Boosting Job Matching Accuracy: Implementing Content-Based Filtering in Job Applicant Recommendation Systems

Rasyid Alim Aulia[1], SY. Yuliani[2], Angga Aditya Permana[3]

[1, 2, 3] Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia

Abstract. The selection process after the recruitment stage becomes a challenge for recruiters in verifying the suitability of candidates' qualifications for the required positions. Currently, the selection process still relies on Microsoft Excel as a tool for keyword searches in candidate data, requiring approximately 30 seconds to process a single keyword. The implementation of the Content-Based Filtering method in the applicant recommendation system is highly suitable for enhancing the recruitment process efficiency, as it focuses on analyzing the similarity of descriptions in the curriculum vitae data uploaded by candidates. This system operates by calculating the similarity of candidate data based on the entered keywords. The system development process includes several stages, such as data preprocessing, Term Frequency–Inverse Document Frequency (TF-IDF) calculation, Cosine Similarity computation, and ranking the results based on Cosine Similarity values from highest to lowest. The experiment results show that searching for a single keyword takes approximately 0.7 seconds, which is 29.3 seconds faster than the current candidate selection process. Additionally, a user satisfaction survey evaluated using the DeLone and McLean model, achieved a score of 95.4%, indicating that the system effectively enhances the accuracy and efficiency of applicant recommendations in the recruitment process.

Keywords: Applicant Recommendation System, Content-Based Filtering, Cosine Similarity, Term Frequency–Inverse Document Frequency, Recruitment Process Efficiency.

1 Introduction

The selection process involving a large number of applicants requires significant time and effort to verify their qualifications and suitability for the desired positions [1]. To process a single keyword takes approximately 30 seconds or more, depending on the number of keywords and candidates to be selected. Technology plays a vital role in addressing this challenge by streamlining and accelerating recruitment processes [2]. Technological advancements help optimize several selection stages and improve the quality of decisions made by recruitment teams [3]. They also contribute to creating a more transparent and efficient recruitment process [4, 5]. While minimizing the potential for subjective evaluations that may occur in manual selection processes [6]. Therefore, a system capable of providing applicant recommendations is essential.

This study employs the Content-Based Filtering method, where the system calculates specific attributes or features possessed by applicants, such as skills, work experience, and education [7]. The system then compares these attributes with existing data to calculate the degree of similarity based on user input [8, 9]. A key advantage of Content-Based Filtering is its independence from other users' data, as the system evaluates similarity solely based on individual applicant data [10]. This enables the system to deliver more specific recommendations by relying exclusively on internal data relevant to applicant searches [11].

Currently, research on applicant recommendation systems utilizing the Content-Based Filtering approach remains limited. According to references on applicant recommendation systems, methods such as Simple Adaptive Weighting (SAW) and Simple Multi-Attribute Rating Technique (SMART) are more frequently used for decision support systems [12, 13]. Although SAW is simpler to implement, it has a drawback in terms of subjectivity in weighting [12, 14]. In this study, Cosine Similarity is used to calculate similarity by measuring the angle between two vectors [9, 15]. Additionally, the TF-IDF method is used for its efficiency, simplicity, and accuracy in word weighting [16].

2 Methodology

2.1 Business Design

As shown in Fig. 1, the program first executes the Recommendation function to provide applicant suggestions based on user preferences from previous searches. Recommended applicant data is stored as suggestion applicants and displayed through the recommendation feature. When searching for applicants, users are prompted to enter search keywords. The system then runs the Search Applicants function to process the search and display matching candidates in a table. If no data is found, the system returns a "Data not found" message. The recommendation feature can display up to three suggested applicants. Additionally, users can reset the stored search keywords to view the full list of applicants.

