BM25 Implementation For Information Retrieval: Candidate Shortlister For Recruitment Process

Sunny Dhokane
Department of Electronics and Telecommunication
Pune Institute of Computer Technology
Pune, India
sunnydhokane777@gmail.com

Chinmay Deshmukh
Department of Electronics and Telecommunication
Pune Institute of Computer Technology
Pune, India
chinmay.deshmukh21@gmail.com

Akshit Bollabattin
Department of Electronics and Telecommunication
Pune Institute of Computer Technology
Pune, India
akshit.bollabattin18@gmail.com

Siddhesh Karande
Department of Computer Engineeirng
Vishwakarama Institute of Technology
Pune, India
siddheshkarande1017@gmail.com

Bhakti Karangale
Department of Computer Engineeirng
Vishwakarama Institute of Technology
Pune, India
bhaktikarangale0201@gmail.com

Pradip S. Varade
Department of Electronics and Telecommunication
Pune Institute of Computer Technology
Pune, India
psvarade@pict.edu

2024 Intelligent Systems and Machine Learning Conference (ISML) | 979-8-3503-4387-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISML60050.2024.11007378

Abstract— In today's digital age, efficient information retrieval systems are crucial for various applications, including candi date shortlisting in recruitment processes. This research paper presents the implementation of Best Match 25 (BM25), a robust ranking algorithm, for information retrieval in the context of candidate shortlisting. The approach aims to improve the accuracy and effectiveness of candidate shortlisting compared to traditional methods. This paper proposes a method to integrate BM25 into the recruitment process, facilitating the selection of relevant candidates from a corpus of resumes or candidate profiles. This study introduces a candidate shortlisting system empowered by code-driven methodologies for information retrieval, targeting assistance to recruiters and Human Resources (HR) professionals in pinpointing candidates possessing specific skill sets. Through the integration of custom parsers and query processors, this code driven solution efficiently aligns job requirements with candidate qualifications. The primary functionalities encompass query and candidate corpus parsing, execution of matching algorithms, and result ranking. Illustrated by the provided code snippet, the system's inner workings are elucidated, showcasing its capability. By integrating indexing, ranking algorithms, and relevance scoring, the system furnishes a curated selection of top candidate matches, simplifying the shortlisting process. Furthermore, this approach demonstrates scalability and adaptability to handle extensive datasets, rendering it applicable across diverse recruitment scenarios. This study underscores the pivotal role of code driven solutions in modern candidate shortlisting, underscoring the imperative for efficient information retrieval techniques to address the evolving demands of talent acquisition.

Keywords— Candidate Shortlisting, Recruitment Process, BM25, Information Retrieval, Document Ranking, Query Processing, Code-Driven Solutions, Resume Parsing.

I. Introduction

In today's data-centric milieu, the imperative to efficiently identify suitable job candidates has become paramount for organizations spanning diverse industries. The relentless influx of resumes, profiles, and applicant data necessitates a precise and streamlined approach to candidate shortlisting. Traditionally, this task has been labor-intensive, prone to biases inherent in manual screening methods. Consequently, there is a burgeoning demand for automated solutions capable of efficiently filtering and ranking candidates based on their relevance to the job description. To address this challenge, this study delves into candidate shortlisting by leveraging Information Retrieval (IR) techniques[4]. Information Retrieval, rooted in the extraction of valuable insights from extensive datasets, offers a potent remedy to the recruitment conundrum. By harnessing advanced algorithms and data processing methods, this project endeavors to empower recruiters and HR professionals to swiftly and accurately discern the most promising candidates. This endeavor epitomizes the fusion of contemporary technology with recruitment imperatives, seeking to enhance the precision and efficiency of candidate shortlisting through the prism of Information Retrieval. This paper introduces BM25, a probabilistic information retrieval algorithm renowned for its effectiveness in various applications, including web search and document ranking. BM25's comprehensive consideration of both term frequency and document length offers a balanced approach to ranking documents, making it an ideal candidate for enhancing the efficiency and accuracy of the recruitment process[1]. By implementing BM25 for candidate shortlisting the aim is to revolutionize the traditional methods, paving the way for a more streamlined and effective approach to talent acquisition.

II. Proposed Methods

A. BM25

1) Data Preprocessing: Initially, the corpus of resumes or candidate profiles is preprocessed to extract pertinent information, including skills, experience, and education. This preprocessing step ensures that the candidate data is structured and ready for analysis[11],[12].

