Online version at https://journal.lenterailmu.com/index.php/josapen

JOSAPEN

E-ISSN: 3031-2272 (Online)

Vector Space Model-based Information Retrieval Systems at South Sumatera Regional Libraries

M. Akbar As Shiddiqi, A Sanmarino

University of Indo Global Mandiri, Palembang, Indonesia

ARTICLE INFO

Article history:
Received 15 January 2023
Revised 21 March 2023
Accepted 29 May 2023

Keywords:
Information Retrieval
Vector Space Model (VSM)
Library Collections
TF-IDF Calculation
Document Representation

ABSTRACT

This study presents an overview of the research aimed at optimizing library information retrieval through the utilization of the Vector Space Model (VSM) method in a computer science context. Libraries, as publicly financed collections, provide extensive knowledge resources, eliminating the need for individual book purchases. However, the challenge lies in efficiently navigating the expanding library collections. To tackle this issue, the study employs information retrieval techniques, particularly the VSM method, which assesses term similarity by assigning weights to terms, enabling document and query representation as vectors. The relevance between documents and queries is measured through vector similarity. This approach, integrated with indexing, streamlines collection retrieval in libraries. Employing the Waterfall model for system development, the research outlines phases like analysis, design, coding, testing, and implementation. While effective, the model's rigidity in accommodating evolving requirements poses limitations. The VSM method's numerical representation of text documents facilitates precise similarity calculations, supported by TF-IDF values indicating term importance in documents relative to the corpus. The study further extends to system design using UML diagrams and a visitor interface, integrating VSM for efficient search functionality. Black-box testing confirms the robustness of the system components and interfaces. Overall, this research presents a systematic approach to enhance information retrieval in libraries, emphasizing the VSM's pivotal role in optimizing document searches within expansive collections.

1. Introduction

A library is a collection of books and magazines. Although it can be interpreted as an individual's private collection, a library is more generally known as a large collection that is financed and operated by a city or institution and is used by the community. With the existence of libraries, to increase people's knowledge there is no need to buy many books at their own expense. As time goes by, the number of book and magazine collections in libraries

* *Corresponding author:* A Sanmarino
E-mail address: sanmorino@uigm.ac.id

increases. With such a large collection, it creates a new problem, namely the difficulty of searching for book or magazine collections. So searching for a collection of books or magazines takes quite a long time.

Based on these problems, the author tries to provide a solution by utilizing one of the fields of computer science, namely information retrieval. Information retrieval is related to the representation of storage, structure, and access of documents which aims to facilitate the search for information. Therefore, this system is very influential for library visitors in searching for documents or information that is relevant to what they want. To get relevant documents based on a query, the author uses the Vector Space Model (VSM) method [1], [2]. VSM is a method for determining the level of closeness or similarity of terms by weighting the terms. This Vector Space Model method represents a document and query in a vector. The relevance of a document to a query is based on the similarity between the document vector and the query vector. One way is to enter one or more terms. These terms will later be matched with a data representation called an index. Indexes are the data structures most widely used by information retrieval systems. Indexes include parts of the types of library materials. The purpose of the search mechanism is as a search tool (collection retrieval) in the library. So with this guide (index), users can search for collections quickly and precisely in the library. Furthermore, this index is used to search for a document using the concept of information retrieval.

Looking at the use of indexes, the author tries to use the concept of information retrieval which is implemented in a web-based text document storage system. By applying the concept of information retrieval, it is hoped that the system can search for documents more quickly and accurately.

2. Method

Figure 1 shows the research method that the author used to achieve the objectives set at the beginning. The system development stage used is the waterfall model. The waterfall model is an SDLC method that has the characteristic that each result in Waterfall must be completed first before proceeding to the next phase. The Waterfall Model is a software development methodology that follows a linear and structured flow [3], [4], [5]. It consists of a series of phases that must be completed sequentially, and each phase is dependent on the completion of the previous phase. Following are some of the main phases in the Waterfall model:

a. Analysis: The stage where system requirements are gathered and thoroughly understood. It involves interaction with users and stakeholders to define functional and non-functional requirements.
b. Design: After the requirements are collected, the next step is to design the system architecture. This includes designing the system structure, identifying algorithms, and preparing the necessary technical specifications.
c. Coding: This stage involves coding the software according to the specifications created at the design stage. The development team creates code based on the approved design.
d. Testing: After implementation, the system is tested to ensure that all requirements have been met and that there are no significant bugs or errors. These tests include functional, performance, and security tests.
e. Delivery/Implementation: Once the system passes all the tests, it is ready to be implemented and released into a production environment or used by end users.

One of the main disadvantages of the Waterfall model is its inability to handle the frequent changes in requirements that occur in the software development cycle. Due to the linear nature of this model, it is difficult to return to a previous phase once a particular phase has been completed.