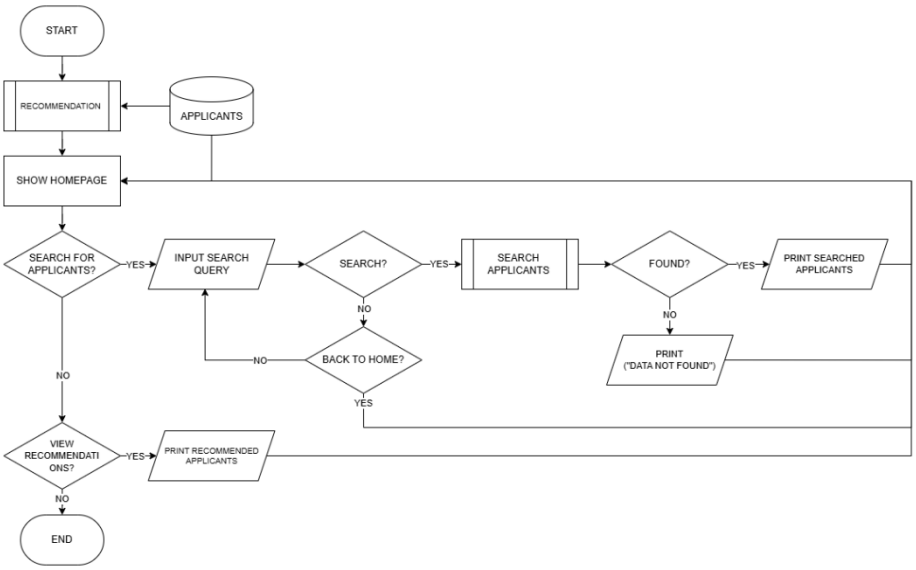

Fig. 1. Business design

2.2 Content-based Filtering Flowchart

As illustrated in Fig. 2, the Content-Based Filtering process consists of seven interconnected subprocesses that run sequentially: Case Folding, Stopword Removal, Tokenizing, and Stemming, which are part of the word preprocessing stage in the document; TF-IDF, which assigns weights to words to determine their importance based on their frequency of occurrence; Cosine Similarity, which is used to calculate the similarity between two vectors; and finally, the recommendation process, which is performed by sorting the cosine similarity values from highest to lowest.

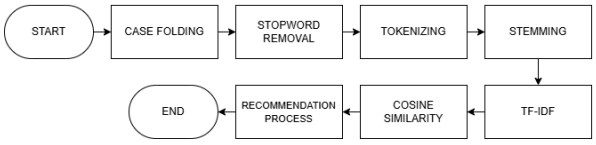

Fig. 2. Content-based filtering flowchart

3 Related Study

3.1 Recommendation System

A recommendation system is designed to provide suggestions or predict relevant information based on user preferences. This system aims to assist users in finding information or items that meet their needs [17]. In the development of recommendation systems, several methods are commonly used, including Collaborative Filtering,

Content-Based Filtering, Demographic Filtering, and Hybrid Filtering [18]. The use of the Content-Based Filtering method remains relatively rare. However, this method holds significant potential for delivering relevant recommendations, especially when applicant data is available in sufficient quantities and structured properly [19].

3.1.1 Collaborative Filtering Method

Collaborative Filtering is one of the most widely used methods in system development [18]. This method can overcome the cold start problem that often occurs with new users, allowing the system to generate more relevant recommendations by comparing preferences based on the ratings of other users [20]. To generate recommendations, this method uses all or part of the user-item database [18].

3.1.2 Content-based Filtering Method

Content-Based Filtering consists of several algorithms that compare features or attributes of specific items with user preferences [8, 21]. This method compares item descriptions with search inputs to measure similarity and utilizes previous search data to provide item recommendations [22]. However, this method faces limitations like the cold start problem, where recommendations are difficult without initial user or item data [23].

3.1.3 Demographic Filtering Method

In the Demographic Filtering method, the recommendation process is based on user demographic attributes. The information used includes nationality, age, gender, and other demographic factors. With the obtained information, the data is grouped into several segments to facilitate the similarity measurement process based on the user's personal data and the object to be recommended [24].

3.1.4 Hybrid Filtering Method

Hybrid Filtering is a method that combines multiple approaches to enhance the performance of recommendation systems [25]. This method is commonly used to address issues found in other filtering methods [26]. By combining the implicit relations between user preferences and additional taxonomic preferences, the system can generate higher-quality recommendations and address the cold-start problem [27].

