# MMSR Report 2024

**Group A (Team Spontaneous Piroshki)**

**Sofiia Guchenko** **Milana Koval** **Darina Vorona** **Vladyslav Shevchuk** **Vladyslav Kravchenko**

## Contributions

Vladyslav Shevchuk was responsible for implementing the text-based retrieval system using TF-IDF, beyond-accuracy metrics, and investigating the trade-offs between NDCG and other metrics. Vladyslav Kravchenko was responsible for developing the web-based user interface and optimizing and evaluating the diversity metric. Darina Vorona worked on building the baseline system, calculating accuracy metrics (Precision@10, Recall@10, NDCG@10, MRR), and preparing the Lightning Talk presentation. Milana Koval and Sofiia Guchenko collaboratively worked on writing the report, focusing on the implementation and results sections.

## 1 Fundamentals

### 1.1 Baseline System

To establish a benchmark for evaluating our retrieval performance, we implemented a baseline system. This system randomly selects $N$ items from the dataset, excluding the query song itself. The random selection process provides a lower-bound benchmark for the retrieval system's accuracy, as the output is generated without considering the query song's attributes.

Mathematically, given a dataset $D$ of size $|D|$ and a query song $q \in D$, the baseline retrieval system produces a set $R \subset D \setminus \{q\}$ such that $|R| = N$ and $R$ is uniformly sampled from $D \setminus \{q\}$.

The input to the system consists of the *artist name* and *song title* of the query song. The output is a list of $N = 10$ songs randomly selected from the dataset, excluding the query song itself.

### 1.2 Accuracy Metrics

To assess the effectiveness of our retrieval systems, we employed four commonly used evaluation metrics: Precision@10, Recall@10, Normalized Discounted Cumulative Gain (NDCG@10) and Mean Reciprocal Rank (MRR). These metrics help us understand how well the system finds songs that belong to the same genre as the query song.

*Precision@10.* It tells us how accurate our search results are for the top 10 suggestions. In other words, out of the top 10 songs it shows us how many are actually relevant to what the user was looking for.

$$\text{Precision@10} = \frac{\sum_{i=1}^{10} \text{relevance}_i}{10}.$$

*Recall@10.* This metric looks at how complete our search results are. It measures what percentage of all the relevant songs we could have found are actually included in our top 10 suggestions.

$$\text{Recall@10} = \frac{\sum_{i=1}^{10} \text{relevance}_i}{|\text{Rel}|}.$$

*NDCG@10.* This metric considers not just whether a song is relevant, but also how high up in our list of suggestions it is. If a very relevant song is shown right at the top, this metric gives a higher score. It is like giving bonus points for showing the best matches first.

$$\text{NDCG@10} = \frac{\text{DCG@10}}{\text{IDCG@10}},$$

where:

$$\text{DCG@10} = \sum_{i=1}^{10} \frac{\text{relevance}_i}{\log_2(i+1)},$$

$$\text{IDCG@10} = \sum_{i=1}^{10} \frac{\text{relevance}_i^*}{\log_2(i+1)},$$

where *relevance*(i)* represents the ideal score for the item at the position *i* in the list. In other words, it shows how relevant the item should be if the list were perfectly ordered, with the most relevant items at the very top. This helps us understand the maximum possible relevance we could achieve within the top 10 results.

*Mean Reciprocal Rank (MRR).* It focuses on how quickly the first relevant song is found. If the first suggested song is a perfect match and appears at the top of the list (rank 1), the score is at its maximum value of 1. If we have to go down the list a long way to find a good match, the score is lower. Basically, it evaluates the rank position of the first relevant item.

$$\text{MRR} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{\text{rank}_q},$$

where: $Q$ - the set of all queries used to evaluate the system, $|Q|$ - the number of queries in the set $Q$, $q$ - an individual query from the set $Q$ and $\text{rank}_q$ - the position of the first relevant result in the ranked list for query $q$.

### 1.3 Text-Based Retrieval Using TF-IDF

In our system we used a method called TF-IDF to analyze song lyrics. TF-IDF helps us understand which words are most important in each song. It does this by considering how often a word appears in a song and how rare that word is across all the songs in our dataset.

