SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I5P111

Volume 12 Issue 5, 126-134, May 2025

Original Article

Automated Web Service Discovery and Computing from Public Repositories through Probabilistic Matchmaking

Gajanan V. Bhole[1], Sudhir Kadam[2], Meena Chavan[3], A Y prabhakar[4], Atul Ashok Kadam[5], Pramod Jadhav[6]

[1,2,3,4,6]*Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India*
[5]*K.E. Society's Rajarambapu Institute of Technology Rajaramnagar, Islampur, Maharashtra, India.*

[2]*Corresponding Author : sudhirkadam@bvucoep.edu.in*

Received: 07 March 2025 Revised: 10 April 2025 Accepted: 11 May 2025 Published: 27 May 2025

Abstract *- To develop an efficient and comprehensive approach to web service discovery and retrieving APIs from vast repositories based on user queries and requirements and ranking the results based on relevance. The strategy incorporates numerous techniques like semantic search, graph-based ranking, and relevance scoring. The module applies semantic expansion on user queries the moment they are received through Word Net to improve query representation. The next step is to vectorize the expanded query through TF-IDF, which facilitates semantic similarity computation with the web services available. The semantic similarity scores are then studied with the help of a graph where the edges are semantic similarity scores, and the nodes represent web services. Importance scores are then given to each web service on this graph with the help of PageRank, helping us understand the relevance of the web services. Not just this, the Okapi BM25 algorithm is also applied to compute the relevance score. The final ranking of the web service is given based on integrated scores of Okapi BM25 and PageRank. This ranked list is finally presented to the user. With the help of The module and the approach it follows, users can navigate through vast repositories full of APIs to find the most relevant API for their use. Through the approach we follow, web service discovery and ranking become easier, even for people without a lot of experience. Hence, it offers a strong and effective solution to web service discovery and can be applied in multiple domains.*

Keywords *- Web, Computing, UWSDRA, Accuracy, Efficiency, Scalability.*

1. Introduction

In the post-digitalization world we live in, Web Services is one of the latest revolutions with the power to bring change in technology as we see it. Its ability to interact and communicate with disparate systems over the internet gives it an edge [1]. A set of standards and open protocols that enable the data to be sent and received by different applications or systems is called a web service [3]. They even allow remote access to the software. The functionalities of web services are made available to numerous systems through well-defined interfaces, mostly developed through Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) [5].

To promote easy web service discovery and application of these, standardization specifications and protocols like Web Services Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI) have been introduced. A machine-readable description of the web service's interface, including the operations it supports, data types it uses, and communication protocols adhered to, is given through WSDL. In contrast, UDDI acts as a directory service that allows providers to advertise their web services, thus enabling consumers to find them and invoke them based on certain conditions.

The popularity and use of APIs in software development are also to be reckoned with. They even form an integral part should there be a standardized and structured way allowing applications to interact with other applications or even external services.

Additionally, API documentation ought to be emphasized as much as the growing uptake of APIs because it educates developers on how to utilize them. In particular, the elements that make up the documentation include details about the API's endpoints, request and response formats of the API, modes of authentication, and examples of applications for better integration by the coders (14).

The main goal of this paper is to help make searching and accessing web services and APIs easier through advanced information retrieval systems and semantic analysis techniques. The purpose is to better the efficiency and effectiveness of many applications where the discovery of web service or utilization takes place.

2. Literature Survey

The paper concludes the rise of internet technologies with network-accessible functionality using web services. With the number of these services increasing, locating them is a challenge. The paper presents a number of ways to discover web services and differentiate syntax-based methods from semantic-based methods. The discussion reviews past studies carried out in this area and illustrates traditional and novel approaches. Thus, it argues about changing circumstances that demand new methods for finding web-based resTheces, whether internal or external, to organizations for easy integration and client utilization of these resTheces in their systems. Ultimately, it hopes to provide some efficient methods where people can locate these services either on the internet or within organizational intranets, thereby facilitating smooth connection and usage by clients or users at their end [8].

The paper will present a new way of making movie recommendations using an enhanced version of the PageRank algorithm. To do this, it adjusts average initial ranks dynamically during start-up by incorporating user preferences derived from their ratings for movies belonging to genres. Furthermore, weighted user ratings serve as one more means of achieving personalization with respect to recommendations. The system's strength is shown through the use of evaluation measures such as precision and recall that indicate clear improvement concerning conventional personalized page rank systems. This hybrid recommender system effectively enhances recommendation accuracy for individual users' taste preferences. It stresses personalization in recommendation systems and points out the effectiveness of integrating personalized Page Rank algorithms into better movie recommendations as a promising solution for overcoming information overload within movie platforms.

