UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours
Final Paper 2020

Title: DanceBase: A Dance Multimedia Database and Search Engine implementing Query Expansion

Author: Carryn Joseph

Project Abbreviation: SoDA

Supervisor(s): Associate Professor Maria Keet

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	20
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	
System Development and Implementation	*0*	*20*	20
Results, Findings and Conclusions	*10*	*20*	10
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	**80**

DanceBase: A Dance Multimedia Database and Search Engine implementing Query Expansion

Carryn Joseph
Department of Computer Science
University of Cape Town
Cape Town, South Africa
jspcar006@myuct.ac.za

ABSTRACT

Dancing is an activity commonly known to many. Learning dance of any dance genre can be difficult and even more difficult without professional training. Dance move terminology often gets lost in the learning process. This can make searching for digital dance resources challenging. This project explores the design of a database to store dance terminology and multimedia, e.g. video, image and audio, in conjunction with a dance system that implements a comprehensive search engine. The search engine uses query expansion techniques to improve the effectiveness of the search results by improving the recall of predictive results.

CSS CONCEPTS

- **Multimedia Database** • **Query Expansion** → Content-Based Queries, Pseudo-Relevance Feedback

KEYWORDS

Multimedia Database, Dance, Latin Dance, Query, Search Engine

1 INTRODUCTION

Dance. The word itself is a term familiar to many as the moving of one's body to a rhythm or beat. It is not something only done today but rather is an age-old art form been done for centuries as a form of self-expression and cultural expression. For many cultures, dance forms an important part of cultural heritage. Latin American dance styles are of the many dance styles that hold significant cultural value. Some of Latin dance styles include Salsa, Bachata, Cha-Cha as well as many others. It is important to preserve dance because of its cultural value. This can be done by teaching dance to others as well as digitizing dance resources. Dancers learn to dance through various methods including making hand-written notes and annotating sketches as well as using digital resources such as videos and images of dance movements and choreographies [1]. Dancers can also search for dance information online using YouTube or Google. This brings about a challenge for new dancers who are unfamiliar with dance terminology and don't know what to search for to find what they are looking for.

Over recent years, resources have gone into the research and development of dance tools using modern technology [2]. Strides have been made in the development of tools using virtual and augmented reality such as VR Salsa [18] which was developed for dance education. Other research efforts are going into capturing and analysing dance movements using motion capture, like the WholoDancE project [12], as well as annotating dance movements. Of the literature found, most were focused on European-style dance genres. Though there exist dance projects that focus on salsa dance specifically such as VR Salsa [18] and Let's Dance [19]. There are also mobile applications available aimed at teaching dance, e.g. Learn Salsa and Salsa Anywhere, however, these applications don't have corresponding research from which to gather information. Of the research gathered, none of the Latin or Salsa-based projects focuses on storing multimedia data or how to search for it. This provided an opportunity to explore a tool that utilizes multimedia databases.

A user's informational need is considered to be imprecisely articled by their query [3]. This is particularly true in the case of dance, where dancers may use different or incorrectly terminology to describe the same dance move. The aim of this project is thus to develop a system to help dancers search for Latin-dance multimedia from a multimedia database. The focus is to improve the effectiveness of the results retrieved from a user's query by implementing query-expansion techniques.

This paper goes through the development of this project from gathering requirements to the results of the developed system. The rest of this paper is divided into sub-sections that dive into the details on the development process. Section 2 is the background section, it is an analysis of available literature related to multimedia database, query expansion and existing dance system. Section 3, Requirements Analysis and Design highlights the requirements of the system and provides details on the database design process. Software Design and Implementation is in section 4. Section 4 gives detail into the development methodologies, languages and other software decisions. Included in this section is also details on the system included architecture, data structures and implemented algorithms. Section 5, Query Evaluation provides the methods used to evaluate the search engine and provides the results of the evaluations performed. It also gives a discussion on what those results mean about the success of the system. Lastly, the conclusions, section 6, states the result of the system as well as future work opportunities.

2 BACKGROUND

2.1 Multimedia Databases

Traditional databases today can be divided into SQL and NoSQL databases. The definition of a traditional database is a collection of related data [4]. What makes a traditional database a multimedia database is as the name suggests: the capacity to store multimedia data. Multimedia data refers to a range of different data types including text, audio, image, graphic objects and video [4, 5]. One of the ways dancers learn and record dance movements is through existing storing and viewing media in existing multimedia databases such as a Gallery application or YouTube[6]. The Gallery application that is native to most modern-day phones [7]. The physical memory of the images is stored on a local SQLite database [7]. YouTube is a video-sharing platform that allows users to upload and search for videos based on video metadata, uploader information and video tags [8]. When searching, YouTube also implements its own query expansion by showing videos related to videos of search results, this is a form of relevance feedback.

2.2 Query Expansion

The accuracy of query results directly affects the query itself [9]. This has led to the development of different techniques to improve the results attained from a query. White and Marchionini [9] categorized query adjusting techniques into two forms: real-time query support and retrospective query expansion [9]. Retrospective query expansion is often done through relevance feedback [3]. From the literature arose several different methods to perform relevance feedback. The Rocchio Method uses graphical representations like the vector-space model to determine the relevance of queries based on the distance between query vectors [3]. Other models of determining relevance include the relevance model, the mixture-model feedback method, and the unigram relevance model [3, 10]. Another method proven to improve query results is pseudo relevance feedback, or blind feedback [10]. Just as there are different models to relevance feedback, so are there different approaches to pseudo relevance feedback. Local context analysis is the analysis of the phrase structure of queries to improve the effectiveness of query results [3]. A different approach to query expansion is a content-based query [11]. The objective of content-based querying is finding search results based on similarities within the data i.e. similarities within the content of the data.

2.3 Existing Dance Systems

From the research explored arose three significant dance project namely: WhoLoDancE [12], BalOnSe [13] and the Balinese Dance Preservation System [14]. WhoLoDancE [12] and BalOnSe [13] are both projects focused on the development of systems which aid in the learning of European style dance genres such as Ballet. Both projects developed web-interfaces with querying and annotation capabilities. Similarly, the Balinese Preservation Dance project [14] modelled a website that allowed querying of dance history, dance descriptions, studio information and multimedia. Although the incorporation of media in the system was not the focus of that project.

Of the systems reviewed, a common aspect is the type of database used to store the data. The WhoLoDancE Web Movement Library (WML) [15], BalOnSe system and Balinese Preservation System were all designed to interact with relational databases. The querying of these databases is based on several different methods. While the Balinese System only offers basic keyword search functionality, the WholoDance Interface and BalOnsE offer different approaches to improve the searching capabilities. WhoLoDancE implements multimedia search queries based on media data as well as user inputted annotations while BalOnSe developed a Ballet ontology to improve the search based on different dance terminology, The objective of this is to allow both expert dance users and novice users to search for dance media, this aligns with project aim for the DanceBase system,

3 REQUIREMENTS ANALYSIS & DESIGN

3.1 Requirements Gathering

An introductory meeting with the client was arranged to establish the fundamental requirements of the project. Initially, the client for this project was to be a dance expert from the Evolution Dance Company, however, due to COVID-19 and limited communication, this plan fell through and Associate Professor Maria Keet took the position as a client for the duration of the project.