Let $d$ denote a document corresponding to the lyrics of a song, and $t$ represent a term in the vocabulary. The TF-IDF weight of term $t$ in document $d$, using the $ltc$ variant, is computed as:

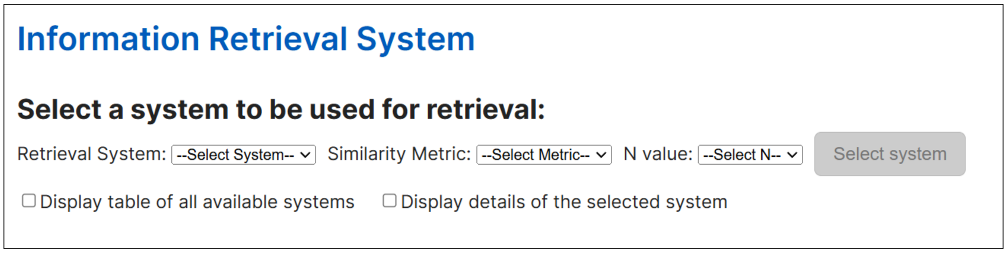
$$\text{tfidf}_{f_{d,t}} = \log(1 + f_{d,t}) \times \log\left(\frac{N}{f_t}\right),$$

where: $f_{d,t}$ - number of occurrences of term $t$ in document $d$ (term frequency), $N$ - total number of documents, $f_t$ - number of documents containing term $t$ (document frequency).

We employed also cosine similarity to evaluate our text-based retrieval system. However, the overall performance revealed that basic text-retrieved features perform mostly poorly, indicating limitations in capturing deeper semantic relationships within the data. The results are shown in the Subsection 1.5.

### 1.4 Web-based User Interface

Web-based User Interface (UI) was created and allows users to select an artist and song as a query input via an intuitive dropdown menu, retrieve a ranked list of similar songs based on the chosen retrieval method and display the results in an organized format, including song titles and corresponding artists. We also incorporated the ability to embed external resources, such as YouTube video links, for further exploration. Figure 1 shows how our homepage look like and Figure 2 shows an example of using our recommender.



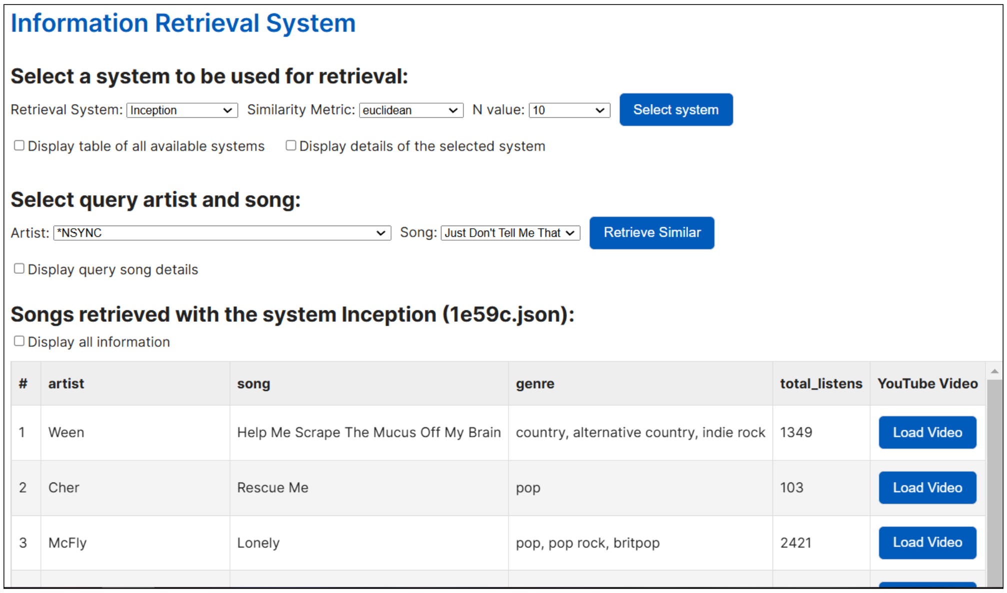
Information Retrieval System

Select a system to be used for retrieval:

Retrieval System: --Select System-- Similarity Metric: --Select Metric-- N value: --Select N--

□ Display table of all available systems □ Display details of the selected system

Figure 1: Web-based UI: Homepage



Information Retrieval System

Select a system to be used for retrieval:

Retrieval System: Inception Similarity Metric: euclidean N value: 10

□ Display table of all available systems □ Display details of the selected system

Select query artist and song:

Artist: *NSYNC Song: Just Don't Tell Me That

□ Display query song details

Songs retrieved with the system Inception (1e59c.json):

□ Display all information

| # | artist | song | genre | total_listens | YouTube Video |
|---|---|---|---|---|---|
| 1 | Ween | Help Me Scrape The Mucus Off My Brain | country, alternative country, indie rock | 1349 | Load Video |
| 2 | Cher | Rescue Me | pop | 103 | Load Video |
| 3 | McFly | Lonely | pop, pop rock, britpop | 2421 | Load Video |

Figure 2: Web-based UI: Results

### 1.5 Evaluation

The performance of the text-based retrieval system, using TF-IDF features and cosine similarity, was evaluated using four standard retrieval metrics: Precision@10, Recall@10, NDCG@10, and MRR. The results, presented in Table 1, provide insights into the system's effectiveness in retrieving relevant songs based on lyrics.