2) Query Processing: Simultaneously, the job description or search query is processed to identify essential terms and criteria for candidate evaluation. This involves tokenization and normalization techniques to extract key keywords and phrases[9].

3) BM25 Implementation: The core of our methodology lies in the implementation of the BM25 algorithm for calculating the relevance score of each candidate profile to the given job description or query. BM25 incorporates factors such as term frequency, document length, and inverse

979-8-3503-4387-8/24/$31.00 ©2024 IEEE

document frequency to compute the relevance score accurately[7].

4) Relevance Scoring: Based on the BM25 scores computed for each candidate profile, relevance scores are assigned to indicate the degree of match between the candidate and the job requirements. These scores provide a quantitative measure of candidate suitability[8].

5) Candidate Ranking: Utilizing the BM25 relevance scores, candidate profiles are ranked in descending order of relevance. This ranked list serves as a shortlist of potential candidates for further evaluation by recruiters or HR professionals[10].

The proposed methodology for candidate shortlisting revolves around the implementation of the BM25 algorithm, a renowned ranking function in information retrieval[2]. The process involves the following steps, as depicted in Fig. 1.

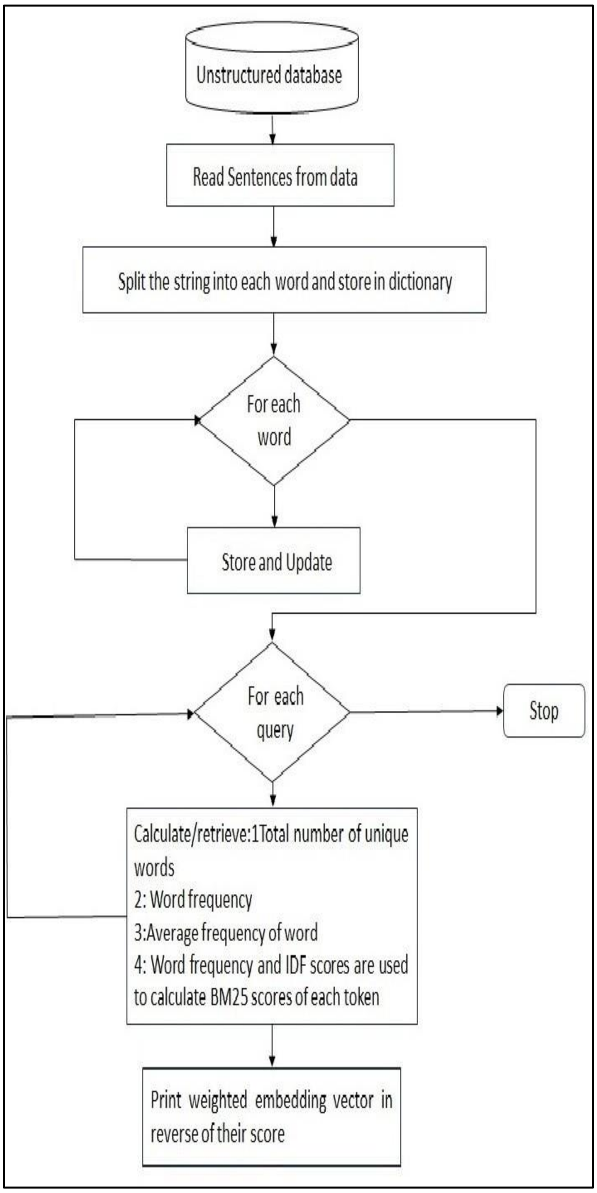

Unstructured database
Read Sentences from data
Split the string into each word and store in dictionary
For each word
Store and Update
For each query
Stop
Calculate/retrieve:1Total number of unique words
2: Word frequency
3:Average frequency of word
4: Word frequency and IDF scores are used to calculate BM25 scores of each token
Print weighted embedding vector in reverse of their score

Fig. 1. Proposed flowchart for implementing BM25.