The client required to produce a tool that could help dancers quickly and easily get notes on dance movements while in a social setting. Based on this request, available and existing dance systems, we suggested a tool that dancers and other users could use to search for dance multimedia which would include tutorials, choreographies and other types of media. This suggestion was approved by the client which lead to the requirements gathering for the new dance system.

The first phase of the requirement gathering was eliciting requirements. Through a literature review, we were able to gain insight into multimedia databases, search techniques and existing dance system that lead to the background section previously stated. From this background information, we devised a list of requirements for the database and the dance system. This information was laid out in a proposal for the client to approve.

Below is the list of requirements as stated in the project proposal:

1. A multimedia database should store and retrieve media.
2. The system should allow the user to search for multimedia
3. The system should implement at least two query expansion techniques.

Further research went into the decision of which query adjusting techniques to implement to improve the search engine in the dance system. We intend to allow users of all experience levels to be able to search for dance media. Below is the chosen query techniques:

- Content-Based Querying, this will help users to search media by its contents not only the metadata of the media.
- Pseudo-Relevance feedback, this will provide more result options that are related to the results of the query. This will improve the chances the user will find what they are looking for even if they don't put in the correct query.

3.2 System Architecture

The initial iteration was dedicated to designing the architecture for the system. Based on functional requirement number 1 in section 3.1, it was clear the architecture of the system would require a database and the appropriate architecture to go with it. For this reason, the DanceBase system implements a layered architecture as shown in figure 1.

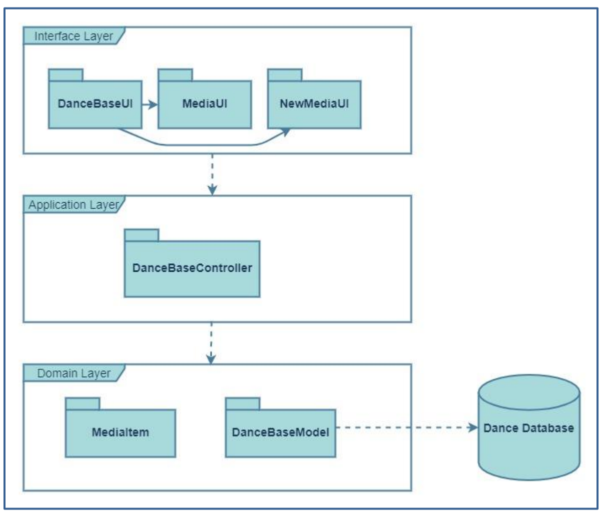


Figure 1: Layered Architecture of DanceBase System

The DanceBase system is a transaction processing system. In this scenario, the queries made to the search engine are the input, sending the query to the database and getting results is the process which then produces the output of media files. The Model-View-Controller (MVC) architecture pattern guided the assignment of responsibilities within the system to ensure a high degree of cohesion and low degree of coupling.

Based on the following architecture design decisions the following design was produced:

- Interface Layer: this layer acts as the view component of the MVC pattern. All interfaces the user interacts with form part of the interface layer. The three main layers the user interacts with include the main 'DanceBaseUI', the ss'MediaUI' which displays individual media and the 'NewMediaUI' form for adding media to the database.
- Application Layer: The 'DanceBaseController' is the controller component of the MVC architecture. This class acts as the intermediary between the interface and the domain layer. It controls the data that is to be displayed on the interfaces and sends the search inputs from the interfaces to the domain layer to get the necessary information.
- Domain Layer: This layer houses the data of the system. The 'DanceBaseModel' is the model component of the MVC architecture, it provides the connection to the multimedia database and handles all communication between the system and database.

3.3 Multimedia Database Modelling

The multimedia database is a fundamental aspect of the project. It is the backbone of the DanceBase system and houses the data used by the system and the query implementations. The objective of the multimedia database is to design and develop a singular database that holds all relative data including the media file itself without the need for an external repository.

4.2.1 Requirements

Understanding salsa dance was the initial step to designing the database. As a participatory observation, we attended three dance classes - a salsa class, bachata class and a cha-cha class – at the Evolution Dance Company. From these classes, we were able to get a feel for the different dance styles and learnt a few of the fundamental basic dance moves for each style. We were also able to speak to dance expert, Angus Prince, and get a better understanding of dance movements and the three dance styles we incorporated in the system. The information gained from those experiences was invaluable when finding media to populate the database. We were also provided dance media material from the client including videos of dance tutorials. audio and dance annotation images.

4.2.2 Conceptual Data Model 1

Based on the requirements and the information gathered from the participatory observation, the first conceptual database model was designed. This model is shown in figure 2. The model was presented to the client in a feasibility demonstration and was followed with some feedback on the model. Feedback from the client was that the E-R database model is too simplistic and does not display enough technical depth. Additionally, it does not allow for much expansion on queries due to lack of relationships and entities. Although it is not a perfect model, it did help gain some clarity on what sort of information the database should have and what relationships should be included.

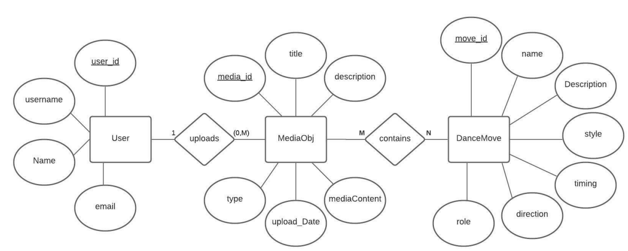


Figure 2: First model of DanceBase Multimedia Database

4.2.3 Conceptual Data Model 2

After receiving feedback on the first conceptual model, we made a few changes to increase the complexity of the design. We removed the dance style attribute from the dance move entity and made it an entity. This decision allowed for a relationship between audio and style since audio files do not contain dance moves and therefore would not have a relationship with the dance move entity. Other additional entity types include a movement entity and a tag entity. The movement entity would keep a record of basic movements, e.g. step right foot forward or put weight on the left foot. These basic small movements form the building blocks of dance movements. The tag entity was inspired by the tag feature available on social media platforms including YouTube and Instagram. The tag allows for associations to be made between different multimedia based on the tags they are related to. Additionally, inheritance was added to the media object entity to produce children entities for each form of media – video, image and audio. This was both with the intent that it might improve search capabilities and increase the quality of the design. Figure 3 shows the second iteration of the database conceptual model.