Table 1: TF-IDF Retrieval Evaluation Metrics

| Evaluation metric | Result |
|---|---|
| Precision@10 | 0.2066 |
| Recall@10 | 0.0289 |
| NDCG@10 | 0.5146 |
| MRR | 0.3663 |

Overall, the results show that while the system can find some related songs, its performance is limited by its reliance on simple word-counting techniques (like TF-IDF). The relatively low recall and precision scores suggest that the system may not fully capture the nuances of lyrical similarities between songs.

$$\text{Relevance}(\text{song}_i) = \begin{cases} 1, & \text{if } |\text{Genres}(\text{query}) \cap \text{Genres}(\text{song}_i)| \geq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Our retrieval system determines song relevance by assessing the overlap between the genres of the query song and the genres of the retrieved songs. A retrieved song is considered relevant if it shares at least one genre with the query song. This flexible approach accommodates a wide range of genre overlaps, acknowledging the diverse and evolving nature of musical genres. By considering even partial genre matches, the system can more accurately capture the subtle relationships between songs. This enhanced accuracy improves the reliability of evaluation metrics, such as *Recall@10*, which measure the system's ability to retrieve relevant songs.

## 2 Beyond-accuracy and Trade-offs

### 2.1 Additional text feature (BERT)

To improve upon basic TF-IDF features, we employed Bidirectional Encoder Representations from Transformers (BERT) as an additional text-based feature. BERT captures deeper semantic relationships between terms in song lyrics by leveraging contextual embeddings, unlike TF-IDF, which focuses on term frequency alone.

Table 2: Text Evaluation Results for BERT (100 samples)

| **Evaluation Metric** | **Cosine** | **Euclidean** |
|---|---|---|
| Precision@10 | 0.5520 | 0.5340 |
| Recall@10 | 0.0059 | 0.0055 |
| NDCG@10 | 0.7483 | 0.7323 |

The performance of two similarity metrics was evaluated (cosine similarity and Euclidean distance), using BERT-based embeddings on a dataset of 100 samples. Our experiments revealed superior performance for cosine similarity in terms of NDCG@10 compared to Euclidean distance (0.7483 vs. 0.7323, respectively). Additionally, cosine similarity achieved slightly higher values for Precision@10 (0.5520 vs. 0.5340), while both metrics exhibited similar values for Recall@10 (0.0059 vs. 0.0055). These results also shown in the Table 2 and indicate that cosine similarity is better suited for retrieval tasks involving BERT embeddings, as it more effectively captures semantic relationships in the data.

Also, in comparison to TF-IDF (Table 1) and BERT (cosine similarity Table 2), we can see that our experiments demonstrated an improvement in retrieval quality with BERT embeddings over TF-IDF. Specifically, BERT achieved a substantially higher NDCG@10 (0.7483 vs. 0.5146), highlighting its superior ability to rank relevant results. Furthermore, BERT outperformed TF-IDF in terms of Precision@10 (0.5520 vs. 0.2066), indicating better accuracy among the top 10 retrieved items. However, TF-IDF achieved a higher Recall@10 (0.0289 vs. 0.0059), suggesting that TF-IDF retrieves a larger absolute number of relevant items across the dataset, even if these items are ranked lower or appear less frequently in the top 10 results. This difference reflects the tendency of TF-IDF to favor broader retrieval coverage at the expense of ranking quality, rather than indicating increased diversity among the retrieved results.

### 2.2 Metrics for Audio and Visual modalities

#### 2.2.1 Metrics

We chose MFCC (Mel-Frequency Cepstral Coefficients) and Spectral Contrast as our audio features because they work well together and align with the goals of our retrieval task.

MFCC is a widely used feature that captures the tonal and timbral qualities of audio in a way that reflects human hearing. It is efficient in computing and great for identifying similarities in sound, like matching songs with similar tones.

Spectral Contrast complements MFCC by focusing on the harmonic structure of the audio, highlighting differences between peaks and valleys in the sound spectrum. This makes it particularly useful for distinguishing between genres or instruments, adding more depth to the retrieval system.