By exclusively leveraging the BM25 algorithm, the methodology ensures a focused and efficient approach to candidate shortlisting. The BM25 algorithm's effectiveness in document ranking, coupled with its ability to handle various factors influencing relevance, makes it an ideal choice for enhancing the recruitment process's efficiency and accuracy. Through this methodology, the aim is to streamline the candidate shortlisting process and facilitate the identification of the most suitable candidates for organizational needs. BM25 (Best Matching 25) is a robust ranking function widely utilized by search engines to assess the relevance of documents to specific search queries within the field of information retrieval. It extends the scoring function of the binary independence model, incorporating both document and query term weights. The BM25 formula is represented in Equation. 1.

$$\text{BM25 Score: } \log \frac{r+0.5}{R-r+0.5} \cdot \frac{(k1+1)\,.f}{K+f} \cdot \frac{(k2+1)\,.qf}{k2+qf} \qquad (1)$$

Where:

- r is the number of relevant documents containing theterm.
- R is the total number of relevant documents.
- f is the term frequency within the document.
- qf is the term frequency within the query.
- N is the total number of documents.
- dl is the length of the document.
- $avdl$ is the average document length.

Wherein; K is the normalization factor computed as,

$$\text{K} = (\text{k1} * (1\text{-b})) + \text{b} * (\frac{dl}{avdl})\ ) \qquad (2)$$

Here, k1, k2, and b are constants typically set empirically to control the term frequency saturation characteristics and document length normalization. The BM25 formula integrates these parameters to compute a relevance score, guiding the ranking of documents based on their relevance to a given query[1].

B. TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a key metric in information retrieval and machine learning. It consists of two main parts: Term Frequency (TF) and Inverse Document Frequency (IDF). TF measures how often a term appears in a document, while IDF evaluates how rare a term is across the entire corpus. The TF-IDF vectorization process assigns a score to each term-document pair, allowing for similarity comparisons between documents.[5],[6].Text preprocessing steps such as lowercasing, tokenization, stop words removal, and stemming are commonly applied before implementing TF-IDF. A vocabulary-building process indexes each term in the corpus, and TF-IDF scores are calculated using a TF-IDF Transformer after building the vocabulary and generating a sparse matrix. It's noteworthy that TF-IDF values for some terms may be 0 if they do not appear in the training corpus[3].

III. Comparison of BM25 and TF-IDF

While both algorithms are commonly used for information retrieval, they differ in their underlying principles and performance characteristics, as illustrated in Table 1.

BM25 extends TF-IDF by incorporating additional factors such as document length normalization and term saturation. It is designed to address some of the limitations of TF-IDF, such as sensitivity to document length. BM25 tends to perform well in scenarios where document lengths vary significantly. TF-IDF calculates the importance of a term in a document relative to a corpus based on its frequency and rarity. However, TF-IDF does not consider factors such as document length normalization, which can affect the relevance score, especially in long documents. TF-IDF may be more suitable for applications where document lengths are relatively uniform.

TABLE. 1: Comparison of BM25 and TF-IDF

Criteria	BM25	TF-IDF
Full Name	Okapi BM25 (Best Matching 25)	Term Frequency – Inverse document frequency
Mathematical Formula	Complex formula with term and document statistics	Simple multiplication of the term frequency and inverse document frequency
Team Weighting	Adapts to query term frequency and document length	Considers only term frequency and document frequency
Relevance Score Interpretation	Higher scores represent more relevance	Higher scores represent higher term importance

IV. Results and Discussions

A. Corpus Dataset:

1) Document 1:

Keywords: Java, C++, Python, MySql, Javascript, MongoDB, flask, NLP

Content: This document, as shown in Figure 2, likely contains information related to programming languages such as Java, C++, Python, and databases like MySQL. Additionally, it may discuss web technologies like JavaScript and MongoDB, along with topics like Flask and Natural Language Processing (NLP). The mention of "stock market" suggests a connection to financial data or applications.

2) Document 2:

Keywords: singer, dancer, event planner, leadership, entrepreneur

Content: This document, as shown in Figure 2, seems to be related to individuals with skills and roles in the entertainment industry. It may discuss singers, dancers, event planners, and leadership in the context of entrepreneurship.