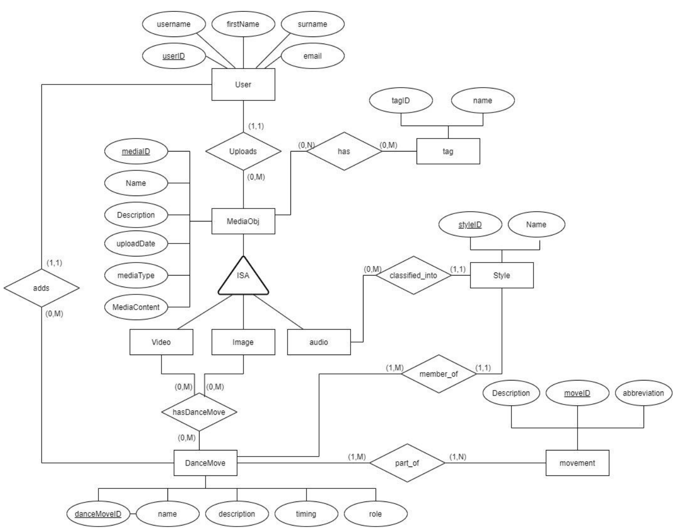


Figure 3: Second model of DanceBase Database Model

4.2.4 Conceptual Data Model 3

The initial development of the database was based on the second conceptual model. Upon development of the query techniques, it was realized that certain entities, including tags and movement, were not being utilized and were not necessary for the query expansion implemented in the dance system. This led to the removal of those entities to produce a third conceptual model. This model is the final E-R model for the multimedia database. The final design consists of seven entity types: user, dance movement, dance style (genre), media object and the inherited entities of media object – audio, video and studio. Almost all relationships are zero-to-many or one-to-many. The relationship between dance move and media object is many-to-many as a media object may refer to more than one dance move and a dance move may be referenced in more than one media object.

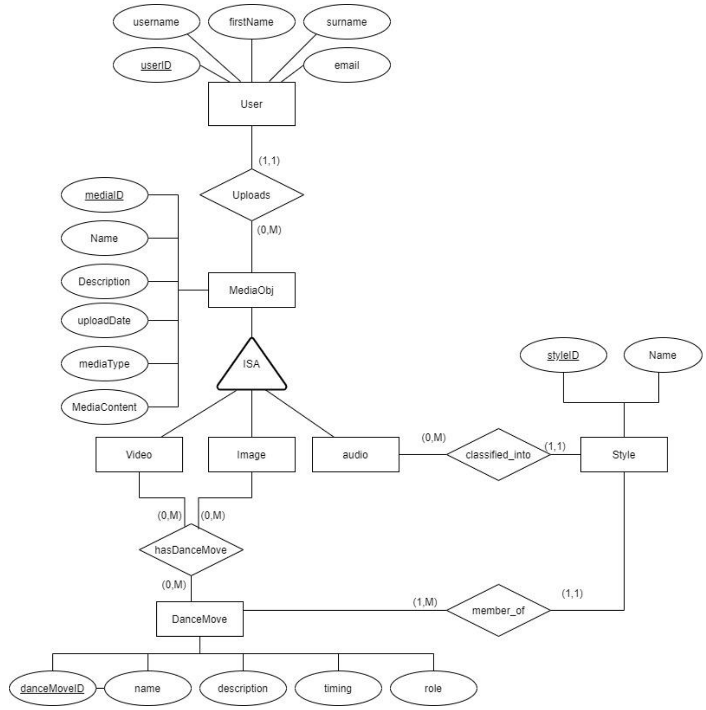


Figure 4: Final Model of DanceBase Database Model

4 SOFTWARE DEVELOPMENT AND IMPLEMENTATION

The DanceBase system was developed using a combination of iterative and waterfall approach [16]. The project followed the standard phases of a waterfall approach, this gave the project direction and structure as to ensure that all aspects of the software development project were addressed. The iterative aspect of the iterative waterfall approach provided the flexibility required to revisit passed phases of the development cycle when requirements or design needed to be adjusted. For the build phase of the project, a bottom-up approach was taken to develop the system [17]. The system development was done in a series of activities. Each activity developed some component or requirement of the system. Each component would build on the next to produce the final DanceBase System. The project included three main components: Multimedia Database, System interface and Query implementation.

4.1 Multimedia Database Development

The multimedia database was the product of the first design activity. The process of modelling the multimedia database overlapped with the development of the database. On the design of each model, a clean database was created to test the viability of the design.

When designing the database, a relational model was deemed the most viable. The structure of relational tables added relationships between data tables that would help in the querying process. The database was developed using PostgreSQL. PostgreSQL is an object-relational database management system (ORDMS), this allows for the use of a relational database with the addition of object-oriented database support which includes support for features such as inheritance which is included in the DanceBase database model.

Based on the conceptual diagram of the database, a table was created for each entity of the model. The attribute of each entity reflected as a column in that entities respective table. Foreign keys were used to create zero-to-many and one-to-many relationships. For many-to-many relationships, a relational table was created. From the model, 'hasDanceMove' was the only relational table created. Its only attribute was a primary produced from the combination of the 'mediaobject' primary key and 'dance movement' primary key. Although PostgreSQL offers the option of hierarchical relationships for inheritance, it is limited in what can be done with the feature. To avoid errors in the querying process, Audio, Image and Video entities were denormalized into a single table inheritance. This was done by adding the attributes of the entities to the superclass (entity) 'mediaobject'. The database implemented no indexing, data domains, triggers or stored procedures. The data stored in the database were primarily primitive data types including VARCHAR and Integer. The physical media files are stored in mediaobject table using a byte array data type. This allows the physical files to be stored in the database rather than just a file path to the media files.

4.2 DanceBase System

4.2.1 Development Language

The DanceBase system was then developed using java. NetBeans IDE was the platform used for development with the interfaces being developed using Swing. Imported into the system was a java archive file 'bcprov-jdk15on-165.jar' to allow for connection and communication with the PostgreSQL dance database.

4.2.2 Classes in DanceBase System

The java classes for the system were guided by the MVC pattern. The model class 'DanceBaseModel' handles the data connection to the PostgreSQL database. Queries formalised based on user searches are sent to the model class. The model class then sends the query statement to the database and retrieves the results of the query. The 'DanceBaseController', the controller class, gets user searches from the interface and produces SQL query statements that are then sent to 'DanceBaseModel'. The controller class also handles which information is given to the interface to be displayed based on the results received by the model. The 'MediaItem' class is an object class which stores media object information. This class is used to contain information related to individual media. The class aids in the display of media object information. The remaining classes are boundary classes which display the interfaces the user interacts with.

4.2.3 Data Structures in DanceBase System

The DanceBase system implements object-oriented programming as well as other non-primitive and primitive data structures.

MediaItem object

MediaItem is an object class used to store information relating to media objects. The objective of this class is to store the data related to individual media items within its structure. When displaying an overview of all the media, MediaItem objects make it easier to display the necessary information for each media object obtained from the database.

ArrayList

The above-mentioned media objects are then stored in an ArrayList. The ArrayList created in the DanceBaseModel is stored in the DanceBaseController class. This non-primitive data structure is ideal for the storing due to its flexibility in size and the ability to add objects during run-time.

4.2.4 Interfaces

The interface was designed to be simple and easy to use. The focus of the interface design is to show the available functionality and to display search results. For this reason, the functionality is readily available from the opening interface. The main purpose of the interfaces is to visually represent the results retrieved from querying the system. The system consists of three interfaces: 'DanceBaseUI', 'MediaUI', 'NewMediaUI'.

4. 'DanceBaseUI' is the opening interface. Upon opening, the interface displays an overview of all media available in the database. Each media item is displayed as a Jbutton the user can click on to open and view the media and media details. Additionally, there is a side panel on the left of the interface with the functionality the system has to offer. From this interface, the user can search for media, view media items and open the interface to add media.
5. 'MediaUI' opens when a user clicks on a media item button. Then display opens in a new frame displaying the name of the media, the description and the media itself.
6. 'NewMediaUI' is a new frame that opens when the user clicks the "Upload Media" button on the 'DanceBaseUI'. This interface allows the user to enter in the details of a new media object and upload it to the database.