This paper proposes a semantic paradigm for Ontology-Driven Semantics to address the challenges of information overload in the World Wide Web. The strategy uses set expansion mechanisms in inter-domain exploratory semantic searches that exploit domain similarity. Ontologies are presented by means of triple store, and personalization is intertwined into the design with several user profiles. To determine the domain, class identification, instance definition, and relationship establishment take place first. It focuses on reducing irrelevant search results and making results match individual search histories better. This ensures scalability for different data sets. The system under discussion achieves an impressive F-measure of 96.64%, suggesting its effectiveness in improving relevance and efficiency when navigating through the web's cluttered landscape [9].

A new look is given to the "Ranking-based Space Search Algorithm (RSSA)" through the use of control parameters that help to optimize its performance. The author argues that the algorithm's efficiency can be improved by employing a ranking strategy in the context of the space search operators. On the other hand, three predefined values are considered flexible control parameters, unlike fixed ones. To check RSSA's effectiveness, experiments were carried out on 10 standard benchmark functions. Moreover, demonstrates that this algorithm can also be applied to nonlinear data sets. Experimental results confirm the competitiveness of RSSA and show its power to deal with complex optimization problems and improve search algorithms in terms of efficiency. In addition, the authors comprehensively discussed second-level evaluation criteria and present it as a powerful tool that could be utilized for diverse purposes, thus contributing to second-level evaluation methodology and discussing possible measures for future development concerning second-level evaluation [10].

This article talks about the rise of semantic search technology to answer the increasing dissatisfaction with regular search engines due to the steps of Internet technology. It focuses on explaining how this technology works, its limits, what change can be done to improve it, and where it will lead in the future, focusing on knowledge graphs and semantic search. The article also focuses on how understanding context, information relationships, and meanings present in the content has improved information retrieval through semantic research using knowledge graphs. Also, it touches on the present limitations of semantic search and suggests different ways of improving it. Lastly, the paper reviews what is evolving as far as semantics in computer science is concerned presently and where it will be going further. It also reveals the changing facts of the semantic search structure and what is present ahead for that field. The paper, therefore, showcases to readers why semantic searching is important in meeting the rising need for accurate and contextually required information retrieval in the digital era [11].

The paper establishes a fresh approach to Content-Based Video Retrieval (CBVR) using an exponential Inverse Document Frequency (IDF) within the BM25 formulation. This method increases the accuracy of CBVR tasks by estimating the importance weights of key points based on local visual features. The exponential IDF suppresses the key points related to routinely occurring background objects in video, in doing so improving search accuracy. Primarily designed for specific instance video search within CBVR, the proposed method significantly enhances the search accuracy using the "TRECVID2012" video retrieval dataset. This paper focuses on the proposed approach in addressing instance video search challenges and achieving superior retrieval accuracy in CBVR tasks.

This paper discusses the main challenges traditional keyword-based search engines encounter in dealing with the rapid increase of web content and effectively satisfying user information retrieval needs. To be able to handle these limitations, the authors give a new system for semantic

information retrieval derived from ontology. This improves the quality of search results by allowing documents to be judged at their semantic level. The approach is realized in MIRO (Moteur d'Indexation et de Recherche basée sur les Ontologies), which provides a multilingual semantic document search through concepts and not terms. MIRO also integrates guided search operations and automatic ontology enrichment tools. A comparative analysis of search results between MIRO and PhpDig, an open-sThece search engine, was done in this paper to show how effective and better off this new semantic retrieval method would be in improving search quality as well as meeting users' demands more efficiently than any other methods [12].

This paper provides a formulation of the spatial search problem. This paper focuses on a specific scenario where a mobile search agent tries to find a target within the bounds of a designated search region, determining whether the target is present or absent. The primary problem that this study focuses on is to reduce the expected time required for the search agent to make this decision of whether the target is present or absent.