3) Document 3:

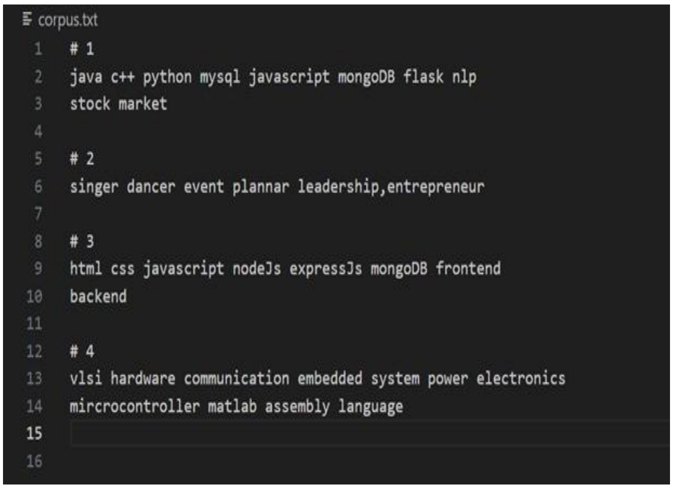
Keywords: HTML, CSS, Javascript, NodeJs, ExpressJs, MongoDB, frontend, backend

Content: This document, as shown in Figure 2, likely contains information related to web development technologies. It includes HTML, CSS, and JavaScript, as well as server-side technologies like Node.js and Express.js. The presence of MongoDB suggests a focus on databases. "Frontend" and "backend" indicate discussions about web development roles and technologies.

4) Document 4:

Keywords: VLSI, hardware, communication, embedded system, power electronics, microcontroller, Matlab, assembly language

Content: This document, as shown in Figure 2, is likely related to topics in electronics and embedded systems. It includes keywords like VLSI (Very Large-Scale Integration), hardware, communication, power electronics, microcontrollers, MATLAB (a programming environment), and assembly language. These keywords suggest a technical document on electronic systems and components.



```
# 1
java c++ python mysql javascript mongoDB flask nlp
stock market

# 2
singer dancer event plannar leadership,entrepreneur

# 3
html css javascript nodeJs expressJs mongoDB frontend
backend

# 4
vlsi hardware communication embedded system power electronics
mircrocontroller matlab assembly language


```

Fig. 2. Corpus Dataset.

B. Given Query:

1) Query 1 (as shown in Figure 3):

Keywords: java, C++, Python, Mysql, Javascript

This query is seeking information related to programming languages and web technologies, specifically Java, C++, Python, MySQL, and JavaScript. It appears to be interested in topics related to software development and web programming.

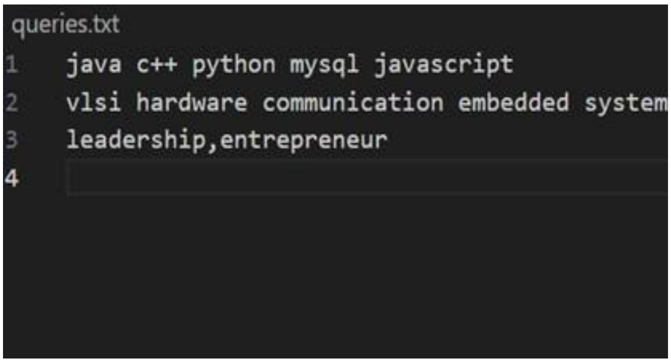
2) Query 2 (as shown in Figure 3):

Keywords: VLSI hardware communication embedded system\\

This query is looking for information about VLSI (Very Large-Scale Integration), hardware components, communication technologies, and embedded systems, which are all related to the field of electronics and embedded systems.

3) Query 3 (as shown in Figure 3):

Keywords: leadership, entrepreneur
This query is focused on leadership and entrepreneurship, indicating an interest in topics related to leadership qualities and entrepreneurial activities.

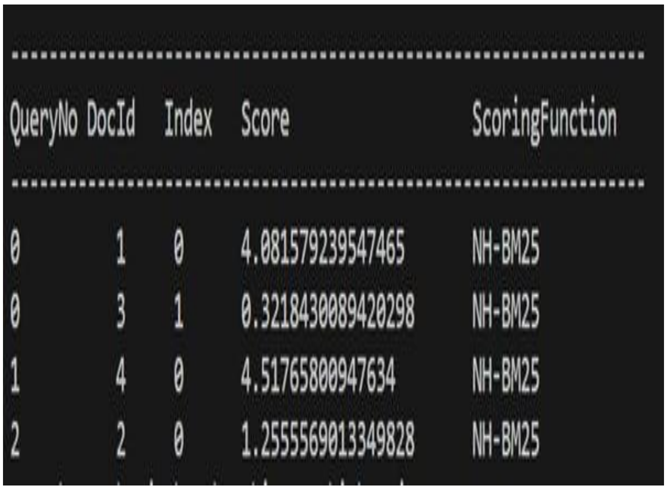


```
queries.txt
1   java c++ python mysql javascript
2   vlsi hardware communication embedded system
3   leadership,entrepreneur
4
```