4.3 Query Implementation

This section further describes the details about the querying techniques used in the system to improve the searching capabilities of the DanceBase System. Querying the database was performed using Structured Query Language (SQL). The searching process was performed in several stages. First, the user's query input is retrieved from the 'DanceBaseUI.' The query is then sent to three different functions to undergo the three different query implementations: standard baseline query, content-based query and pseudo-relevance feedback. Each function produces a SQL query using the canned query format and the user's search. A function from the 'DanceBaseModel' is then called to send the query statement to the dance database and retrieve the results. The results of the queries retrieved from the database are then added to individual 'MediaItem' objects which are recorded in an ArrayList Data Structure.

4.3.1 Standard Baseline Query

The standard query refers to basic search functionality. For this project, the standard query is baseline as it does not include any query modification. The system allows for searching media items only by media metadata including names associated with media objects, descriptions of media and file type (i.e. audio, video or image).

Upon receiving a search input, from the user, the input is stored in a string. This string is cleaned for common non-value words such as 'a', 'the', 'of'. After this, the string is split into individual words and is stored in an Array. When performing the baseline query function, in the 'DanceBaseController' class, a brute force algorithm is used to search every media item to determine if that item's media name or its description contains at least one of the search words that were stored in the array. Figure 5 shows the SQL pattern query used to determine if either of the media object attributes contain one of the search words. The search words are represented as 'term' in figure 5. This query is performed for every search word in the array and every media object in the database.

```
SELECT media_id, media_name, description, media_type
FROM public.mediaobject
WHERE media_name ~ 'term' OR description ~ 'term';
```

Figure 5: SQL query for Standard Baseline Search

4.3.2 Content-Based Query

Most videos and images contain descriptions, annotations or demonstrations of dance movements. This makes the dance moves the content of the media file. The purpose of the content-based query is to allow users to search for media files based on the content it holds, i.e. based on the dance moves it is related to. The database has two entities that play a role in this query, the media object and the dance move. The two entities have a many-to-many relationship. This relationship has a relational table called 'hasDanceMove'. This table is what connects to two entities in the query process. See figure 7 for a conceptual representation of this relationship. Figure 6, below, shows the SQL query used to determine the media objects that are related to the dance move mentioned by the user in his/her search where 'search input' is the user's search input from the 'DanceBaseUI' interface.

```
SELECT media_id, media_name, description FROM public.mediaobject
WHERE media_id in (SELECT media_id FROM public.hasdancemove
WHERE dancemove_id = (SELECT dancemove_id FROM public.dancemove
WHERE LOWER(dance_name) = LOWER('search input')));
```

Figure 6: SQL query of Content-Based Search

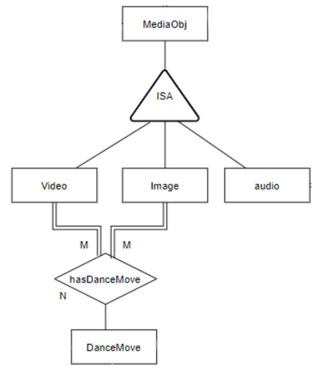


Figure 7: Partial model of the relationship between media object and Dance Mode in Database

4.3.3 Pseudo Relevance Feedback Query

Relevance feedback is an approach used to reduce vocabulary mismatch. In dance, vocabulary mismatch may occur when dancers or other users search for a dance move but doesn't know the exact name of the dance move or enters a variation of the dance move name. The purpose of relevance feedback is to expand the terms of a query to produce more results relating to the user's search. This hopes of this are to increase the number of results that could be relevant to the user's query. There are several different models to performing relevance feedback, however, most relevant feedback models require user input to decide which results are relevant and which are non-relevant. Pseudo-Relevance Feedback also referred to as blind feedback [10], is a way to perform relevance feedback without user input.

The assumption used in pseudo-relevance feedback is that the results retrieved by the original query are relevant as opposed to the user deciding which results are relevant. The pseudo-relevance in this system is performed in a function 'pseudoFeedbackSearch' in the controller class. Below is the step-by-step process of determining the pseudo-relevance search results:

Step 1: Perform a baseline query. The results of this query are stored in an ArrayList.

Step 2: Determine terms used in the results. The terms were collected from words in the name and description of all the media items in the ArrayList from step 1.

Step 3: Determine the tf-idf weighting for all terms retrieved in step 2. Tf-idf (term frequency-inverted document frequency) is a statistical measure used to determine the importance of words in the database. Tf-idf is the product of the term frequency, how often term appears in results metadata, and the inverted document frequency, the logarithm of the number of documents over the number of documents that contain the term. Below is the formula used to determine the weighting.

$$Tf\text{-}idf(t,d,D) = tf(t,d) \cdot idf(t,D)$$

Where

$$Tf(t,d) = freq(t,d)$$
$$Idf(t,D) = log(\frac{N}{|\{d \in D : t \in d\}|})$$

Step 4: Determine the top 5 ranked terms based on highest tf-idf weights.
Step 5: Perform baseline query, using the SQL query in figure 5, on the top-ranked terms retrieved and add results to the results of the original baseline query.

4.4 Software Testing

To test the software, we executed a series of functional tests. The objective of these tests is to ensure the functionalities of the system work as planned and as the requirements mentioned in section 3.1. Appendix A shows the test plan used to evaluate the system. The main functions that need to be tested are if the user can search for media and if the user can add media to the database.

5.1.1 System Testing Results

Below is a list of the functional tests performed on the system, according to the test plan, and the outcome thereof. The results show that all functionalities work as required.

Test Case	Dataset Description	Test Cases
1	Start the app and check if it opens onto the overview page	Passed – App starts and opens on the overview page
2	Clicking on a media object takes the user to a new interface of the media and its information.	Passed – Clicking on the media opens new interface and displays media information.
3	Videos and Audio play when clicked on	Passed – videos and audio play when clicked on.
4	The user adds media object to the database	Passed – Media object is added to the database.
5	User can search for media using search bar	Passed – Search functions filters results based on user's input into the search bar
6	Query Expansion Testing: application shows results based on the content of the media file	Passed – Results based on the content of the media file are displayed.
7	Query Expansion Testing: application suggests media relevant to the input query	Passed – Some results displayed that are relevant to the search input

Table 1: Functional Test Plan of DanceBase System

5 QUERY EVALUATION

5.1 Material and Methods

The objective of the system is to have an effective search engine and to do so by implementing query expansion techniques. Precision and recall measures were used to evaluate the effectiveness of the search techniques implemented. In the context of this project, precision refers to how many of the results retrieved were relevant while recall shows how many of the relevant media objects were retrieved.
The formulas applied to determine precision and recall were as follows:

$$\text{Precision} = \frac{\text{Number of relevant media objects in results}}{\text{number of media objects in results}}$$

$$\text{Recall} = \frac{\text{Number of relevant media object in results}}{\text{number of relevant media objects}}$$

The system implemented three querying techniques: baseline querying, content-based querying and pseudo-relevance feedback. The tests were held on four variations of the search function:

Variation 1: Baseline querying
Variation 2: Baseline querying and Content-Based querying
Variation 3: Baseline querying and Pseudo-Relevance Feedback
Variation 4: All three query techniques