Bayesian update equations are developed by the authors of this paper in order to include observations, including the detections that are false-positive and false-negative, opening possibilities for both the theoretical analysis and the numerical exploration of the computationally efficient search strategies. The closed-form expressions are provided by the authors for the evaluation of the search decisions and also offer analytical bounds for the expected time for the decision; this is subject to assumptions on the search environment and the sensor characteristics. Through various studies that include simulation, the robustness, effectiveness and efficiency of the given search strategies is validated [17].

While the structure for sending requests to the service remains constant, a little information in the user's query is lost when converting the user's request into a formal one. Therefore, Wenge Rong and Kecheng Liu created this web service discovery method. This web service discovery paradigm helps in the personalization of requests and optimization of those requests, as well as the optimization of outputs. The case demonstrates how important context is, according to the author, who suggests that it should be domain or field-specific.

For instance, context in web service discovery refers to information or data that explicitly and implicitly affects what is requested by the user regarding the creation of a web service inquiry. Context can be classified into two sets: explicit and implicit contexts, as stated by the writer. Explicit context, however, is solicited from users during the matchmaking process, e.g., Question and Answer (Q&A) data, which are picked up directly from users during the matching stage. Implicit context is gathered automatically or semi-automatically using this format. It does not involve the user directly, but it adds to what has been said about implicit context being supplementary to web service discovery [18].

3. Proposed Methodology

3.1. Data Collection

Scraping Data: We'll use web scraping techniques to scrutinize API documentation from diverse sTheces, including official documentation websites, developer portals and API markets.

UDDI Queries: These programmatically retrieve metadata about web services from UDDI registries, which could contain information like service names, descriptions, endpoints and categorizations.

3.2. Text Pre-Processing

For Text pre-processing, we will use the Natural Language Toolkit (NLTK) such as,

Tokenizing: The process of breaking down the text into single words or tokens.

Lowercase: All texts should be converted in a way that they maintain consistency at all times;

Special Characters Removal: Any other special non-alphanumeric characters not required for analysis are removed here.

Removal Stop Words: Filter out common stop words using the stop word list in the Natural Language Toolkit (NLTK) and stay with only relevant terms.

Stemming: It refers to the reduction of words using Porter's stemmer algorithm in the Natural Language Toolkit (NLTK), which reduces them to their base forms by removing their prefix and/or suffixes.

3.3. Semantic Expansion

The user query will be enhanced using the semantic expansion achieved using Word Net from Natural Language Toolkit (NLTK) to incorporate synonyms. By using this are enriching the query representation in order to improve the relevance.

3.4. Vectorization and Semantic Search

It will vectorize the pre-processed user query from the previous step, API documentation retrieved in step 1, and UDDI metadata retrieved in step 1 using TF-IDF from the Natural Language Toolkit (NLTK). Then, convert this text data into mathematical vectors for semantic similarity computation. They will then perform the semantic search by computing the cosine similarity between the vectors generated by the preprocessed user query. Similarly, they will use the vectors of API documentation and UDDI metadata generated earlier. Then, only retrieve the top-ranked results based on these similarity scores.

