and G. Venkatesan, "An effective image retrieval system using machine learning and fuzzy c-means clustering approach," *Multimedia Tools and Applications*, vol. 79, no. 15, pp. 10123–10140, 2020.

[53] J. Mohan and M. S. Nair, "Domain independent static video summarization using sparse autoencoders and k-means clustering," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 1945–1955, 2019.

[54] C. Liu, Z. Wu, J. Wen, Y. Xu, and C. Huang, "Localized sparse incomplete multi-view clustering," *IEEE Transactions on Multimedia*, 2022.

[55] A. Rahangdale and S. Raut, "Machine learning methods for ranking," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 06, pp. 729–761, 2019.

[56] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International journal of computer vision*, vol. 87, pp. 316–336, 2010.

[57] W. H. Hsu, L. S. Kennedy, and S.-F. Chang, "Video search reranking through random walk over document-level context graph," in *Proceedings of the 15th ACM international conference on Multimedia*, pp. 971–980, 2007.

[58] W. H. Hsu, L. S. Kennedy, and S.-F. Chang, "Reranking methods for visual search," *IEEE MultiMedia*, vol. 14, no. 3, pp. 14–22, 2007.

[59] W. Chen, T.-Y. Liu, Y. Lan, Z.-M. Ma, and H. Li, "Ranking measures and loss functions in learning to rank," *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[60] C. J. Burges, "From ranknet to lambdarank to lambdamart: An overview," *Learning*, vol. 11, no. 23-581, p. 81, 2010.

[61] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank: theory and algorithm," in *Proceedings of the 25th international conference on Machine learning*, pp. 1192–1199, 2008.

[62] H. Li, *Learning to rank for information retrieval and natural language processing*. Springer Nature, 2022.

[63] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1861–1870, 2019.

[64] P. Kumar, D. Brahma, H. Karnick, and P. Rai, "Deep attentive ranking networks for learning to order sentences," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 8115–8122, 2020.

[65] M. Köppel, A. Segner, M. Wagener, L. Pensel, A. Karwath, and S. Kramer, "Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part III*, pp. 237–252, Springer, 2020.

[66] F. Tang and Q. Ling, "Learning to rank proposals for siamese visual tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 8785–8796, 2021.

[67] S. U. Khan, I. U. Haq, N. Khan, K. Muhammad, M. Hijji, and S. W. Baik, "Learning to rank: An intelligent system for person reidentification," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5924–5948, 2022.

[68] M. Choudhary, V. Tiwari, and S. Jain, "Person re-identification using deep siamese network with multi-layer similarity constraints," *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 42099–42115, 2022.

[69] T. Souček and J. Lokoč, "Transnet v2: an effective deep network architecture for fast shot transition detection," *arXiv preprint arXiv:2008.04838*, 2020.

[70] Y. Wang, W. Liang, H. Huang, Y. Zhang, D. Li, and L.-F. Yu, "Toward automatic audio description generation for accessible videos," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2021.

[71] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[72] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. d. C. Quitry, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Haviv, "Towards learning a universal non-semantic representation of speech," *arXiv preprint arXiv:2002.12764*, 2020.

[73] T. Adewumi, F. Liwicki, and M. Liwicki, "Word2vec: Optimal hyperparameters and their impact on natural language processing downstream tasks," *Open Computer Science*, vol. 12, no. 1, pp. 134–141, 2022.

[74] M. Ulčar and M. Robnik-Šikonja, "Cross-lingual alignments of elmo contextual embeddings," *Neural Computing and Applications*, vol. 34, no. 15, pp. 13043–13061, 2022.

[75] M. K. Gupta and P. Chandra, "Effects of similarity/distance metrics on k-means algorithm with respect to its applications in iot and multimedia: A review," *Multimedia Tools and Applications*, vol. 81, no. 26, pp. 37007–37032, 2022.

[76] S.-S. Li, "An improved dbscan algorithm based on the neighbor similarity and fast nearest neighbor query," *Ieee Access*, vol. 8, pp. 47468–47476, 2020.

[77] V.-H. Vu, "Content-based image retrieval with fuzzy clustering for feature vector normalization," *Multimedia Tools and Applications*, pp. 1–21, 2023.

[78] S. Champathong, S. Wongthanavasu, and K. Sunat, "Alternative adaptive fuzzy c-means clustering," in *Proceedings of the 7th WSEAS International Conference on Evolutionary Computing*, pp. 7–11, Citeseer, 2006.

[79] W. Meng, C. Yu, and K.-L. Liu, "Building efficient and effective metasearch engines," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 48–89, 2002.

[80] P. Chi, Y. Feng, M. Zhou, X.-c. Xiong, Y.-h. Wang, and B.-h. Qiang, "Tiar: Text-image-audio retrieval with weighted multimodal re-ranking," *Applied Intelligence*, pp. 1–19, 2023.