3.2 Term Frequency – Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical method used to evaluate the importance of a word to a document within a collection of documents [28]. Term Frequency (TF) measures the frequency of a word's appearance in a document, while Inverse Document Frequency (IDF) calculates weighting values based on how rarely the word appears across all documents [29]. A higher TF-IDF score indicates that the word has a high level of relevance to a specific document [30]. Combining the TF-IDF method with the Cosine Similarity algorithm enables recommendation systems to accurately compute the

similarity between documents, resulting in relevant and appropriate recommendations. The TF-IDF formula is as follows.

$$\text{TF-IDF}(p,q)=\text{TF}(p,q)\times\text{IDF}(p) \quad (1)$$

Where:

a) Term Frequency (TF):

$$\text{TF}(p,q)=\text{Total word p in document}/\text{Total words in document q} \quad (2)$$

b) Inverse Document Frequency (IDF):

$$\text{IDF}(p)=\log\left(N/\text{DF}(p)\right) \quad (3)$$

3.3 Cosine Similarity

After the TF-IDF calculation is complete, similarity analysis can be conducted to assess how closely user input matches the data in the applicant database. The Cosine Similarity method is employed to measure the proximity of two vectors in vector space by calculating the angle between them. The smaller the angle, the higher the similarity level. The Cosine Similarity formula produces values between -1 and 1, where a value of 1 indicates very high similarity, 0 signifies no similarity, and -1 is opposite direction [31]. The formula for Cosine Similarity is as follows.

$$\text{CosSim}(A,B)=\frac{A\cdot B}{|A||B|}=\frac{\sum_{i=1}^{n}A_i\times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\times\sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (4)$$

4 Result and Discussion

4.1 Recommendation Process in Content-Based Filtering

Content-Based Filtering consists of seven interconnected subprocesses that operate sequentially. Each subprocess serves a specific and critical function to ensure optimal recommendation results. These subprocesses work together to produce an accurate and effective recommendation system that assists in the applicant selection process. The explanation for each subprocess is as follows [32].

a) Case Folding

The first stage involves Case Folding, which converts all text to lowercase and removes unnecessary characters, such as punctuation marks and other symbols. This step ensures data consistency by eliminating differences between uppercase and lowercase letters, which can often cause errors in text analysis. An example of Case Folding is illustrated in Table 1.

Table 1. Case folding process

BEFORE	AFTER
Software Engineer, Programming, and Analyst	software engineer programming and analyst

b) Stopword Removal

The system performs Stopword Removal by eliminating commonly used words that do not carry significant meaning in analysis, such as conjunctions, articles, and prepositions. This process filters out non-informative words and improves the accuracy of text processing. An example of Stopword Removal is illustrated in Table 2.

Table 2. Stopword removal process

BEFORE	AFTER
software engineer programming and analyst	software engineer programming analyst

c) Tokenizing

Tokenizing is the process of breaking text into its smallest units called tokens. Without tokenizing, text becomes difficult to process due to a lack of structure. Therefore, tokenizing serves as a foundation for more in-depth text analysis, such as keyword searches, sentiment analysis, or text classification. An example of Tokenizing is illustrated in Table 3.

Table 3. Tokenizing process

BEFORE	AFTER
software engineer programming analyst	[“software”, “engineer”, “programming”, “analyst”]

d) Stemming

The next step focuses on removing affixes such as prefixes or suffixes to obtain the root form of a word by invoking a stemming function, as illustrated in the pseudocode. This ensures that the system recognizes a single entity even if words are used in different contexts. An example of Stemming is shown in Table 4.

Table 4. Stemming process

BEFORE	AFTER
“software”, “engineer”, “programming”, “analyst”	software engineer program analyst

e) Term Frequency – Inverse Document Frequency (TF-IDF)

At this stage, the weight or importance of a word in a document is calculated based on its frequency of occurrence (Term Frequency) and the rarity of its appearance across other documents (Inverse Document Frequency). This calculation can be performed using the TfidfVectorizer function. The Term Frequency (TF) calculation is as follows:

$$TF(software, 1) = \frac{1}{93} = 0.0108$$

The overall TF calculation results are as shown in Table 5.