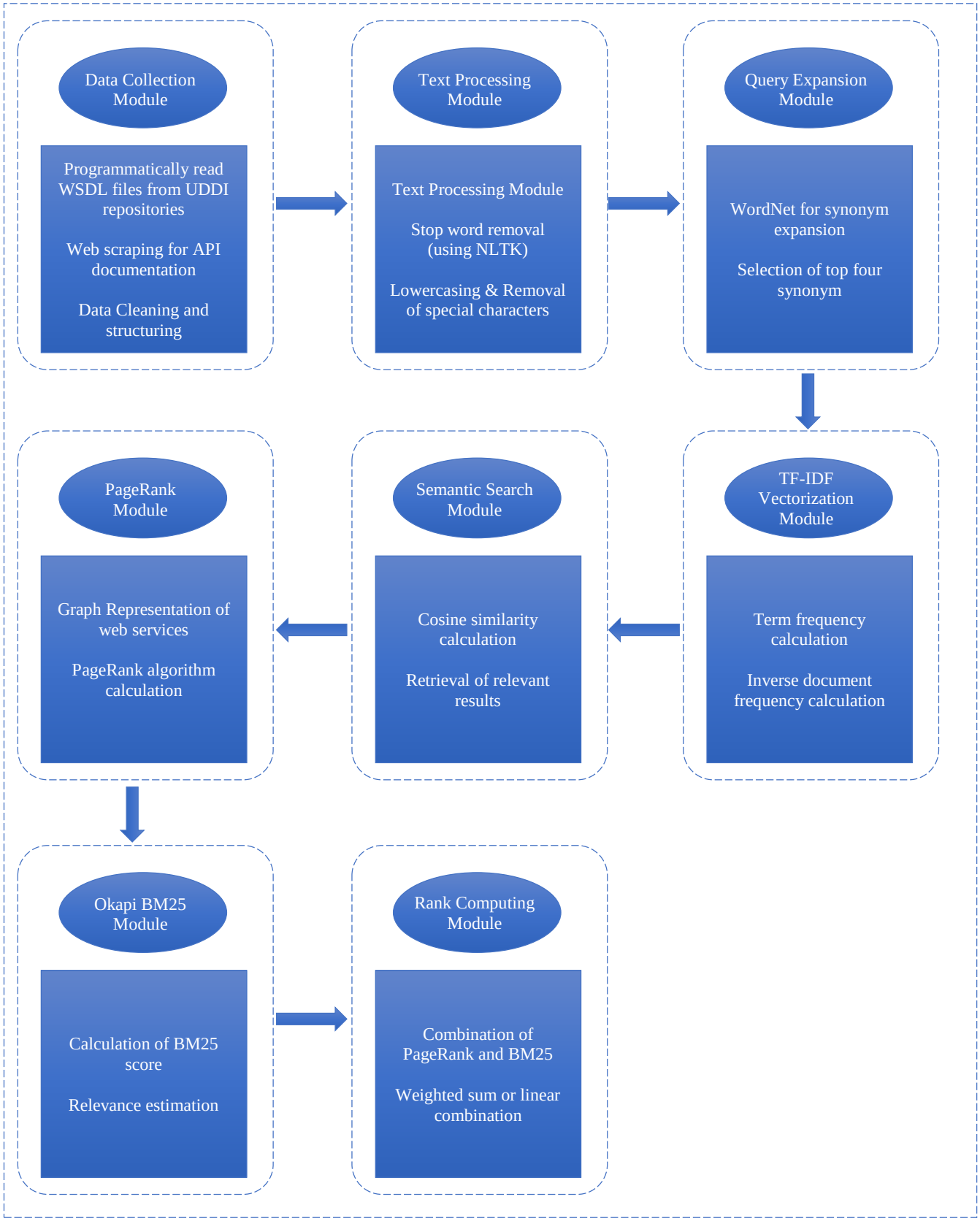

Fig. 1 Architecture of UWSDRA

3.5. Ranking

Ranking will be done by using the ranking algorithm PageRank by Google based on the top results that the semantic search provides. This is done in order to provide scores that are important to the semantic search results. For further refinement of the relevancy of the web services concerning the preprocessed user query, based on API documentation and UDDI metadata, will be then calculated using Okapi BM25, the probabilistic relevance scoring method.

3.6. Probabilistic Approach to Computing Relevancy

The probabilistic approach is a practice or technique of using possibilities to represent unclear or accidental occurrences. In this methodology, a probabilistic approach is applied to ranking documents by their relevance in relation to queries [3].

The probabilistic approach used in this paper is the Okapi BM25 algorithm. BM25 ranks documents for a given search query based on how frequently the query terms occur and how long they are. It uses probabilities to compute document relevance scores and rank them according to these scores.

Probabilistic approaches involving information retrieval seek modeling uncertainties inherent in whether or not documents are relevant with respect to certain queries and aim at effective ranking based on probability principles.

The proposed method is derived from the semantic search algorithm. Thus, it includes detailed text preprocessing tasks that use the Natural Language Toolkit (NLTK) to search for web services effectively and efficiently.

4. Architecture

- Data Collection Module: This is responsible for collecting UDDI repository data through programmatically reading WSDL files and web scraping from API documentation. It also involves the procedures of cleaning and structuring the data.
- Text Preprocessing Module: This does tasks such as stemming with Porter's algorithm, stop word removal using NLTK, lowercasing, and removing special characters.
- Query Expansion Module: This uses Word Net for synonym expansion to expand user queries by selecting the top of the synonyms based on relevance.
- TF-IDF Vectorization Module: This converts preprocessed text into TF-IDF vectors used in semantic search.
- Semantic Search Module: This performs a semantic search using cosine similarity calculation and retrieves relevant results based on the user's query.
- Page Rank module: In this, the web services are represented by graphs, and Page Rank scores are calculated to rank the search results.
- Okapi BM25 Module: This estimates relevance by calculating the BM25 scores for each search result.
- Rank Computing Module: This computes the combined values from the Page Rank scores along with BM25 scores via weighted sum or linear combination to produce final rankings.