The hypothesis for this evaluation is that the addition of each query technique would improve the recall of the query results at the expense of the precision and that variation 4 would have the highest recall value.
To effectively evaluate the system, the queries used in the evaluation had to be varied and apply to different aspects of the system. Ten queries were selected for evaluation based on five possible categories users would search by - style, multimedia type, media name, media content (e.g. tutorial, choreography) and dance move. The evaluation consisted of two queries from each category. The 2 style queries were selected at random out of the available styles in the database. Dance move queries and media content queries were randomly selected from the dance database. While the media type queries were selected from the combinations of style and media type, e.g. salsa video or cha-cha audio. The random selection process led to the following table of queries:

Category	Query 1	Query 2
Style	Salsa	Bachata
Multimedia type	Bachata video	Cha Cha audio
Media name	Let's get loud – Jennifer Lopez	Salsa side step
Media content	Bachata Choreography	Cha Cha tutorial
Dance Move	Tigeure Special	Side Break

Table 2: Table of Queries used in Evaluation

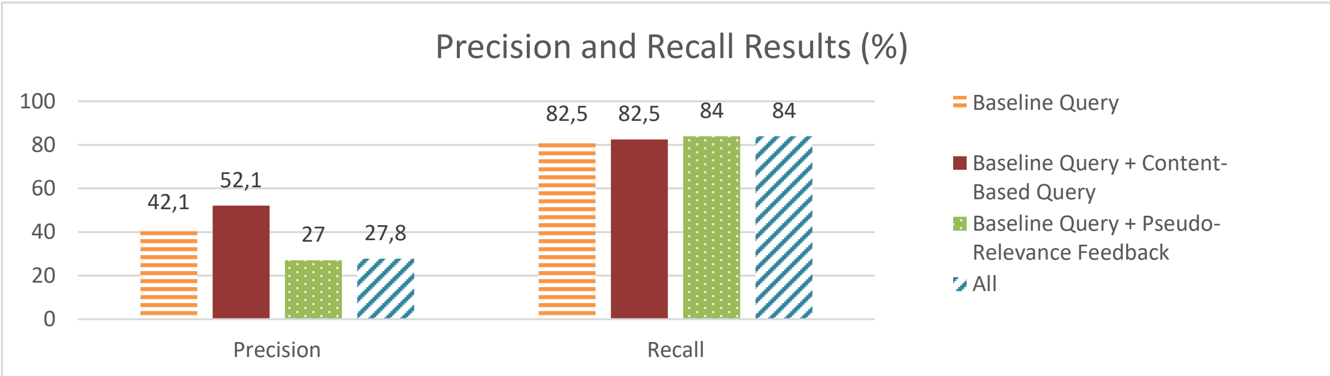


Figure 8: Precision and Recall Results from Query Evaluation

5.2 Results and Discussion

As mentioned in section 5.1, the queries defined in table 2 were applied to each variation of the search function in the DanceBase system. The precision and recall values were calculated from each query performed. The full list of query evaluations and results can be found in Appendix F. The table shows the number of relevant and non-relevant media objects for each query. It also provides the precision and recall percentages for the individual queries. Figure 8, above, shows the average precision and recall results from the ten queries for each variation of the search engine.

The precision result for baseline and content-based querying was the highest at 52%. This increase of 10% from the baseline query precision value is due to the content-based query retrieving results based on the dance moves where the dance move may not be mentioned in the media metadata. When looking at all the queries, the precision results for baseline querying with and without content-based querying are generally the same for the different categories of search, except where the input was dance move related. This indicates that the content-based querying technique only improves precision when the querying dance move-related queries.

Contrary to the content-based query, the precision of the baseline query and pseudo-relevance feedback was reduced from 42% to 27%, and 27.8% with the implementation of all query techniques. These results coincide with the hypothesis that with the implementation of query expansion technique, the precision would decrease. This decrease is due to new terms being added to the query, producing more results than the original baseline query. Because the pseudo-relevance feedback adds results to the baseline query results, the number of true relevant results are either the same or more than that of the baseline query results for all query evaluations.

The recall results were not as promising as had hoped, however, there was some improvement. As mentioned above, the results for the baseline query with and without content-based querying were similar except for dance move-related queries. Dance move-related queries made up only 20% of the queries applied in the evaluation which resulted in the average recall value not being affected by the differences in its results. According to the hypothesis, the addition of each query expansion technique would increase the recall. Based on the average recall, this is not true for content-based querying, however, this finding could differ if more of the queries evaluated contained dance moves.

The hypothesis did end up being true for the pseudo-relevance feedback which increased by 1.5%. Overall, the recall values for pseudo-relevance feedback on the queries were improved in comparison to the baseline querying. For the majority of the queries performed on the system with all query techniques implemented the results were like that of the results from the baseline query with the pseudo-relevance feedback. This was as a result of the baseline and content-based queries being like the baseline queries because the pseudo-relevance feedback is based on the results retrieved from the query.

From the results and the findings, we found that pseudo-relevance feedback did indeed improve the recall of the search engine results while the content-based query technique improved the precision but had little to no effect on the recall. Based on averages, the addition of query expansion techniques did improve the recall of the DanceBase search engine.

6 CONCLUSIONS

This project aimed to design a multimedia database and develop a Latin dance application that could search for multimedia using a search engine that would be improved by query expansion techniques. The system implemented standard baseline query, content-based query and pseudo-relevance feedback techniques. The results showed an increase in average recall with the addition of both query expansion techniques. This along with a reduction in precision value as expected. These results and the findings mentioned above prove that the implementation of the query expansion techniques did improve the effectiveness of the search-engine in the intended manner. The final product of this project is a system that connects to a multimedia dance database and has an effective search engine with query expansion implementation.

There are several opportunities to expand on this project in future work. There is the potential for the DanceBase system to be developed on a mobile platform, this would allow for quick and easy access to the system when in a social setting. Several other query expansion techniques could be implemented to further improve the quality of the search engine. One of these techniques includes incorporating the dance ontology developed by project partner, Kouthar Dollie, to improve relevance feedback.

REFERENCES

[1] Marianela Ciolfi Felice Sarah Fdili Alaoui Wendy E. Mackay. 2018. Knotation: Exploring and Documenting Choreographic Processes. In *Proceedings of the Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, April 2018, Montréal. DOI: https://doi.org/10.1145/3173574.3174022"

[2] Katerina El Raheb, Marina Stergiou, Akrivi Katifori, and Yannis Ioannidis. 2019. Dance Interactive Learning Systems: A Study on Interaction Workflow and Teaching Approaches. ACMComput. Surv. 52, 3, Article 50, June 2019, 37 pages. DOI: https://doi.org/10.1145/3323335

[3] Symonds, M., Bruza, P., Zuccon, G., Koopman, B., Sitbon, L. and Turner, I. Automatic query expansion: A structural linguistic perspective. *Journal of the Association for Information Science and Technology*, 65, 8 (2014), 1577-1596.

