

Profile-Based Retrieval

Information Retrieval, Extraction & Integration

Course 2022/23

Víctor Morcuende Castell

Guillermo Nájera Lavid

TABLE OF CONTENTS

[1. Introduction III](#_Toc131699812)

[2. Problem Context and Dataset Selection IV](#_Toc131699813)

[3. Preprocessing and Cleaning the Dataset V](#_Toc131699814)

[4. Profile’s Creation VI](#_Toc131699815)

[**Balancing the dataset** VI](#_Toc131699816)

[**Topics** VII](#_Toc131699817)

[**TF-IDF and WTF for vectorizing documents and profiles** VII](#_Toc131699818)

[**Creation of profiles** VIII](#_Toc131699819)

[5. Profile-based Retrieval Implementation IX](#_Toc131699820)

[**Representing User Profiles** IX](#_Toc131699821)

[**Representing Documents** IX](#_Toc131699822)

[**Similarity Calculation** IX](#_Toc131699823)

[**Ranking and Recommendation** X](#_Toc131699824)

[6. Performance Assessment X](#_Toc131699825)

# **Introduction**

The proliferation of digital content on the internet has led to an ever-increasing volume of information available to users. While this provides access to a vast source of knowledge, it also introduces the challenge of efficiently finding relevant and personalized content. In response to this challenge, profile-based information retrieval systems have emerged as an effective solution for tailoring content recommendations to individual users based on their unique interests and preferences.

Profile-based information retrieval systems construct user profiles that encapsulate users’ interests, preferences, and previous interactions with content. These profiles serve as the basis for personalized content recommendations, enabling the system to present users with content that is more relevant to their individual preferences. Such systems have widespread applications across various domains, including news websites, e-commerce platforms, and entertainment services.

Diagram

Description automatically generated

Figure 1: Profile-based information retrieval system

The primary objective in this assignment is to design, implement, and evaluate a content-based recommendation system that identifies and recommends small text snippets (or articles) tailored to user profiles. The system aims to accurately discern articles that are relevant to users’ interests, resulting in a more personalized and engaging user experience. In the pursuit of this goal, we will explore different techniques for constructing user profiles, such as the “Weighted Topic Frequency” (WTF) method and assess the efficacy of various classifiers in pinpointing relevant content.

Our project will comprise several stages, beginning with data preprocessing and feature extraction, followed by user profile creation using the WTF method. Subsequently, we will employ similarity measurements to establish connections between user profiles and documents, leading to personalized content recommendations. Finally, we will evaluate the system’s performance using a range of metrics, assessing the results obtained from multiple classifiers.

In the forthcoming sections, we will delve into the intricacies of our content-based recommendation system, providing a comprehensive overview of each stage in the process. The data preprocessing and feature extraction phases will be outlined, followed by a detailed explanation of user profile creation using the WTF method. Next, we will discuss the method for measuring similarity between profiles and documents, which forms the foundation for generating content recommendations. The system evaluation will then be presented, encompassing the various performance metrics employed, as well as an in-depth analysis of the results obtained from different classifiers. Through this extensive exploration, we aim to provide a thorough understanding of our profile-based information retrieval system and its potential for delivering highly personalized content recommendations.

# **Problem Context and Dataset Selection**

To tackle the problem described in the previous section, we require a dataset that contains a substantial number of articles with sufficient information to construct meaningful user profiles and develop a robust recommendation system. For this purpose, we have chosen the “BBC News Train” dataset, which is a collection of news articles from the BBC. The dataset comprises 1.490 articles spanning across five different categories: business, entertainment, politics, sport, and tech.

Text

Description automatically generated

Figure 2: Dataset Appearance

The choice of the BBC News Train dataset offers several advantages. Firstly, the diverse range of topics covered in the dataset allows us to create user profiles that encompass a variety of interests, making it suitable for developing a versatile recommendation system. Secondly, the dataset contains high-quality and well-structured articles, ensuring that our system can accurately capture the fine differences of the content. Additionally, the dataset has been used in numerous research studies and machine learning competitions, which facilitates the comparison of our results with existing work in the field.

# **Preprocessing and Cleaning the Dataset**

To develop an effective profile-based information retrieval system, it is essential to preprocess and clean the dataset to ensure that it is in a suitable format for analysis. This process involves several steps:

1. Importing the dataset: the first step is to import the dataset into our Python environment. We read the CSV file containing the BBC News Train dataset and stored it in a suitable data structure, such as a Pandas DataFrame, for easy manipulation and analysis. This allowed us to access and process the data in an efficient and intuitive manner.
2. Removing duplicate articles: to ensure that our dataset is accurate and representative, we removed any duplicate articles that may be presented, as duplicate articles can lead to biased results and affect the performance of our recommendation system. We achieved this by identifying and removing rows with identical values in the “Text” column using the ‘drop\_duplicates()’ function available in Pandas. This helped us maintain the integrity of the dataset and prevented any distortions in the results of our analysis. Moreover, as it can be appreciated below, the dataset contains an unbalanced number of articles for each topic:

Chart, bar chart

Description automatically generated

Figure 3: Unbalanced Dataset

1. Text cleaning: as the dataset contains textual data, it is crucial to clean and preprocess the text to ensure it is in a suitable format for analysis. Therefore, to carry out this process we performed the following steps:
   1. Removing punctuation marks: punctuation marks can introduce noise and make it difficult to process the text. By removing them, we simplified the text and facilitated the analysis.
   2. Converting the text to lowercase: this helped us to ensure that the text is consistent and that words are not treated as separate entities due to differences in case.
   3. Tokenization: breaking the text into individual words (or tokens), enabled us to analyze the text at the word level and helped us in identifying meaningful patterns and relationships. Furthermore, we removed the tokens whose length was lower than 3 characters, since typically these words do not provide useful information and they would not help us to achieve our goals.
   4. Removing stop words: commonly used words such as “and”, “the”, and “in” do not carry much meaning and can be safely excluded from the analysis. By applying this change, we reduced the size of the dataset and improved the efficiency of subsequent processing steps.
   5. Lemmatization: this technique reduces words to their dictionary base forms, further simplifying the text and reducing the number of unique words in the dataset, which could help us in our aim of improving the performance and efficiency of the analysis.
2. Creating a clean dataset: after completing the preprocessing and cleaning steps, we were left with a clean dataset that was ready for further analysis. This dataset was now free of duplicates, missing values, and irrelevant information, and the textual data has been preprocessed and simplified to ensure it is in a suitable format for analysis. With this clean dataset, we can proceed to develop our profile-based information retrieval system, extract relevant features, and evaluate its performance using appropriate metrics and evaluation techniques.

# **Profile’s Creation**

After that, we started with the creation of user profiles, which is a pivotal component in a profile-based information retrieval system. This section delves deeper into balancing the dataset, the reasoning behind selecting the TF-IDF and WTF methods for vectorizing documents and profiles, and the formation of topics and profiles.

**Balancing the dataset**

In the context of a multi-class classification problem, it is crucial to maintain a balanced dataset to prevent potential biases in the model’s performance. These biases could lead to overemphasis on certain categories and neglect of others, thereby compromising the model’s accuracy and efficiency. For this reason, we reduced the size of each topic by the size of the smallest one, which in this case was the tech topic with 234 articles. By ensuring that an equal number of articles are presented for each category, the model is consequently exposed to a representative sample of each topic, which in turn enhances its ability to discern between different categories and provide more accurate results.

Chart, bar chart

Description automatically generated

Figure 4: Balanced Dataset

## **Topics**

The selected dataset comprises news articles from five distinct categories: business, entertainment, politics, sport, and tech. Therefore, to create the user profiles, a set of keywords was identified for each category or topic. These keywords encapsulate the topic’s primary themes and act as the foundation for defining user interests.

To carry out the keywords selection, we performed an analysis of the dataset, retrieving the 100th most frequent features (words) of each category. Then, we conducted an analysis of these features and filtered them for each topic, keeping only the 50 most important ones, from our point of view. As a result, we ended up with 50 keywords for each of the categories.

## **TF-IDF and WTF for vectorizing documents and profiles**

The Term Frequency-Inverse Document Frequency (TF-IDF) method was chosen for vectorizing the documents and user profiles due to its robust and widely accepted performance in text representation. TF-IDF’s popularity stems from its ability to convert text into numerical format, a requirement for machine learning algorithms to effectively process and analyze data. This technique calculates the importance of each term within a document based on its frequency in the document and its rarity across the entire corpus. Consequently, it helps identify terms that are significant and distinctive, which can then be utilized as features to differentiate between various topics. The TF-IDF matrix will be later used to build the user profiles by associating the users’ preferred topics with the weighted terms in the articles.

In addition, after obtaining the TF-IDF representation of the news articles, we implemented the “Weighted Topic Frequency” (WTF) method, an algorithm that we developed to create user profiles that represent the users’ interests and preferences. The WTF method is a technique for representing user interests by analyzing their interactions with documents belonging to specific topics. In this approach, we assign a weight to each topic based on the user’s interaction with the articles and use this information to create user profiles. The weight represents the importance of the topic for the user and is determined by the number of times the user has interacted with articles from that topic. By identifying user preferences and interests in the different topics, we can create comprehensive profiles of their preferences. To explain this method better, the steps followed to implement it are the following:

1. Term Frequency (TF): first, we use a function which counts the frequency of each word in the profile’s interests and normalize it by the total number of words in that same profile’s interests.
2. Inverse Topic Frequency (ITF): then, another method is used, in which, for each word in the profile’s interests, its presence in all topics is calculated. Then, it computes the inverse of this presence (total number of topics / number of topics containing the word). This will give as a result higher weights to words that are more unique to a particular profile’s interests.
3. Weighted Topic Frequency (WTF): finally, by multiplying the TF and ITF functions for each word, the WTF value is obtained. This will emphasize words that are both frequent in the profile’s interests and unique to their topics.