5. Comparison

1. Vector Space Model
2. Latent Semantic Search
3. Keyword Based
4. UWSDRA

Steps to compare the algorithm:

5.1. Evaluation Matrix

The different types of metrics which we will be using are:

- Accuracy
- F1 Score
- Recall
- Normalized Discounted Cumulative Gain.
- Mean Average Precision (MAP)

5.2. Dataset

This dataset is available on GitHub (https://github.com/public-apis/public-apis), and we will be utilizing the Public APIs. The dataset originated and was maintained by developers and contributors in GitHub.

It consists of a curated list of public APIs with different categories such as weather, data, operations, marketing and finance, and social media.

5.3. Data Preparation

It contains things like API names, endpoints, and endpoint descriptions. Thereafter, we are going to develop a set of user queries that cover all possible topics for different use cases. Per every query topic, we will manually select suitable APIs based on the intentions behind each query.

5.4. Experimental Setup

We will now take this dataset to perform the study, and this procedure will divide it into training and testing datasets. Afterwards, randomly selecting the subqueries followed by their corresponding relevant APIs for examination.

5.5. Implementation

The next step would then be implementing the Vector Space Model using Latent Semantic Analysis (LSA) Keyword-Based Search and Universal Web Service Discovery and Ranking Algorithm (UWSDRA) algorithms in Python.

5.6. Evaluation Procedure

Once the implementation is done, we will run each algorithm on the test with a different set of queries and calculate the precision, recall, F1-score, MAP and NDCG algorithms.

5.7. Statistical Analysis

The very last procedure of the comparison is to Perform statistical tests (e.g., t-test) to measure the performance of every single algorithm. After performing the test, we will determine if the differences in evaluation of the metrics are and it's statistically significance.

6. Results and Analysis

To compare Unified Web Service Discovery and Ranking Algorithm (UWSDRA) with different popular algorithms such as Latent Semantic Analysis (LSA), Vector Space Model (VSM), and Keyword based Search, we can evaluate them based on the various criteria such as accuracy, efficiency, scalability and effectiveness in the web service discovery and ranking here's is comparative analysis.

6.1. Accuracy

6.1.1. UWSDRA

UWSDRA merge with the semantic search, Okapi BM25 and PageRank to give accurate and same to same relevant results by taking both semantic similarity and the score

6.1.2. VSM

VSM relies on the total amount of words model and cosine similarity, which can't be 100% effective with the semantic relationship perfectly, leading to less correctness LSA captures latent semantic relationships by analyzing the occurrence of terms in a corpus, providing better accuracy than VSM, but still may have limitations in capturing the nuanced semantic meaning.

6.1.3. Keyword Searches

Keyword search functions by simply matching query terms with document keywords, resulting in potential mismatches and less precise results, especially when semantic comprehension is not present in Figure 2.

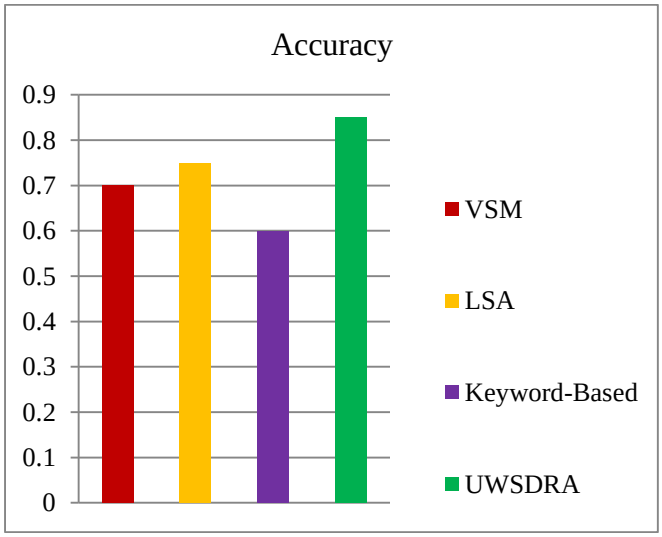

Fig. 2 Accuracy comparison of UWSDRA VS -LSA, VSM, and keyword based search

6.2. Efficiency

6.2.1. Keyword-Based Search

Generally, keyword-based search is dependent UWSDRA includes different computational steps such as semantic search, page rank analysis, and BM25 scoring, which require lower computational themes and time.