Fig. 3 Given Query

C. Output on Console

In Figure. 4, the presented output showcases the key components of an information retrieval system's results. The "QueryNo" column represents the query number, beginning at 0 and incrementing for each query. "DocId" serves as a reference to the document in the corpus, while the "Index" column indicates the rank or position of the document in the result list, with 0 signifying the highest rank. The "Score" reveals the relevance score calculated using the NH-BM25 scoring function, commonly employed in information retrieval. This output clarifies how the system ranked and scored documents in response to three queries. For instance, in "Query 0," the top result holds a high score of around 4.08, signifying strong relevance, while the second result lags behind with a lower score of approximately 0.32. Similarly, "Query 1" yields a highly relevant top result with a score of about 4.52, while "Query 2" produces a top result with a moderate relevance score of roughly 1.26.



```
=================================================================
QueryNo DocId  Index  Score               ScoringFunction
=================================================================
0         1    0      4.081579239547465   NH-BM25
0         3    1      0.3218430089420298  NH-BM25
1         4    0      4.51765800947634    NH-BM25
2         2    0      1.2555569013349828  NH-BM25
```

Fig. 4. BM25 score results

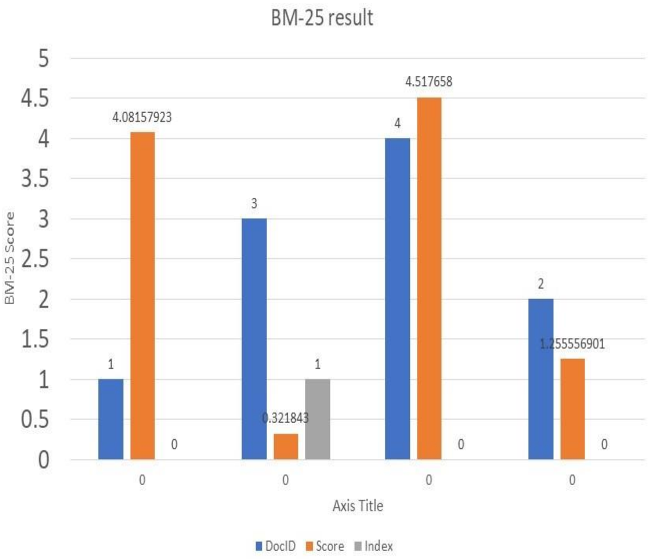


Fig. 5 Bar Chart of BM25 Scores

D. Integration with Frontend and Backend

In the integration with frontend and backend systems, the aim was to provide a seamless and intuitive platform for both recruiters and candidates. The system allows candidates to upload their resumes, which are then processed by the backend engine using the BM25 algorithm to calculate relevance scores based on the job description requirement.

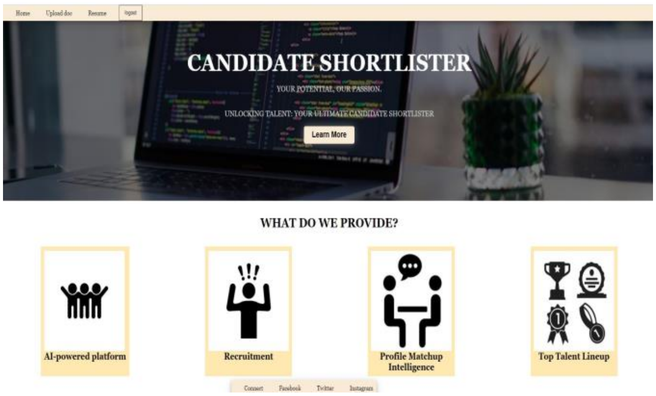
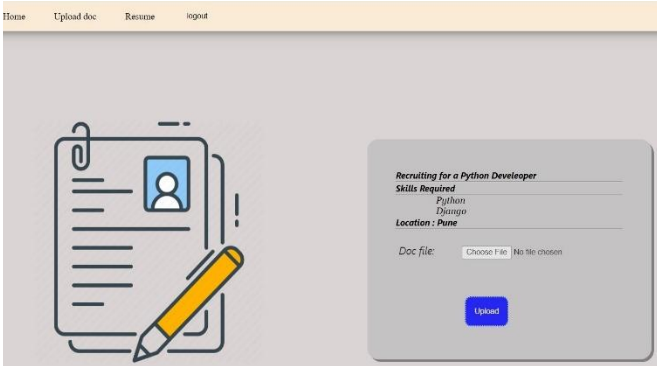