[81] R. Nuray and F. Can, "Automatic ranking of information retrieval systems using data fusion," *Information processing & management*, vol. 42, no. 3, pp. 595–614, 2006.

[82] D. Wei, M. M. Islam, B. Schieber, and S. Basu Roy, "Rank aggregation with proportionate fairness," in *Proceedings of the 2022 International Conference on Management of Data*, pp. 262–275, 2022.

[83] F. Franceschini, D. A. Maisano, and L. Mastrogiacomo, "Ranking aggregation techniques," in *Rankings and Decisions in Engineering: Conceptual and Practical Insights*, pp. 85–160, Springer, 2022.

[84] A. Garba, S. Wu, and S. Khalid, "Federated search techniques: an overview of the trends and state of the art," *Knowledge and Information Systems*, pp. 1–31, 2023.

[85] Y. Li, C. P. Chen, and T. Zhang, "A survey on siamese network: Methodologies, applications, and opportunities," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 994–1014, 2022.

[86] B. Barz and J. Denzler, "Deep learning on small datasets without pre-training using cosine loss," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1371–1380, 2020.

[87] M. E. Renda and U. Straccia, “Web metasearch: rank vs. score based rank aggregation methods,” in *Proceedings of the 2003 ACM symposium on Applied computing*, pp. 841–846, 2003.

[88] R. Fagin, R. Kumar, and D. Sivakumar, “Comparing top k lists,” *SIAM Journal on discrete mathematics*, vol. 17, no. 1, pp. 134–160, 2003.

[89] D. Carmel, D. Cohen, R. Fagin, E. Farchi, M. Herscovici, Y. S. Maarek, and A. Soffer, “Static index pruning for information retrieval systems,” in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 43–50, 2001.

[90] S. Sattari, “Multimedia Database Research Group.” `http://multimedia.ceng.metu.edu.tr/index.php/en/projects/metu-mmds/`, 2023. [Dataset].

[91] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “Yfcc100m: The new data in multimedia research,” *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[92] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, “Nus-wide: a real-world web image database from national university of singapore,” in *Proceedings of the ACM international conference on image and video retrieval*, pp. 1–9, 2009.

[93] Y. Peng, X. Huang, and Y. Zhao, “An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges,” *IEEE Transactions on circuits and systems for video technology*, vol. 28, no. 9, pp. 2372–2385, 2017.

[94] S. J. Hwang and K. Grauman, “Reading between the lines: Object localization using implicit cues from image tags,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 6, pp. 1145–1158, 2011.

[95] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, “A new approach to cross-modal multimedia retrieval,” in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 251–260, 2010.

[96] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[97] Q. Wang, L. Zhan, P. Thompson, and J. Zhou, "Multimodal learning with incomplete modalities by knowledge distillation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1828–1838, 2020.

[98] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[99] N. Wang, H. Cao, J. Zhao, R. Chen, D. Yan, and J. Zhang, "M2r2: Missing-modality robust emotion recognition framework with iterative data augmentation," *IEEE Transactions on Artificial Intelligence*, 2022.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: SATTARI, SAEID
Nationality: Turkish (TC)

EDUCATION

Degree	Institution	Year of Graduation
M.S.	METU	2013
B.S.	IAU	2006
High School	Sefa High School	2001

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2023-Current	OBSS	Data Architect
2022-2023	EPAM	Senior Data Engineer
2021-2022	Firefly	Senior Data Engineer
2016-2021	E-Kalite	Big Data Engineer

PUBLICATIONS

1) Sattari, S., & Yazici, A. (2018). Multimodal query-level fusion for efficient multimedia information retrieval. International Journal of Intelligent Systems, 33(10), 2019–2037.
2) Yazici, A., Koyuncu, M., Yilmaz, T., Sattari, S., Sert, M., & Gulen, E. (2018). An intelligent multimedia information system for multimodal content extraction and

querying. Multimedia Tools and Applications, 77(2), 2225–2260.

3) Sattari, S., & Yazici, A. (2017). Multimedia information retrieval using fuzzy cluster-based model learning. 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 1–6.

4) Sattari, S., & Yazici, A. (2016). Efficient multimedia information retrieval with query-level fusion. Flexible Query Answering Systems 2015: Proceedings of the 11th International Conference FQAS 2015, Cracow, Poland, October 26-28, 2015, 367–379.

5) Yazici, A., Sattari, S., Yilmaz, T., Sert, M., Koyuncu, M., & Gulen, E. (2016). Metu-mmds: An intelligent multimedia database system for multimodal content extraction and querying. MultiMedia Modeling: 22nd International Conference, MMM 2016, Miami, FL, USA, January 4-6, 2016, Proceedings, Part II 22, 354–360.