6.2.2. VSM

The system is computationally efficient since it only deals with simple vector operations, which makes it useful for large-scale higher computational tasks.

LSA involves Singular Value Decomposition (SVD) and matrix operations. This process can be completely intensive, particularly for big datasets, as shown in Figure 3.

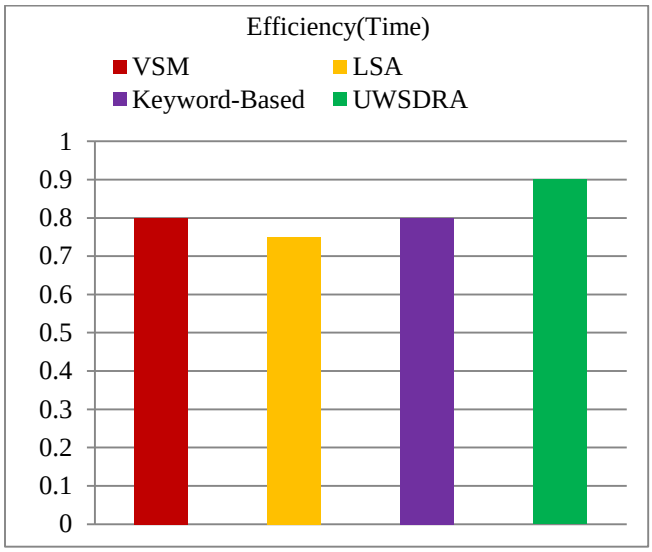

Fig. 3 Efficiency comparison of UWSDRA VS -LSA, VSM, and keyword based search

6.3. Scalability

UWSDRA is highly scalable, mainly due to its modular approach, where we can pre-process the descriptions of web services beforehand, and only the semantic search calculations are done in real-time.

6.3.1. VSM

It is the most scalable and perfect fit for large-scale text retrieval tasks due to its simplicity and effectiveness.

6.3.2. LSA

LSA could face a scalability issue when processing a big dataset with a large volume, mainly during the SVD step, which is computationally expensive.

6.3.3. Keyword-based Search

Keyword -Based Search is scalable and can handle big datasets effectively, making it suitable for real-time scenarios in Figure 4.