[4] Oya Kalipsiz. 2000. Multimedia databases. In *Proceedings of the 2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics*, Istanbul, 2000. doi: https://doi.org//10.1109/IV.2000.859745

[5] Vincenzo Moscato and Antonio Picariello. 2018. A. *Multimedia Data Modeling and Management*. Springer International Publishing AG 2018, City, 2018. https://doi.org/10.1007/978-3-319-61893-7_16

[6] Nathaniel Ostashewski, Doug Reid and Marcia Ostashewski. 2016. Utilizing Multimedia Database Access: Teaching Strategies Using the iPad in the Dance Classroom, Journal of Dance Education, 16:4, 122-128, DOI: https://doi.org/10.1080/15290824.2015.1091939

[7] Jingchun Jiang, Bin Xi, Shunxiang Wu, Huan Li, Yu Rao, Ziliang Wang, Shaobo Zhou. 2018. Acquisition System of Physical Memory Image on MTK-based Mobile Phone. In *Proceedings of the Proceedings of the 2nd International Conference on Computer Science and Application Engineering - CSAE '18*, 22-24 October, 2018, Hohhot. DOI: https://doi.org/10.1145/3207677.3278050

[8] Ankur Satyendrakumar Sharma and Mohamed Elidrisi. 2008. Classification of Multi-Media Content (Video's on YouTube) Using Tags and Focal Points, University of Minnesota

[9] White, R. W. and Marchionini, G. Examining the effectiveness of real-time query expansion. *Information Processing & Management*, 43, 3 (2007), 685-704.

[10] Yuanhua Lv and ChengXiang Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. Association for Computing Machinery, New York, NY, USA, 579–586. DOI:https://doi.org/10.1145/1835449.1835546

[11] C. Beecks, M. S. Uysal and T. Seidl, "A comparative study of similarity measures for content-based multimedia retrieval," *2010 IEEE International Conference on Multimedia and Expo*, Suntec City, 2010, pp. 1552-1557, doi: 10.1109/ICME.2010.5582949.

[12] Anna Rizzo, Katerina El Raheb, Sarah Whatley, Rosa Maria Cisneros, Massimiliano Zanoni4, Antonio Camurri, Vladimir Viro, Jean-Marc Matos, Stefano Piana, Michele Buccoli, Amalia Markatzi, Pablo Palacio, Oshri Even Zohar, Augusto Sarti, Yannis Ioannidis and Edwin-Morley Fletcher. 2018. WhoLoDancE: Whole-body Interaction Learning for Dance Education. In *Proceedings of the Proceedings of the Workshop on Cultural Informatics (CI 2018)* , November 2018,. CEUR Workshop Proceedings Cyprus. https://doi.org/10.5281/zenodo.1478033

[13] Katerina El Raheb, Nicolas Papapetrou, Vivi Katifori, Yannis Ioannidis. 2016. BalOnSe. In *Proceedings of the Proceedings of the 3rd International Symposium on Movement and Computing (MOCO '16)*, July 05 – 07, 2016, Thessaloniki, GA. http://dx.doi.org/10.1145/2948910.2948926

[14] Januhari, N. U., Sarja, N. L. A. K. Y. and Rudita, M. INFORMATION TECHNOLOGY BASED MODEL FOR BALINESE CLASSICAL DANCE PRESERVATION. *International Journal of Engineering Technologies and Management Research*, 5, 11 (2018).

[15] Nyoman Utami Januhari, Ni Luh Ayu Kartika Yuniastari Sarja, and Made Rudita. 2018. INFORMATION TECHNOLOGY BASED MODEL FOR BALINESE CLASSICAL DANCE PRESERVATION, International Journal of Engineering Technologies and Management Research, 5(11), 71-85. DOI: https://doi.org/10.5281/zenodo.2280898

[16] P. Trivedi and A. Sharma, "A comparative study between iterative waterfall and incremental software development life cycle model for optimizing the resources using computer simulation," 2013 2nd International Conference on Information Management in the Knowledge Economy, Chandigarh, 2013, pp. 188-194.

[17] Jørgensen, M. Top-down and bottom-up expert estimation of software development effort. *Information and Software Technology*, 46, 1 (2004), 3-16.

[18] Montes, J. and Figueroa, P. VR salsa. In *Proceedings of the Proceedings of the IX Latin American Conference on Human Computer Interaction* (2019), Panama City. https://doi.org10.1145/3358961.3358969

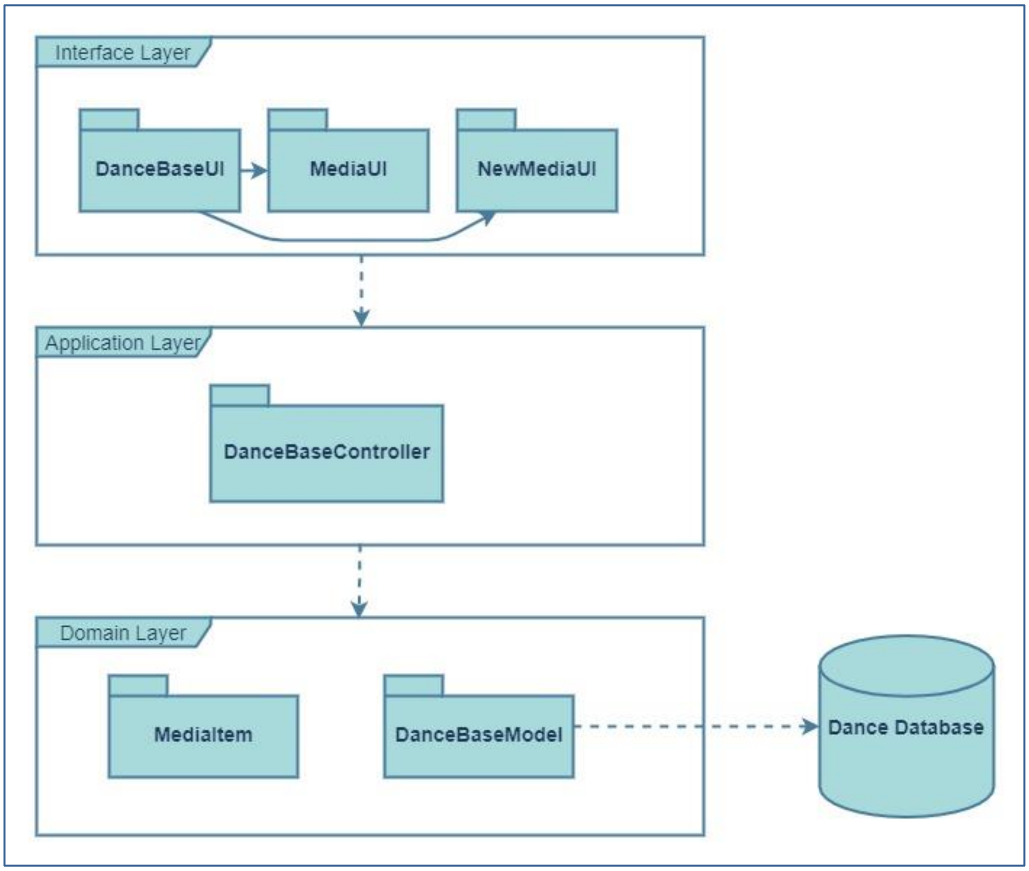
[19] Augusto Dias Pereira dos Santos, Kalina Yacef, and Roberto Martinez-Maldonado. 2017. Let's Dance: How to Build a User Model for Dance Students Using Wearable Technology. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. Association for Computing Machinery, New York, NY, USA, 183–191. DOI: https://doi.org/10.1145/3079628.3079673

