BM25-Powered Sports Application: Algorithm Architecture

Author: Manus AI

Date: September 10, 2025

1. Introduction

This document outlines the algorithm architecture for a personalized sports media application that leverages the BM25 ranking algorithm to deliver relevant and timely content to users. The application will initially support the three major US sports (NFL, NBA, MLB) and is designed with a scalable architecture to accommodate additional sports in the future. This architecture is based on the application requirements detailed in `pasted_content.txt` and a comprehensive understanding of the BM25 algorithm from `BM25_TheCompleteGuidetoModernInformationRetrieval.docx`.

The core of this architecture is the strategic application of the BM25 algorithm across various agents to perform sophisticated ranking, filtering, and classification of sports-related content. By doing so, we can provide a highly personalized and relevant user experience, which is the central goal of the application.

2. Overall System Architecture

The system is designed as a multi-layered architecture with a clear separation of concerns, ensuring scalability, maintainability, and real-time performance. The architecture consists of four main layers: the Data Ingestion Layer, the Processing and Analytics Layer, the Application and API Layer, and the Presentation Layer.

2.1. Data Ingestion Layer

This layer is responsible for collecting raw data from a multitude of external sources. It will consist of a distributed network of web scrapers and crawlers, each tailored to specific source types (e.g., news websites, sports statistics APIs, ticketing platforms, social media). To ensure scalability and robustness, this layer will be designed with the following principles:

- **Source Abstraction:** Each data source will be represented by a standardized interface, allowing for easy addition of new sources without impacting the rest of the system.
- **Dynamic Configuration:** Scraper configurations will be stored in a central repository and can be updated dynamically, enabling rapid adaptation to changes in source website structures.

- **Rate Limiting and Politeness:** The crawlers will adhere to `robots.txt` and implement politeness policies to avoid overloading external servers.
- **Data Buffering:** Raw ingested data will be temporarily stored in a message queue (e.g., Apache Kafka or RabbitMQ) to decouple the ingestion process from the processing layer and handle bursts of data.

2.2. Processing and Analytics Layer

This is the core of the system where the ingested data is processed, enriched, and analyzed. The BM25 algorithm plays a central role in this layer. The key components of this layer are:

- **Content Extraction and Normalization:** Raw HTML and other data formats will be parsed to extract relevant content (e.g., article text, headlines, author, date). The extracted text will be normalized by converting it to a consistent format (e.g., lowercase, removing punctuation) to prepare it for BM25 indexing.
- **BM25 Indexing Engine:** A dedicated service will be responsible for creating and maintaining BM25 indexes for different content types and sports. This engine will be built using a high-performance information retrieval library like Apache Lucene or a modern Python implementation with optimizations for speed.
- **Content Classification Engine:** This engine will use a combination of BM25 and other machine learning models to classify content into predefined categories (injury reports, roster moves, trade rumors, etc.). This will involve training a supervised model on a labeled dataset of sports news.
- **Entity Recognition and Linking:** A named entity recognition (NER) service will identify and extract entities such as player names, team names, and locations from the text. These entities will be linked to a canonical knowledge base to ensure consistency and enable advanced filtering and search capabilities.
- **AI Summary Bot:** This component will leverage a large language model (LLM) to generate concise summaries of the top-ranked news articles for a given team. The selection of articles for summarization will be based on their BM25 scores.

2.3. Application and API Layer

This layer exposes the processed data and functionalities to the frontend application through a set of RESTful APIs. It will be responsible for handling user requests, authentication, and personalization. Key components include:

- **User Profile Service:** Manages user data, including preferred sports, favorite teams, and other settings.
- **Personalization Engine:** Tailors the content feed to each user based on their preferences and interaction history. This engine will use the BM25 scores as a primary

input for ranking and may incorporate other signals like recency and user feedback.

- **API Gateway:** Provides a single entry point for all client requests and handles routing, authentication, and rate limiting.

2.4. Presentation Layer

This is the user-facing part of the application, which will be a responsive web application or a native mobile app. It will consume the APIs provided by the Application and API Layer to display the personalized sports content to the user. The UI will be designed to be intuitive and user-friendly, with features like:

- A customizable dashboard showing the user's favorite teams and sports.
- A real-time feed of news, scores, and other content.
- Advanced search and filtering capabilities.
- Interactive visualizations of data, such as depth charts and player statistics.

3. BM25 Implementation by Agent

The strategic application of the BM25 algorithm is what will differentiate this application. Each agent will leverage BM25 in a unique way to fulfill its specific function. The following sections detail the BM25 implementation for each agent.