Fig. 6 Frontend Interface

1) **Frontend Interface** The frontend interface offers a user-friendly platform for candidates to upload their resumes and for recruiters to specify job requirements. Here are the key components of the frontend interface:
   - Resume Upload: Candidates can easily upload their resumes using a straightforward interface. The system supports document format such as DOC.
   - Job Description Input: Recruiters can input job requirements, including desired skills, experience levels, and qualifications, using a simple form.

- Submit Button: Once the resume and job descriptionare provided, a submit button triggers the backend processing to calculate the relevance score.



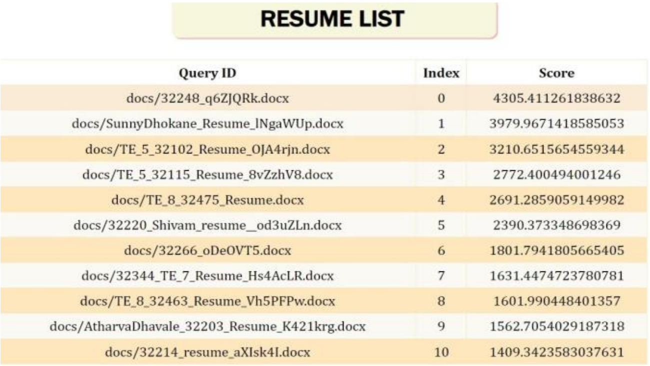
Recruiting for a Python Develeoper
Skills Required
Python
Django
Location : Pune

Doc file:

Fig. 7 Frontend Interface

2) Backend Processing and Scoring

- Resume Parsing: The backend system parses the uploaded resumes to extract relevant information such as skills, experience, and education.
- Job Description Analysis: Similarly, the job description provided by the recruiter is analysed to identify key requirements and criteria.
- BM25 Scoring: The BM25 algorithm is applied to calculate relevance scores for each resume based on its alignment with the job description. The scoring considers factors such as term frequency, document length, and inverse document frequency.



RESUME LIST

Query ID	Index	Score
docs/32248_q6ZJQRk.docx	0	4305.411261838632
docs/SunnyDhokane_Resume_lNgaWUp.docx	1	3979.9671418585053
docs/TE_5_32102_Resume_OJA4rjn.docx	2	3210.6515654559344
docs/TE_5_32115_Resume_8vZzhV8.docx	3	2772.400494001246
docs/TE_8_32475_Resume.docx	4	2691.2859059149982
docs/32220_Shivam_resume__od3uZLn.docx	5	2390.373348698369
docs/32266_oDeOVT5.docx	6	1801.7941805665405
docs/32344_TE_7_Resume_Hs4AcLR.docx	7	1631.4474723780781
docs/TE_8_32463_Resume_Vh5PFPw.docx	8	1601.990448401357
docs/AtharvaDhavale_32203_Resume_K421krg.docx	9	1562.7054029187318
docs/32214_resume_aXIsk4I.docx	10	1409.3423583037631

Fig. 8. Top Candidates Shortlisted

V. Conclusions

This research paper introduces a pioneering method for candidate shortlisting within the recruitment process by leveraging the BM25 algorithm for information retrieval. Integration of BM25 into the recruitment workflow aims to optimize candidate selection procedures and elevate the quality of hires. Through empirical validation, the study demonstrates the superior efficacy of BM25 over traditional methodologies like TF-IDF, signaling its potential to reshape the landscape of recruitment and talent acquisition. Findings underscore the transformative impact of BM25 in revolutionizing candidate shortlisting practices, offering recruiters and HR professionals a powerful tool to identify top candidates swiftly and accurately. Moreover, the comparative analysis between BM25 and TF-IDF reveals nuanced performance disparities, shedding light on the unique strengths of each approach within the context of candidate shortlisting.