## **Creation of profiles**

By combining the TF-IDF and WTF methods, a more precise representation of the user’s interests can be achieved, as the WTF method accounts for the term’s significance within the topic, while the TF-IDF method considers its significance across the entire corpus. This combination leads to a more robust and accurate user profile.

Therefore, after the detailed processes mentioned above, we combined the WTF method and the TF-IDF matrix to create accurate user profiles that capture the users’ interests and preferences within the context of the five predefined topics. We did this by associating the users’ preferred topics, as determined by the WTF method, with the weighted terms in the articles derived from the TF-IDF representation. This results in detailed user profiles that can be used to provide personalized recommendations and enhance the information retrieval process.

Consequently, we created several user profiles based on the five predefined topics (business, entertainment, politics, sport, and tech). Being precise, we decided to create first a profile for each topic, that is, the user would only be interested in one topic:

* Profile 1: sport
* Profile 2: business
* Profile 3: entertainment
* Profile 4: politics
* Profile 5: tech

After that, we opted to create profiles by combining different categories for the purpose of analyzing and evaluating how our weights’ methods (TF-IDF and WTF) worked:

* Profile 6: sport and business
* Profile 7: entertainment and politics
* Profile 8: tech and sport
* Profile 9: business and entertainment
* Profile 10: politics, tech and business

# **Profile-based Retrieval Implementation**

In this section, we discuss the implementation of the profile-based retrieval system using the user profiles created in the previous step. The primary objective of our system is to deliver personalized content recommendations to users based on their interests and preferences.

## **Representing User Profiles**

First, the user profiles generated earlier are represented as vectors containing the weighted terms derived from the combination of the WTF method and the TF-IDF matrix. These vectors effectively capture the users’ preferences across the five predefined topics in the BBC News dataset, enabling us to make personalized recommendations.

## **Representing Documents**

Then, similar to the user profiles, we represented the news articles in the dataset as vectors using their TF-IDF values. This representation allowed us to quantify the relevance of each article to a particular user by comparing their vector representations.

## **Similarity Calculation**

To determine the relevance of a news article to a specific user, we computed the similarity between the user’s profile vector and the document’s vector representation. To accomplish this, we used the cosine similarity measure, which calculates the cosine of the angle between the two vectors. The cosine similarity ranges from -1 (completely dissimilar) to 1 (identical), with higher values indicating a stronger similarity between the user’s profile and the particular document. This similarity score forms the basis for our content recommendations.

Diagram

Description automatically generated

Figure 5: Cosine Similarity Explanation

## **Ranking and Recommendation**

After calculating the cosine similarity between each user’s profile vector and the document vectors, we proceeded to rank the articles based on their similarity scores. The top-ranked documents were considered to be the most relevant to the user’s preferences and are recommended accordingly. This ranking system ensured that the content recommendations were tailored to each user’s unique interests, providing a more personalized and satisfying information retrieval experience.

Text

Description automatically generated

Figure 6: IR System Implementation

# **Performance Assessment**

After computing the cosine similarity between user profiles and documents, we needed to carry out a proper evaluation phase to assess the system’s performance. The reason behind is that evaluating the system’s performance is crucial to understand its effectiveness in providing relevant and personalized content recommendations to users. We used various evaluation metrics, such as accuracy, precision, recall, F1-score, and ROC curves. Each of these metrics provides insight into different aspects of the system’s effectiveness:

* Accuracy: the ratio of correctly predicted instances to the total instances in the dataset. It is a commonly used evaluation metric to gauge the overall performance of a classification system. In our context, accuracy measures how well our retrieval system recommends relevant articles/documents to users based on their profiles.
* Precision: the ratio of true positive predictions to the sum of true positive and false positive predictions. In the context of our profile-based retrieval system, precision measures the proportion of relevant documents among the recommended articles. A high precision score indicates that the system is effective in providing relevant content recommendations to users.
* Recall: the ratio of true positive predictions to the sum of true positive and false negative predictions. In our system, recall measures the proportion of relevant articles that were successfully recommended to users out of all the relevant articles available. A high recall score suggests that the system is efficient in capturing most of the relevant content for each user.
* F1-score: the harmonic mean of precision and recall, providing a single metric that combines the two. This metric is particularly useful when dealing with imbalanced datasets, where one class might be under-represented. The F1-score allows us to assess the trade-off between precision and recall, with higher values indicating a better balance between the two.
* ROC Curves: Receiver Operating Characteristic (ROC) curves are graphical representations of the true positive rate (sensitivity) against the false positive rate (1-specificity) for different classification thresholds. The area under the ROC curve (AUC-ROC) quantifies the system’s ability to distinguish between relevant and irrelevant articles across various threshold values. A higher AUC-ROC score indicates that the system has a better discriminative power, which is essential for providing personalized content recommendations.

Chart, line chart

Description automatically generated

Figure 7: ROC Curves of all the categories

7. Model Creation and Evaluation

8. Results

9. Conclusions