Appendix A

Test Case	Description	Input	Behaviour	Expected output
1	Start the system and check if it opens onto the explore page	Open the project	Application starts up. Connects to database and shows recently uploaded media	Application opens and displays a list of media
2	Clicking on a media object takes user to a new interface of the media and its information.	User clicks on media object	Application gets media and associated information from database and displays it on a new interface	New interface opens with view of media object as well as information on style, dance moves, and move descriptions.
3	Videos and Audio play when clicked on	Click play button on video or audio	media begins playing after the button is clicked.	Video or audio starts playing
4	User adds media object to database	Media object and other attribute information as well as included dance moves	Application sends information to database to create new media object	Media object is added to the database and user gets a pop-up confirming upload
5	User can search for media using search bar	User inputs a search input relevant to media object meta data	Query is sent to database; media object is fetched and displayed search results on application interface	List of media objects relevant to search
6	Query Expansion Testing: application shows results based on the content of the media file	User inputs a search query	Search method queries database for media files which have a relationship with the dance move search input	Display media files which contain dance move inputted in the search bar.
8	Query Expansion Testing: application suggests media relevant to the input query	User inputs a search query	Relevant feedback method performed on query to determine similar/related queries and displays results	List results of search as well as relevant results similar to search query

Table 3: Test Plan for DanceBase System

Appendix B



Appendix C

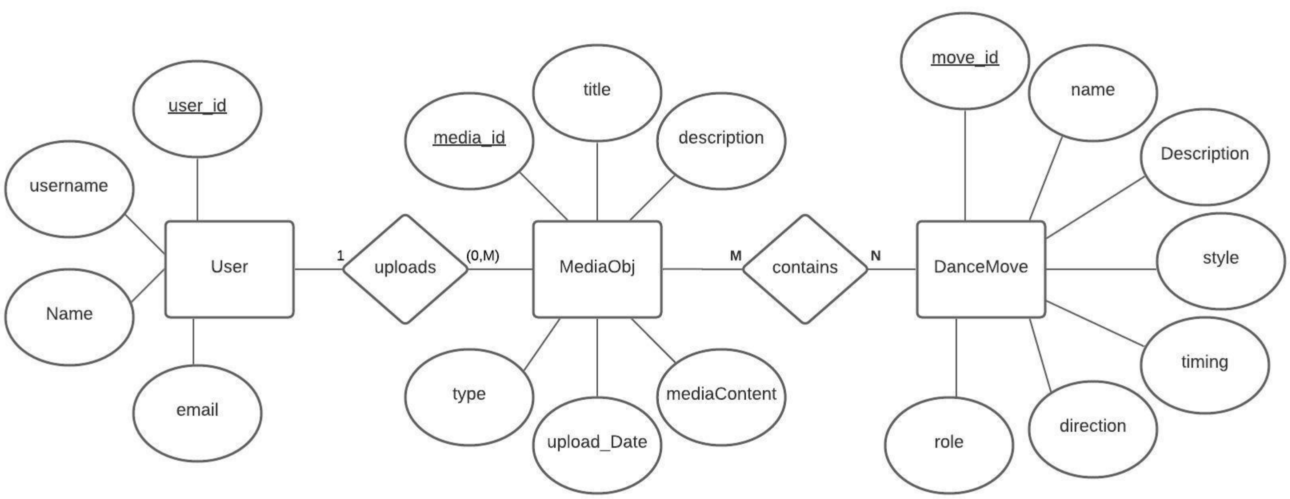


Figure 6: First model of DanceBase Multimedia Database

Appendix D

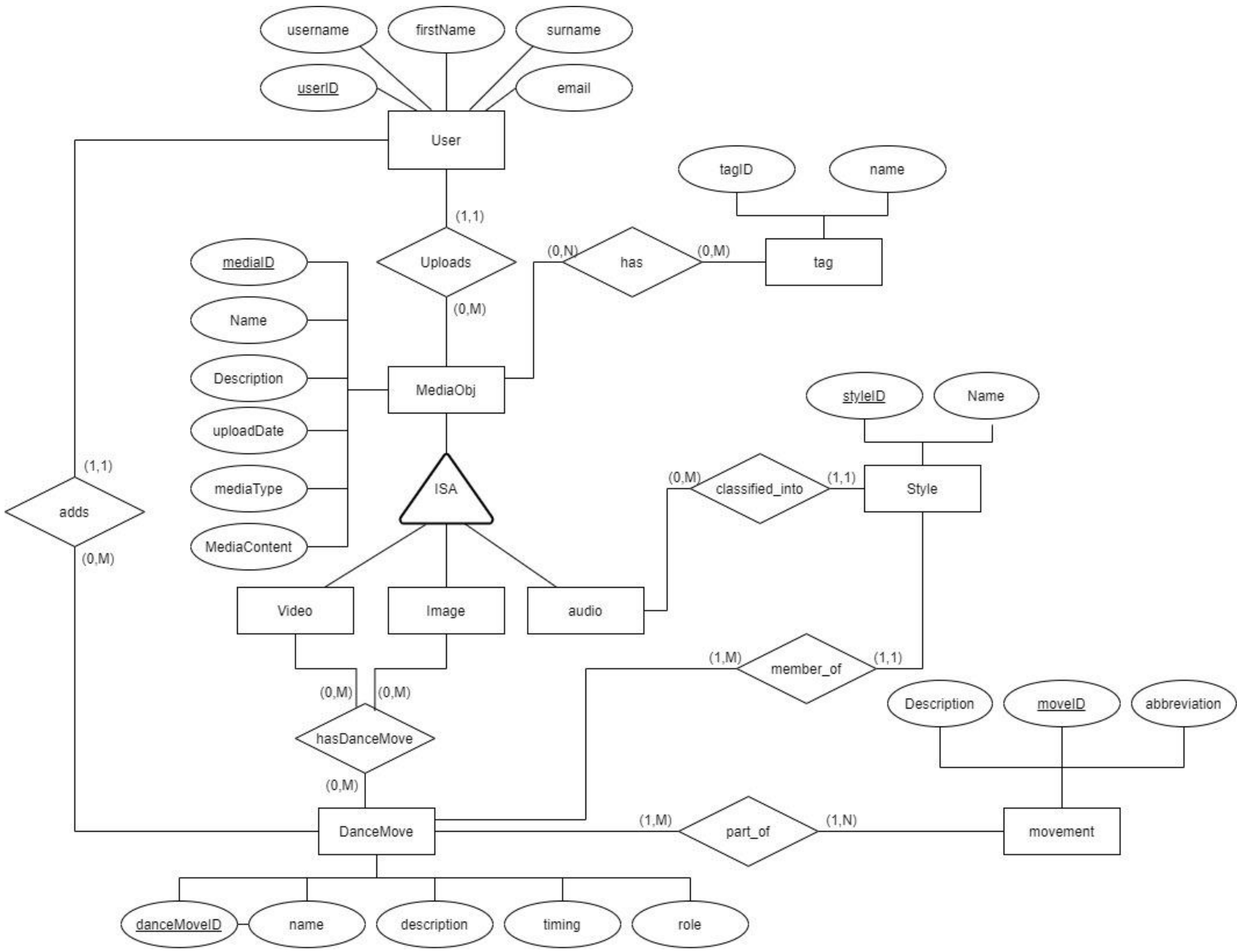


Figure 7: Second model of DanceBase Database Model

Appendix E