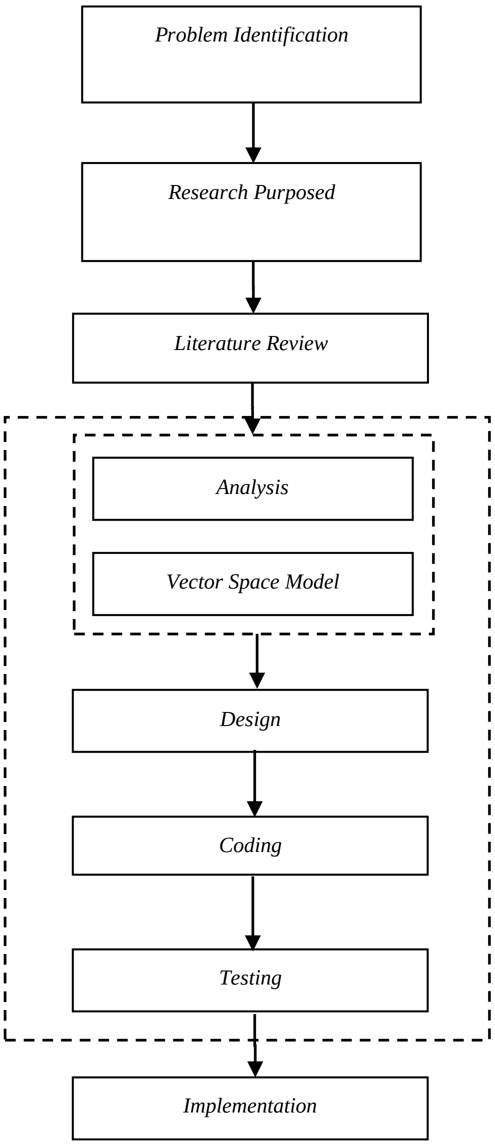

Figure 1 – Research design

The Vector Space Model (VSM) is a mathematical framework used in information retrieval and natural language processing to represent text documents as numerical vectors in a high-dimensional space. It's a way to quantify and compare the similarity between documents based on the occurrences of words or terms within them.

How the Vector Space Model Works:

1. Document-Term Matrix:
   - Each document in a corpus is represented as a vector.
   - The entire corpus forms a matrix where each row represents a document, and each column represents a term in the vocabulary.
2. Term Frequency (TF):
   - TF measures the frequency of a term in a document.
   - Higher TF indicates the importance of a term within a document.
3. Inverse Document Frequency (IDF):
   - IDF measures the importance of a term in the entire corpus.
   - Terms that appear in many documents have lower IDF values, while terms appearing in fewer documents have higher IDF values.
4. TF-IDF Weighting:
   - TF-IDF combines TF and IDF to calculate the weight of a term in a document.
   - High weight is assigned to terms that are frequent in a document but rare in the entire corpus.

For example, consider a small corpus with three documents:

1. Document 1: "Machine learning is fascinating."
2. Document 2: "Learning about machine learning is important."
3. Document 3: "Natural language processing is a part of machine learning."

Steps to Construct Vector Space Model:

1. Tokenization and Vocabulary Creation:
   - Tokenize the documents into terms: "machine", "learning", "fascinating", "important", "natural", "language", "processing", "part".
   - Form a vocabulary: ["machine", "learning", "fascinating", "important", "natural", "language", "processing", "part"].
2. TF-IDF Calculation (Table 1):

Table 1 – TF-IDF Calculation

Term	Document 1	Document 2	Document 3
machine	1	1	1
learning	1	2	1
fascinating	1	0	0
important	0	1	0
natural	0	0	1
language	0	0	1
processing	0	0	1
part	0	0	1

3. Vector Representation:
   - Document 1: [1, 1, 1, 0, 0, 0, 0, 0]
   - Document 2: [1, 2, 0, 1, 0, 0, 0, 0]
   - Document 3: [1, 1, 0, 0, 1, 1, 1, 1]

The Vector Space Model provides a structured way to represent text documents numerically, allowing for efficient information retrieval, document similarity calculations, and other text processing tasks.