Future research endeavors hold the potential to explore additional enhancements and optimizations to further refine the proposed methodology. By delving deeper into the intricacies of information retrieval techniques and recruitment dynamics, new avenues for innovation can be unlocked, leading to the development of more advanced and intelligent recruitment systems. In essence, this research delineates a robust framework for candidate shortlisting that harnesses the capabilities of code-driven methodologies for information retrieval. By amalgamating cutting-edge algorithms and data processing techniques, the proposed system not only simplifies the candidate shortlisting process but also underscores the critical importance of precision and efficiency in modern recruitment practices. Ultimately, this study contributes valuable insights into the evolving landscape of talent acquisition, emphasizing the imperative for continuous advancement and adaptation in recruitment strategies.

ACKNOWLEDGEMENTS

Extended heartfelt gratitude to all who have supported in completing this research. Encouragement, guidance, and assistance have been invaluable throughout this journey. Thank you for the unwavering support and belief in this work

REFERENCES

1. Robertson, Stephen & Zaragoza, Hugo. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval. 3. 333-389. 10.1561/1500000019.
2. Svore, Krysta & Burges, Christopher. (2009). A machine learning approach for improved BM25 retrieval. International Conference on Information and Knowledge Management, Proceedings. 1811-1814. 10.1145/1645953.1646237.
3. Kim, SW., Gil, JM. Research paper classification systems based on TF-IDF and LDA schemes. Hum. Cent. Comput. Inf. Sci. **9**, 30 (2019). https://doi.org/10.1186/s13673-019-0192-7
4. A.Barducci, S.Iannaccone, V.La, V.Moscato, G. Sperl'ı, and S. Zavota, "An end-to-end framework for information extraction from Italian resumes," Expert Systems With Applications, vol. 210, no. October 2021, p. 118487, 2022.
5. A. I. Kadhim, "Term Weighting for Feature Extraction on Twitter: A Comparison Between BM25 and TF-IDF," 2019 International Conference on Advanced Science and Engineering (ICOASE),

Zakho - Duhok, Iraq, 2019, pp. 124-128, doi: 10.1109/ICOASE.2019.8723825.

6. X. Wang and F. Yuan, "Course Recommendation by Improving BM25 to Identity Students' Different Levels of Interests in Courses," 2009 International Conference on New Trends in Information and Service Science, Beijing, China, 2009, pp. 1372-1377, doi: 10.1109/NISS.2009.104.

7. Lixin Xu, Guang Chen and Lei Yang, "Incremental clustering in short text streams based on BM25," 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, Shenzhen, 2014, pp. 8-12, doi: 10.1109/CCIS.2014.7175694.

8. M. Murata, H. Nagano, R. Mukai, K. Kashino and S. Satoh, "BM25 With Exponential IDF for Instance Search," in IEEE Transactions on Multimedia, vol. 16, no. 6, pp. 1690-1699, Oct. 2014, doi: 10.1109/TMM.2014.2323945

9. A. R. Gilang Purnama, I. Nurma Yulita and A. Helen, "Search System for Translation of Al-Qur'an Verses in Indonesian using BM25 and Semantic Query Expansion," 2021 International Conference on Artificial Intelligence and Big Data Analytics, Bandung, Indonesia, 2021, pp. 1-7, doi: 10.1109/ICAIBDA53487.2021.9689757.

10. S. Sari and M. Adriani, "Learning to rank for determining relevant document in Indonesian-English cross language information retrieval using BM25," 2014 International Conference on Advanced Computer Science and Information System, Jakarta, Indonesia, 2014, pp. 309-314, doi: 10.1109/ICACSIS.2014.7065896.

11. L. Wang, M. Tanaka and H. Yamana, "What is your Mother Tongue?: Improving Chinese native language identification by cleaning noisy data and adopting BM25," 2016 IEEE International Conference on Big Data Analysis (ICBDA), Hangzhou, China, 2016, pp. 1-6, doi: 10.1109/ICBDA.2016.7509793.

12. E. LaBouve and L. Stanchev, "Combining Parts of Speech, Term Proximity, and Query Expansion for Document Retrieval," 2019 IEEE 13th International Conference on Semantic Computing (ICSC), Newport Beach, CA, USA, 2019, pp. 150-153, doi: 10.1109/ICOSC.2019.8665507.