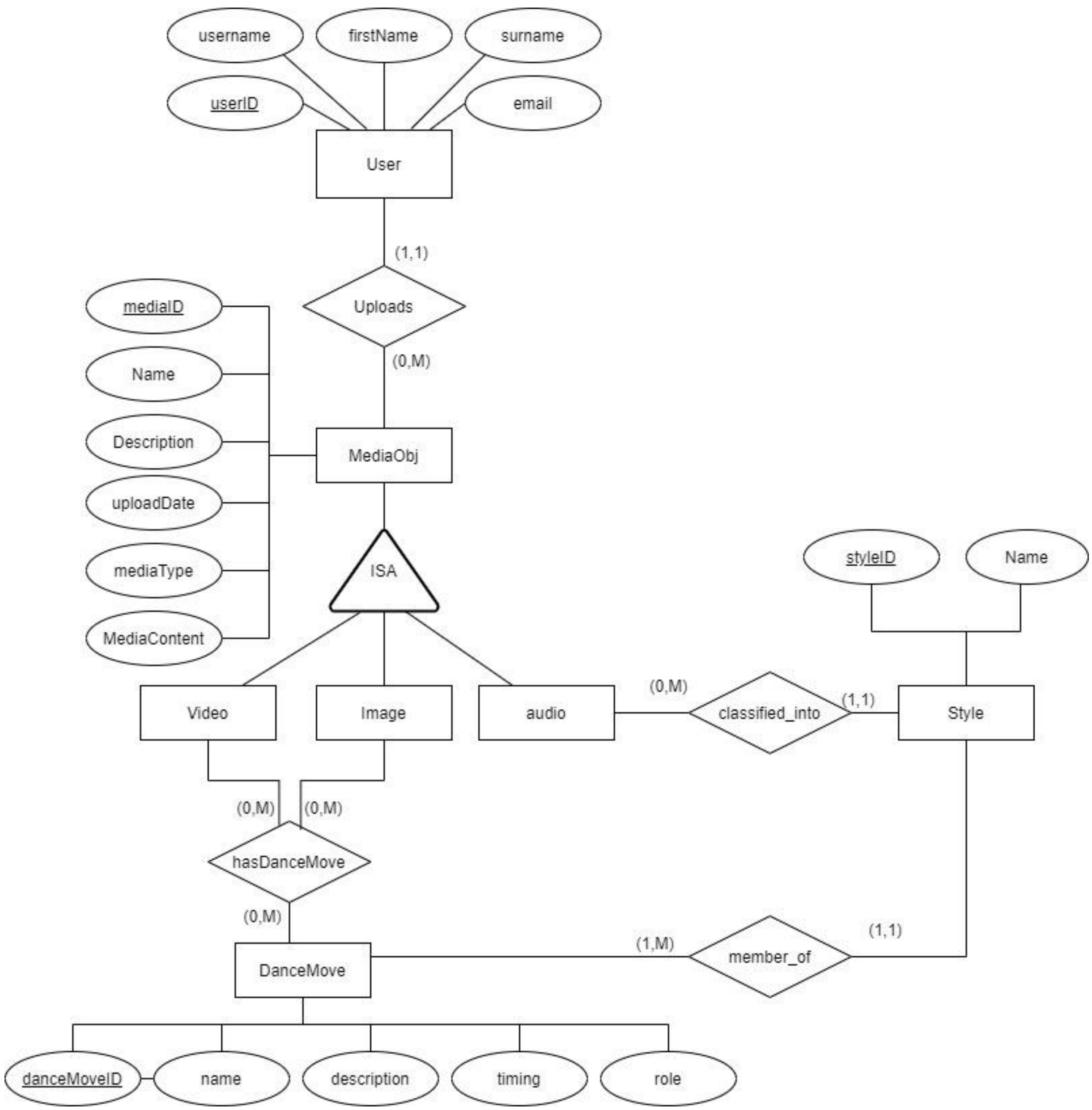


Figure 8: Final Model of DanceBase Database Model

Appendix F

Baseline Query								
No.	Query	Total Relevant	True Positive	False Positive	False Negative	True Negative	Precision %	Recall %
1	Salsa	13	8	0	5	17	100	62
2	Bachata	8	5	0	3	22	100	63
3	Bachata videos	2	2	3	0	25	40	100
4	Cha Cha audio	3	3	16	0	11	16	100
5	Let's get loud – Jennifer Lopez	1	1	0	0	29	100	100
6	Salsa side step	2	2	15	0	15	12	100
7	Bachata Choreography	1	1	6	0	23	14	100
8	Cha Cha tutorial	5	5	15	0	10	25	100
9	Tigeure Special	1	0	1	0	29	0	0
10	Side Break	1	1	6	0	23	14	100

Baseline and Content-Based Query								
No.	Query	Total Relevant	True Positive	False Positive	False Negative	True Negative	Precision %	Recall %
1	Salsa	13	8	0	5	17	100	62
2	Bachata	8	5	0	3	22	100	63
3	Bachata videos	2	2	3	0	25	40	100
4	Cha Cha audio	3	3	16	0	11	16	100
5	Let's get loud – Jennifer Lopez	1	1	0	0	29	100	100
6	Salsa side step	2	2	15	0	15	12	100
7	Bachata Choreography	1	1	6	0	23	14	100
8	Cha Cha tutorial	5	5	15	0	10	25	100
9	Tigeure Special	1	1	0	0	29	100	0
10	Side Break	1	1	6	0	23	14	100

Baseline and Pseudo-Relevance Feedback								
No.	Query	Total Relevant	True Positive	False Positive	False Negative	True Negative	Precision %	Recall %
1	Salsa	13	10	7	3	10	59	77
2	Bachata	8	5	11	3	11	31	63
3	Bachata videos	2	2	14	0	14	13	100
4	Cha Cha audio	3	3	18	0	9	14	100
5	Let's get loud – Jennifer Lopez	1	1	0	0	29	100	100
6	Salsa side step	2	2	17	0	13	11	100
7	Bachata Choreography	1	1	16	0	13	6	100
8	Cha Cha tutorial	5	5	16	0	9	24	100
9	Tigeure Special	1	1	29	0	0	3	0
10	Side Break	1	1	10	0	19	9	100

Baseline, Content-Based Query and Pseudo-Relevance Feedback								
No.	Query	Total Relevant	True Positive	False Positive	False Negative	True Negative	Precision %	Recall %
1	Salsa	13	10	7	3	10	59	77
2	Bachata	8	5	11	3	11	31	63
3	Bachata videos	2	2	14	0	14	13	100
4	Cha Cha audio	3	3	18	0	9	14	100
5	Let's get loud – Jennifer Lopez	1	1	0	0	29	100	100
6	Salsa side step	2	2	17	0	13	11	100
7	Bachata Choreography	1	1	16	0	13	6	100
8	Cha Cha tutorial	5	5	16	0	9	24	100
9	Tigeure Special	1	1	8	0	21	11	0
10	Side Break	1	1	10	0	19	9	100