3. Result and Discussion

In this section, we will continue the discussion of TF-IDF Calculation (Table 1) with a detailed explanation of the equations. Inverse Document Frequency (IDF) measures the importance of a term in the entire corpus. Calculate IDF for each term in the corpus using Equation (1):

$$IDF(term) = log(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ the\ term}) \qquad (1)$$

$$IDF(machine) = log(\frac{3}{3}) = 0\ (since\ all\ documents\ contain\ "machine")$$

$$IDF(learning) = log(\frac{3}{3}) = 0\ (since\ all\ documents\ contain\ "learning")$$

$$IDF(fascinating) = log(\frac{3}{1}) = log(3)$$

$$IDF(important) = log(\frac{3}{2}) = log(3/2)$$

$$IDF(natural) = log(\frac{3}{1}) = log(3)$$

$$IDF(language) = log(\frac{3}{1}) = log(3)$$

$$IDF(processing) = log(\frac{3}{1}) = log(3)$$

$$IDF(part) = log(\frac{3}{1}) = log(3)$$

TF-IDF is the product of TF and IDF for each term in each document. Calculate TF-IDF for each term in each document using Equation (2):

$$TF - IDF(term,\ document) = TF(term,\ document) \times IDF(term) \qquad (2)$$

For example, TF-IDF(machine, Document 1) = 1 * 0 = 0. The TF-IDF values form the vector representation of each document in the vector space.

- Document 1: [0, 0, log(3), 0, 0, 0, 0, 0]
- Document 2: [0, 0, 0, log(2/3), 0, 0, 0, 0]
- Document 3: [0, 0, 0, 0, log(3), log(3), log(3), log(3)]

The values in these vectors are the TF-IDF weights, reflecting the importance of terms in the documents relative to the entire corpus. This model enables comparisons and similarity calculations between documents based on their vector representations. After done with the Vector Space Model, the author started designing the proposed system by creating a UML diagram [6], [7]. The proposed system design includes Use case diagrams, Activity diagrams, Sequence diagrams, and Class diagrams.

3.1. Use diagram

Several things need to be described, namely actors and use cases. Actors are users who are connected to the system and can be people (indicated by their role). The actor is symbolized by the figure of a stick man with a noun at the bottom that states the role/system. Use cases are depicted with an ellipse symbol with the name of the active verb inside which states the activity from the actor's perspective [8], [9].

3.2. Activity diagram

An activity diagram is a description of function paths in an information system [10]. In full, the activity diagram defines where the system process starts, where it stops, what activities occur during the system process, and what sequence these activities occur in.

3.3. Sequence diagram

Based on the use case that has been created, a sequence diagram is obtained which describes the behavior of objects in the use case by describing the lifetime of the object and the messages sent and received between objects.

3.4. Class diagram

Class diagrams describe the types of objects in the system and the various static relationships that exist between them [11]. Class diagrams show the properties and operations of a class and the boundaries contained in the object relationships.

3.5. System Interface

This visitor interface is a page for visitors in the book search process (Figure 2). Search process based on book title, search for books by directly clicking the Search button.

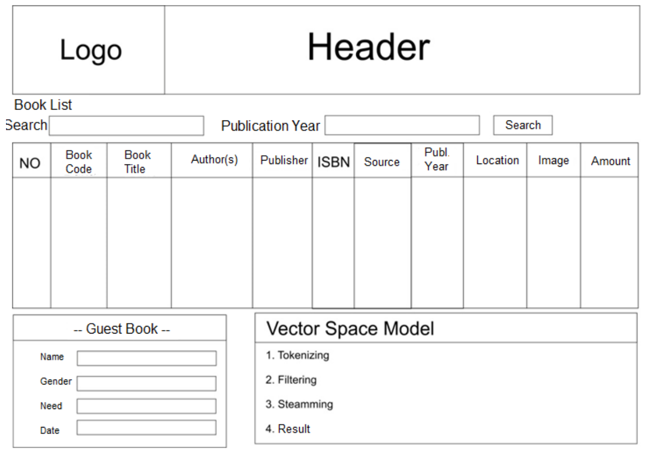

Figure 2 – Login page

Next, the author carries out black box testing as an initial stage of evaluation of the system that has been created [12]-[15]. The test results show that all functions and interfaces of the proposed system can run well.

4. Conclusion

The utilization of libraries as repositories of knowledge eliminates the necessity for individuals to amass extensive personal book collections, as these public institutions provide access to a broad range of materials financed and managed by cities or institutions. However, the accumulation of a vast array of books and magazines within libraries introduces the challenge of efficiently locating specific collections. To address this issue, the author leverages information retrieval principles, specifically employing the Vector Space Model (VSM) method, which quantifies term similarity by assigning weights to terms in a document. This technique involves representing documents and queries as vectors, and their relevance is measured based on vector similarity. To expedite and refine the search process, terms are indexed, a crucial aspect widely employed in information retrieval systems, aiding in rapid and accurate collection retrieval within the library. The author proceeds to apply information retrieval concepts within a web-based text document storage system, aiming to enhance the speed and precision of document searches. The research employs the Waterfall model for system development, a structured approach that progresses sequentially through phases like analysis, design, coding, testing, and implementation. However, the model's inflexibility in accommodating changing requirements throughout development stands as a significant drawback. The Vector Space Model, a key methodology in information retrieval, enables numerical representation of text documents, facilitating efficient document similarity calculations and information retrieval processes. The calculated TF-IDF values, representing term importance within documents relative to the entire corpus, support comparisons between documents. The subsequent phase involves system design with UML diagrams, including use case, activity, sequence, and class diagrams, delineating system functionality, user interaction, and object behavior. Additionally, a visitor interface for book search processes is created, integrating the Vector Space Model stages for efficient search functionalities. The system undergoes black-box testing, affirming the proper functionality of all system components and interfaces.