Table 5. TF calculation

DOC.	SOFTWARE	ENGINEER	PROGRAM	ANALYST
0 (query)	0.25	0.25	0.25	0.25
1	0.0108	0.0108	0	0
2	0	0	0	0.0294
3	0	0	0	0
4	0.0096	0	0	0
5	0	0	0.0098	0.0196
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0.0182	0	0
10	0	0	0.0096	0.0096

Additionally, the Inverse Document Frequency (IDF) calculation is as follows:

$IDF(software) = \log(10/2) = 0.69897$

$IDF(engineer) = \log(10/2) = 0.69897$

$IDF(program) = \log(10/2) = 0.69897$

$IDF(analyst) = \log(10/3) = 0.522878$

IDF is computed using a logarithmic formula that compares the total number of documents (N) to the number of documents containing a particular word (DF(p)). Words that rarely appear across the analyzed documents are considered more significant in distinguishing one document from another. Thus, the TF-IDF calculation is obtained as follows:

$$TF - IDF(software, 1) = 0.0108 \times 0.69897 = 0.00754$$

The overall TF-IDF calculation results are as shown in Table 6.

Table 6. TF-IDF calculation

DOC.	SOFTWARE	ENGINEER	PROGRAM	ANALYST
0 (query)	0.17474	0. 17474	0. 17474	1.30719
1	0.00754	0. 00754	0	0
2	0	0	0	0.01537
3	0	0	0	0
4	0.00671	0	0	0
5	0	0	0.00685	0.01025
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0.01272	0	0
10	0	0	0. 00671	0.00502

f) Cosine Similarity

The method used to calculate similarity is Cosine Similarity, which measures the closeness between two vectors in a vector space based on the angle between them. The smaller the angle between the two vectors, the higher their similarity. For example, the similarity between the query (A) and Applicant 1 (AP1) is calculated as follows:

$$AP1 = CosSim(0,1) = (0.00131753 + 0.0013175)/(\sqrt{0.122136} \times \sqrt{0.0001137}) = 0.0073$$

The overall Cosine Similarity calculation results are as shown in Table 7.

Table 7. Cosine Similarity calculation

APPLICANT	SCORES
AP1	0.0073
AP2	0.055
AP3	0
AP4	0.0032
AP5	0.0403
AP6	0
AP7	0
AP8	0
AP9	0.0061
AP10	0.0216

g) Recommendation Process

```
FOR Int i < LENGTH OF (cosine_sim)
   IF cosine_sim[i] > 0
      APPEND cosine_sim[i] TO indices_with_scores
   ENDIF
ENDFOR
SORT indices_with_scores FROM HIGHEST TO LOWEST
```

In the recommendation process, the system sorts the Cosine Similarity results from highest to lowest. Results with a value of 0 are excluded from the list of recommended applicants. The higher the computed similarity score, the greater the likelihood that the applicant meets the users' criteria, placing them at the top of the recommendation list.

Based on the previous Cosine Similarity calculations, AP2 achieved the highest similarity score of 0.055, followed by AP5 with 0.0403, AP10 with 0.0216, AP1 with 0.0073, AP9 with 0.0061, and AP4 with 0.0032. Meanwhile, applicants 3, 6, 7, and 8 received a score of 0. Therefore, the priority order is AP2, AP5, AP10, AP1, AP9, and AP4.

4.2 Computational Efficiency and Real-time Scalability

The development of an applicant recommendation system using the Content-Based Filtering method allows users to search for candidates based on various parameters, such as skills, work experience, or education. This feature is designed to assist recruitment teams in the selection process, which currently still relies on the search function in Microsoft Excel. This method is considered inefficient as it takes a long time, especially when there are many applicants and multiple keywords to compare. With this search feature, the candidate selection process can be carried out more quickly and efficiently, as shown in Table 8.