3.1. Scores Agent

The Scores Agent is responsible for providing accurate and timely score information. While seemingly straightforward, the use of BM25 can significantly enhance the reliability and relevance of the score data presented to the user.

- **Query Formulation:** The query for the Scores Agent will be a combination of the team name, the date, and keywords like "score," "final," "live," or "gamecast." For example, a query for a recent Lakers game might be `"Los Angeles Lakers" "score" "2025-09-10"`.
- **Corpus:** The corpus for the Scores Agent will be a curated collection of trusted sports data providers, including official league websites (NFL.com, NBA.com, MLB.com), major sports news outlets (ESPN, CBS Sports), and real-time score APIs.
- **BM25 Ranking:** BM25 will be used to rank the search results from the corpus. The ranking will prioritize official sources and pages that have a high density of score-related terms. The `k1` and `b` parameters of the BM25 algorithm will be tuned to favor shorter, more focused documents that are likely to contain just the score information, rather than long articles that mention the score in passing.
- **Historical Scores:** For historical scores, the query will be modified to include the specific date of the game. The BM25 ranking will help to quickly identify the most

relevant and accurate historical score data from the indexed corpus.

3.2. News Scraping Agent

The News Scraping Agent is the primary content discovery engine of the application. BM25 is the core of this agent, ensuring that users see the most relevant and important news about their favorite teams.

- **Query Formulation:** The query will be the full name of the user's favorite team (e.g., "Golden State Warriors"). To further refine the results, the query can be expanded to include the names of key players on the team.
- **Corpus:** The corpus for the News Scraping Agent will be a vast and continuously growing collection of sports news websites, blogs, and other online publications. The system will employ a sophisticated web crawling and discovery mechanism to identify new sources.
- **BM25 Ranking:** This is the most critical application of BM25 in the system. The algorithm will rank the scraped articles based on their relevance to the user's favorite team. The BM25 score will be a primary factor in determining the order in which news articles are displayed. The parameters of the BM25 algorithm will be tuned to balance term frequency (how many times a team or player is mentioned) with inverse document frequency (how unique those terms are).
- **Duplicate Detection:** A variation of BM25, or a similar hashing technique like MinHash, will be used to identify and filter out duplicate or near-duplicate news articles. This is crucial for providing a clean and non-repetitive user experience.

3.3. Content Classification Agent

The Content Classification Agent is responsible for the sophisticated sorting of news into specific categories: injury reports, roster moves, and trade rumors. This is a challenging task that goes beyond simple keyword matching, and BM25 is surprisingly effective for this purpose.

- **Multi-Corpus Approach:** Instead of a single large corpus, we will create a separate, curated corpus for each content category (injuries, roster moves, trade rumors). Each corpus will contain a collection of documents that are known to belong to that category. For example, the "injury" corpus will contain thousands of verified injury reports.
- **Classification as a Ranking Problem:** To classify a new, un-categorized article, we will treat it as a query and run it against each of the specialized corpora. The article will be assigned to the category of the corpus for which it receives the highest BM25 score. For example, if an article about a player being placed on the injured list gets a much higher

BM25 score when queried against the "injury" corpus than the "trade rumor" corpus, it will be classified as an injury report.

- **Training and Tuning:** The effectiveness of this approach depends on the quality and size of the specialized corpora. We will need to invest in creating and maintaining these corpora. The `k1` and `b` parameters of the BM25 algorithm will be tuned independently for each corpus to optimize classification accuracy.
- **Depth Chart Information:** Scraping and structuring depth chart information will be a more specialized task. While BM25 can help identify pages that contain depth charts, the primary challenge will be parsing the structured data from these pages. This will likely require custom parsers for each major data source.

3.4. Stadium Seat Agent

The Stadium Seat Agent helps users find the best ticket deals for their favorite team's games. BM25 can be used to rank ticket listings from various sources.

- **Query Formulation:** The query will include the team name, the opponent's name, the game date, and terms like "tickets," "seats," or "best deal."
- **Corpus:** The corpus will consist of pages from major ticketing websites (Ticketmaster, StubHub, SeatGeek) and the official team/league ticket exchanges.
- **BM25 Ranking:** BM25 will be used to rank the ticket listings. The ranking will take into account not only the relevance of the listing to the query but also other factors that can be encoded into the document, such as the price, seat location, and the number of tickets available. This will require a custom implementation of BM25 that can incorporate these additional features into the scoring function.