Acknowledgements

We would like to acknowledge University of Indo Global Mandiri for supporting this work.

REFERENCES

[1] D. Xu and T. Miller, “A simple neural vector space model for medical concept normalization using concept embeddings,” *J. Biomed. Inform.*, vol. 130, no. January, p. 104080, 2022, doi: 10.1016/j.jbi.2022.104080.

[2] H. Du and Y. Bin Kang, “An open-source framework for

ExpFinder integrating N-gram vector space model and μCO-HITS[Formula presented],” *Softw. Impacts*, vol. 8, no. March, p. 100069, 2021, doi: 10.1016/j.simpa.2021.100069.

[3] K. D. Prasetya, Suharjito, and D. Pratama, “Effectiveness Analysis of Distributed Scrum Model Compared to Waterfall approach in Third-Party Application Development,” *Procedia Comput. Sci.*, vol. 179, no. 2019, pp. 103–111, 2021, doi: 10.1016/j.procs.2020.12.014.

[4] T. Thesing, C. Feldmann, and M. Burchardt, “Agile versus Waterfall Project Management: Decision model for selecting the appropriate approach to a project,” *Procedia Comput. Sci.*, vol. 181, pp. 746–756, 2021, doi: 10.1016/j.procs.2021.01.227.

[5] A. A. S. Gunawan, B. Clemons, I. F. Halim, K. Anderson, and M. P. Adianti, “Development of e-butler: Introduction of robot system in hospitality with mobile application,” *Procedia Comput. Sci.*, vol. 216, no. 2019, pp. 67–76, 2022, doi: 10.1016/j.procs.2022.12.112.

[6] G. Bergström *et al.*, “Evaluating the layout quality of UML class diagrams using machine learning,” *J. Syst. Softw.*, vol. 192, p. 111413, 2022, doi: 10.1016/j.jss.2022.111413.

[7] H. Wu, “QMaxUSE: A new tool for verifying UML class diagrams and OCL invariants,” *Sci. Comput. Program.*, vol. 228, p. 102955, 2023, doi: 10.1016/j.scico.2023.102955.

[8] P. Danenas, T. Skersys, and R. Butleris, “Natural language processing-enhanced extraction of SBVR business vocabularies and business rules from UML use case diagrams,” *Data Knowl. Eng.*, vol. 128, no. February, p. 101822, 2020, doi: 10.1016/j.datak.2020.101822.

[9] Meiliana, I. Septian, R. S. Alianto, Daniel, and F. L. Gaol, “Automated Test Case Generation from UML Activity Diagram and Sequence Diagram using Depth First Search Algorithm,” *Procedia Comput. Sci.*, vol. 116, pp. 629–637, 2017, doi: 10.1016/j.procs.2017.10.029.

[10] Z. Daw and R. Cleaveland, “Comparing model checkers for timed UML activity diagrams,” *Sci. Comput. Program.*, vol. 111, no. P2, pp. 277–299, 2015, doi: 10.1016/j.scico.2015.05.008.

[11] F. Chen, L. Zhang, X. Lian, and N. Niu, “Automatically recognizing the semantic elements from UML class diagram images,” *J. Syst. Softw.*, vol. 193, p. 111431, 2022, doi: 10.1016/j.jss.2022.111431.

[12] D. Felicio, J. Simao, and N. Datia, “Rapitest: Continuous black-box testing of restful web apis,” *Procedia Comput. Sci.*, vol. 219, no. 2022, pp. 537–545, 2023, doi: 10.1016/j.procs.2023.01.322.

[13] H. Bostani and V. Moonsamy, “EvadeDroid: A Practical Evasion Attack on Machine Learning for Black-box Android Malware Detection,” *Comput. Secur.*, p. 103676, 2021, doi: 10.1016/j.cose.2023.103676.

[14] F. Pagano, A. Romdhana, D. Caputo, L. Verderame, and A. Merlo, “SEBASTiAn: A static and extensible black-box application security testing tool for iOS and Android applications,” *SoftwareX*, vol. 23, p. 101448, 2023, doi: 10.1016/j.softx.2023.101448.

[15] C. Cronley *et al.*, “Designing and evaluating a smartphone app to increase underserved communities’ data representation in transportation policy and planning,” *Transp. Res. Interdiscip. Perspect.*, vol. 18, no. January, p. 100763, 2023, doi: 10.1016/j.trip.2023.100763.