By combining these features, we get a balanced representation of audio: MFCC captures the overall sound, while Spectral Contrast focuses on finer harmonic details. Together, they help the system find relevant results while also supporting diversity in retrieval, which aligns perfectly with our task.

*Mel-Frequency Cepstral Coefficients (MFCC)*. This metric is a well-established feature used in audio analysis. It represents the spectral envelope of audio signals. MFCC coefficients are derived as follows:

1. Perform the Short-Time Fourier Transform (STFT) to obtain the frequency spectrum.
2. Map the frequency spectrum to the Mel scale using triangular filters:

$$f_{\text{mel}} = 2595 \cdot \log_{10}\left(1 + \frac{f_{\text{Hz}}}{700}\right),$$

where $f_{\text{mel}}$ is the frequency on the Mel scale, and $f_{\text{Hz}}$ is the original frequency in Hertz.

3. Apply the logarithm of the Mel-scaled amplitude spectrum.
4. Perform a Discrete Cosine Transform (DCT) to reduce the Mel-scaled spectrum to a finite set of coefficients (e.g., 13-40 coefficients).

*Spectral Contrast*. It highlights the differences between peaks and valleys in the sound spectrum. The steps to compute Spectral Contrast include:

1. Divide the spectrum into critical bands.
2. Compute the energy difference between spectral peaks (harmonic components) and valleys (non-harmonic components) for each band.
3. Normalize and aggregate results to form the spectral contrast vector.

Spectral Contrast provides a robust feature set for tasks like instrument recognition and genre classification.

### 2.2.2 Evaluation

The performance of two similarity metrics was evaluated, cosine similarity and Euclidean distance, for the MFCC and Spectral Contrast audio features on a dataset of 100 samples. For MFCC, the results showed comparable performance between the two metrics, with Precision@10 slightly higher for Euclidean distance (0.5490 vs. 0.5370), while both metrics exhibited identical values for Recall@10 (0.0030). However, NDCG@10 was marginally higher for cosine similarity (0.7300) compared to Euclidean distance (0.7221), indicating better ranking of relevant results using cosine similarity.

For Spectral Contrast, Euclidean distance demonstrated superior performance across all metrics. Specifically, Euclidean distance achieved higher Precision@10 (0.5820 vs. 0.5680), Recall@10 (0.0032 vs. 0.0031), and NDCG@10 (0.7512 vs. 0.7478) compared to cosine similarity. These results suggest that for Spectral Contrast, Euclidean distance is more effective at ranking and retrieving relevant results.

The evaluation results for audio features are summarized in Table 3.

Table 3: Audio Evaluation Results (100 samples)

| **Feature** | **Similarity Metric** | **Precision@10** | **Recall@10** | **NDCG@10** |
|---|---|---|---|---|
| MFCC | Cosine | 0.5370 | 0.0030 | 0.7300 |
| | Euclidean | 0.5490 | 0.0030 | 0.7221 |
| Spectral Contrast | Cosine | 0.5680 | 0.0031 | 0.7478 |
| | Euclidean | 0.5820 | 0.0032 | 0.7512 |

Further the performance of cosine similarity and Euclidean distance for visual features was analyzed, Inception Net and VGG19. For Inception Net, Euclidean distance outperformed cosine similarity across all evaluation metrics. Euclidean distance achieved slightly higher Precision@10 (0.5100 vs. 0.4980), Recall@10 (0.0025 vs. 0.0024), and NDCG@10 (0.7112 vs. 0.6909), highlighting its ability to retrieve and rank relevant visual results more effectively.

For VGG19, the performance of the two metrics was very similar. Cosine similarity achieved a marginally higher NDCG@10 (0.7076 vs. 0.7038), while Euclidean distance slightly underperformed in Precision@10 (0.4910 vs. 0.5020). Both metrics produced identical values for Recall@10 (0.0025). These results suggest that both metrics are equally suited for retrieval tasks involving VGG19 features, with minor differences in ranking effectiveness.

The evaluation results for visual features are presented in Table 4.

Table 4: Visual Evaluation Results (100 samples)

| **Feature** | **Similarity Metric** | **Precision@10** | **Recall@10** | **NDCG@10** |
|---|---|---|---|---|
| Inception Net | Cosine | 0.4980 | 0.0024 | 0.6909 |
| | Euclidean | 0.5100 | 0.0025 | 0.7112 |
| VGG19 | Cosine | 0.5020 | 0.0025 | 0.7076 |
| | Euclidean | 0.4910 | 0.0025 | 0.7038 |

## 3 Conclusion