3.5. Fan Experience Agent

The Fan Experience Agent is designed to surface unique and engaging fan experiences beyond attending a game. The content for this agent is more varied, and BM25 can help to categorize and rank these experiences.

- **Query Formulation:** The query will be a combination of the user's location, their favorite sport, and keywords related to the different types of fan experiences described in the requirements document (e.g., "sports bar," "watch party," "virtual reality," "memorabilia show").
- **Corpus:** The corpus will be a curated collection of websites, articles, and listings related to the fan experiences mentioned in the `pasted_content.txt` file.
- **BM25 Ranking and Categorization:** Similar to the Content Classification Agent, we can use a multi-corpus approach to categorize the different fan experiences. A new experience can be classified by running it as a query against each of the specialized

corpora. Within each category, BM25 can be used to rank the experiences based on their relevance to the user's location and preferences.

4. Scalable Architecture for Multi-Sport Expansion

A key requirement for the application is the ability to scale and incorporate new sports beyond the initial three (NFL, NBA, MLB). The architecture is designed from the ground up to be modular and extensible, allowing for the seamless addition of new sports with minimal engineering effort.

4.1. Sport-Agnostic Core Components

The majority of the system's components are designed to be sport-agnostic. This means that the core logic for data ingestion, processing, and serving is not hard-coded to any specific sport. This includes:

- **Data Ingestion Framework:** The web crawling and data ingestion framework is generic and can be configured to scrape any website, regardless of the sport it covers.
- **Processing Pipeline:** The core processing pipeline, including the BM25 indexing engine and the API layer, is designed to handle data from any sport.
- **User Profile and Personalization Services:** The user profile service can accommodate any number of sports and teams. The personalization engine is also designed to work with any sport-specific data.

4.2. Sport-Specific Configurations

The key to the scalable architecture is the use of sport-specific configurations. For each new sport that is added to the system, we will create a new set of configuration files that define the unique characteristics of that sport. These configurations will include:

- **Data Sources:** A list of trusted data sources for the new sport, including news websites, statistics providers, and ticketing platforms.
- **BM25 Parameters:** The optimal `k1` and `b` parameters for the BM25 algorithm may vary from sport to sport. We will tune these parameters for each new sport to ensure the best possible ranking performance.
- **Content Classification Corpora:** For each new sport, we will need to create a new set of specialized corpora for the Content Classification Agent. This will involve collecting and labeling a dataset of articles for each content category (e.g., "soccer transfer rumors," "cricket injury reports").
- **Entity Lexicon:** We will create a lexicon of entities for the new sport, including team names, player names, and other relevant terms. This will be used by the Named Entity

Recognition (NER) service to identify and extract entities from the text.

4.3. Onboarding a New Sport: A Step-by-Step Process

Adding a new sport to the system will be a well-defined and semi-automated process:

1. **Configuration:** A new sport configuration file will be created, specifying the data sources, BM25 parameters, and entity lexicon for the new sport.
2. **Corpus Creation:** The Content Classification Agent's corpora for the new sport will be built by crawling and labeling a set of documents.
3. **Data Ingestion:** The data ingestion layer will be configured to start crawling the new data sources.
4. **Indexing:** The BM25 indexing engine will create a new set of indexes for the new sport.
5. **API Integration:** The API layer will be updated to expose the new sport's data to the frontend application.
6. **Frontend Update:** The frontend application will be updated to include the new sport in the user's list of available sports.

By following this modular and configuration-driven approach, we can add new sports to the application with minimal code changes, ensuring that the system can scale to cover the entire world of sports.

5. Conclusion

This document has outlined a comprehensive algorithm architecture for a personalized sports media application. By strategically leveraging the BM25 algorithm, the application can deliver a highly relevant, real-time, and scalable content experience to sports fans. The modular, sport-agnostic design ensures that the application can grow to encompass a wide variety of sports in the future, making it a truly global platform for sports information.

The success of this architecture will depend on the careful implementation of each component, particularly the tuning of the BM25 parameters and the creation of high-quality corpora for content classification. With a strong engineering team and a commitment to data quality, this architecture provides a solid foundation for building a world-class sports media application.

6. References

[1] Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, *3*(4), 333–389. https://doi.org/10.1561/1500000019

[2] Jones, K. S., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: Development and comparative experiments. *Information Processing & Management*, *36*(6), 779–808. https://doi.org/10.1016/S0306-4573(00)00015-7