Table 8. Compares the manual selection process in Microsoft Excel with the system's automated process, showing efficiency and time differences

SELECTION PROCESS	**KEYWORD AMOUNT**	
	1 KEYWORD	4 KEYWORDS
Using Recommendation System	±700ms	±750ms
Using Excel	±30,000ms	±120,000ms

5 Conclusion

By implementing the Content-Based Filtering method in the applicant recommendation system, computation time becomes significantly faster compared to administrative selection using the search feature in Microsoft Excel. Test results show that the system requires only about 0.7 second to find candidates based on four keywords: *software*, *engineer*, *programming*, and *analyst*. Meanwhile, using the search feature in Microsoft Excel, the administrative selection process takes at least 30 seconds for a single keyword or more than 2 minutes to complete a search based on four keywords, depending on the number of applicants being screened.

Based on the evaluation of user satisfaction from a total of 7 workers using the DeLone and McLean model, the system achieved a satisfaction score of 95.4%. However, the program still struggles to process numerical search inputs, making it difficult to search for data like age, years of experience, and other similar aspects. As a suggestion for future research, combining Content-Based Filtering with other algorithms or methods could be explored. The choice of combination should align with the system's objectives and requirements. By implementing such methods, it is expected that the accuracy and relevance of the recommendations can improve, allowing the system to provide optimal and user-specific suggestions.

Acknowledgment

This research was carried out with the support of Universitas Multimedia Nusantara.

References

1. Olanrewaju Kazeem Ogunsola, Kareem Abidemi Arikewuyo, Odunayo Oluwarotimi Akintokunbo, Vera Ese Okwegbe: Employee Selection Process: An Approach for Effective Organizational Performance. International Journal of Social Science And Human Research. 6, 6132–6140 (2023).
2. Piotr Horodyski: Applicants' perception of artificial intelligence in the recruitment process. Computers in Human Behavior Reports. 11, (2023).
3. Janusz K. Grabara, Sebastian Kot, Lukasz Pigon: Recruitment Process Optimization: Chosen Findings From Practice In Poland. Journal of International Studies. 9, 217–228 (2016).
4. Kresnawidiansyah Agustian, Aryanda Pohan, Agustian Zen, Wiwin, Aulia Januar Malik: Human Resource Management Strategies in Achieving Competitive Advantage in Business Administration. Journal of Contemporary Administration and Management (ADMAN). 1, 108–117 (2023).
5. Atheer Abdulaziz Alsaif, Mehmet Sabih Aksoy: AI-HRM: Artificial Intelligence in Human Resource Management: A Literature Review. Journal of Computing and Communication. 2, 1–7 (2023).
6. Ramesh Nyathani: AI in Performance Management: Redefining Performance Appraisals in the Digital Age. Journal of Artificial Intelligence & Cloud Computing. 2, 134–139 (2023).

7. Pasquale Lops, Marco de Gemmis, Giovanni Semeraro: Content-based Recommender Systems: State of the Art and Trends. Springer (2011).
8. Peretz Shoval, Veronica Maidel, Bracha Shapira: An Ontology - Content-Based Filtering Method. Information Theories & Applications. 15, (2008).
9. Armadhani Hiro Juni Permana, Agung Toto Wibowo: Movie Recommendation System Based on Synopsis Using Content-Based Filtering with TF-IDF and Cosine Similarity. International Journal on Information and Communication Technology (IJoICT). 9, 1–14 (2023).
10. Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, Rasha Kashef: Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities. Applied Sciences. (2020).
11. Umair Javed, Kamran Shaukat: A Review of Content-Based and Context-Based Recommendation Systems. International Journal of Emerging Technologies in Learning (iJET). 16, (2021).
12. I'tishom Al Khoiry, Dhea Rizky Amelia: Exploring Simple Addictive Weighting (SAW) for Decision-Making. Jurnal INOVTEK POLBENG. 8, (2023).
13. Rika Idmayanti, Dwiny Meidelfi, Indri Rahmayuni, Fanni Sukma, Ramadhani: The Implementation of the Simple Multi Attribute Rating Technique Method for Evaluating the Guidance Process for the Final Project of the Applied Software Engineering Technology Students. International Journal of Advanced Science Computing and Engineering. 3, 153–160 (2021).
14. Pajri Aprilio, SY. Yuliani: Implementation of Internship Decision Support System Using Simple Multi Attribute Rating Technique (SMART). International Conference on Informatics and Computing (ICIC). (2022).
15. Nandang Hermanto, Irma Darmayanti, Dimas Saputra, Aden Hidayatuloh: Development of Mobile Application by Applying Content-Based Filtering. Jurnal dan Penelitian Teknik Informatika. 9, (2025).
16. Jie Chen, Cai Chen, Yi Liang: Optimized TF-IDF Algorithm with the Adaptive Weight of Position of Word. In: Advances in Intelligent Systems Research. Atlantis Press (2016).
17. Robin Burke, Alexander Felfernig, Mehmet H. Goker: Recommender Systems: An Overview. In: Association for the Advancement of Artificial Intelligence. AI MAGAZINE (2011).
18. Poonam B. Thorat, R. M. Goudar, Sunita Barve: Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. Int J Comput Appl. 110, (2015).
19. Dhruval Patel, Foram Patel, Uttam Chauhan: Recommendation Systems: Types, Applications, and Challenges. International Journal of Computing and Digital Systems. 1–18 (2023).
20. Yehuda Koren, Robert Bell: Advances in Collaborative Filtering. Springer US (2021).
21. Robin van Meteren, Maarten van Someren: Using Content-Based Filtering for Recommendation. NetlinQ Group. 1–10.

22. Abdul Aziz, Mubashra Fayyaz: Comparison of Content Based and Collaborative Filtering in Recommendation Systems. In: International Conference on Multimedia Information Technology and Applications. , Vietnam (2021).
23. Hongli Yuan, Alexander A. Hernandez: User Cold Start Problem in Recommendation Systems: A Systematic Review. J Phys Conf Ser. (2023).
24. Iateilang Ryngksai, L. Chameikho: Recommender Systems: Types of Filtering Techniques. International Journal of Engineering Research & Technology (IJERT). 3, (2014).
25. Lionel Ngoupeyou Tondji: Web Recommender System for Job Seeking and Recruiting, (2018).
26. Eriya, P G Kodu, F Nugrahani, A Ghosh: Recommendation system using hybrid collaborative filtering methods for community searching. J Phys Conf Ser. 1193, (2019).
27. Jesus Bobadilla, Fernando Ortega, Antonio Hernando, Jesus Bernal: A Collaborative Filtering Approach to Mitigate The New User Cold Start Problem. Knowl Based Syst. 26, 225–238 (2012).
28. Winda Yulita, Meida Cahyo Untoro, Mugi Praseptiawan, Ilham Firman Ashari, Aidil Afriansyah, Ahmad Naim Bin Che Pee: Automatic Scoring Using Term Frequency Inverse Document Frequency Document Frequency and Cosine Similarity. Scientific Journal of Informatics. 10, (2023).
29. Hans Christian, Mikhael Pramodana Agus, Derwin Suhartono: Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF). ComTech Computer Mathematics and Engineering Applications. 7, 285–294 (2016).
30. Shahzad Qaiser, Ramsha Ali: Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. Int J Comput Appl. 181, 25–29 (2018).
31. Ylber Januzaj, Artan Luma: Cosine Similarity - A Computing Approach to Match Similarity Between Higher Education Programs and Job Market Demands Based on Maximum Number of Common Words. International Journal of Emerging Technologies in Learning (iJET). 17, 258–268 (2022).
32. Daffa Rizki Surya Pratama, Tb Ai Munandar, Khairunnisa Fadhilla Ramdhania: Multinomial Naive Bayes Algorithm for Indonesian language Sentiment Classification Related to Jakarta International Stadium (JIS). International Journal of Information Technology and Computer Science Applications (IJITCSA). 2, 1–11 (2024).