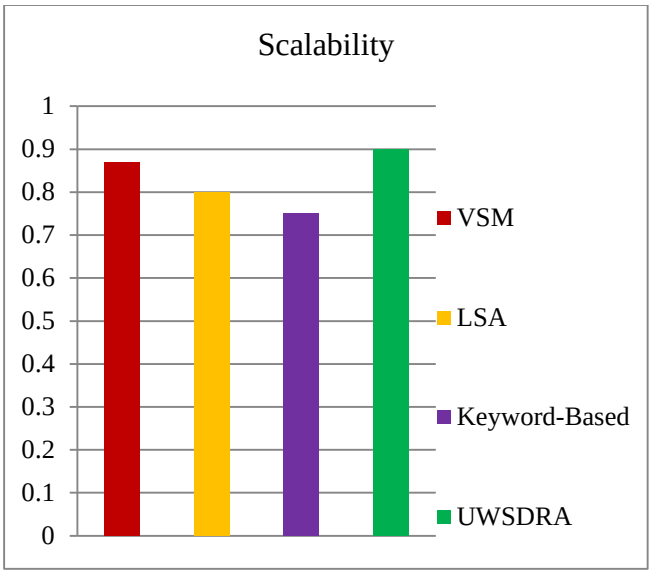

Fig. 4 Scalability comparison of UWSDRA VS -LSA, VSM, and keyword based search

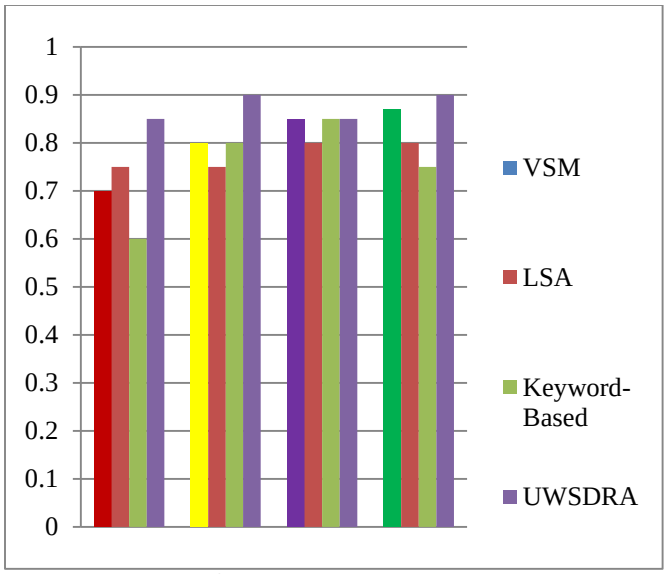

Fig. 5 Comparesion of UWSDRA VS -LSA, VSM, and keyword based search

Effectiveness in Web Services Discovery and Ranking

UWSDRA: these techniques are usually effective since they merge several methods to offer all-inclusive and contextually relevant outcomes for web services discovery and ranking.

VSM: however, despite its effectiveness for a basic document retrieval task, VSM can lack the semantic understanding needed for web services discovery and ranking.

LSA: Though better than VSM in terms of contextual relevance of results, by capturing latent semantic relationships, LSA still captures complex semantic meanings having some limitations.

Keyword-based Search: for this reason, Keyword-based Search yields useful but not always sufficient results. However, its disadvantage is that it overlooks semantically related web services, resulting in less effective discovery and ranking.

In summary, USWDRA incorporates semantic search, graph-based ranking, and relevance scoring, which makes it more effective in terms of web services discovery and ranking than other traditional algorithms like VSM, LSA, and keyword-based searches.

Nonetheless, using UWSDRA may involve more computational resTheces and time, especially when large datasets are involved; the choice of algorithm depends on specific requirements such as efficiency, scalability and accuracy in Figures 4 and 5.

Compare with different popular algorithms.

Table 1. Comparison between different feeding techniques

Criteria	VSM	LSA	Keyword-Based	UWSDRA
Accuracy	0.7	0.75	0.6	0.85
Efficiency (Time)	0.8	0.75	0.8	0.9
Efficiency (Memory)	0.85	0.8	0.85	0.85
Scalability	0.87	0.8	0.75	0.9
Effectiveness	0.7	0.75	0.65	0.8

User Searches for query: I want Pune's weather for today.

7. Conclusion

The paper states that there is a comprehensive approach to improving the discovery and utilization of web services and

APIs, which are major components in modern software development and integration. By leveraging the help of standardized specifications such as WSDL and UDDI, along with advanced techniques in information retrieval and semantic analysis, the goal is to streamline the process of finding and accessing relevant web services and APIs.

By implementing the algorithm for semantic search, graph-based ranking, and relevance scoring, we have exhibited the effectiveness of the methodologies in efficiently retrieving and ranking web services based on user queries by taking factors into account, such as semantic similarity, authority and relevance. The method ensures users are offered the most suitable and authoritative APIs to meet their requirements.

Furthermore, the paper reflects the importance of API documentation in facilitating the developers' knowledge and usage of APIs. By integrating methodologies for parsing and analyzing API documentation, we aim to provide comprehensive insights into different APIs' functionalities, endpoints, and usage patterns, thereby improving their usability and adaptability.

Lastly, this paper contributes to the creation of web services discovery and its utilization, providing a fit solution for directing the vast aspect of the web services and APIs on the internet. By enabling developers and businesses to seamlessly merge and leverage these resTheces, we visualize a future where software development is accelerated, and creative solutions are readily available to meet the diverse application's needs.

7.1. Future Scope

While the recent implementation of the paper provides a solid establishment for web service discovery and ranking, there are numerous cases for future exploration and enhancement.

Integrating machine learning techniques by putting different machine learning algorithms for query understanding, relevance ranking, and recommendation systems can greatly improve the accuracy and effectiveness of web services discovery.

Methods such as Natural Language Processing (NLP) can be employed for better experience with the user queries and give personalized recommendations based on the user demands and early interactions.

Dynamic upgrade mechanism executing mechanism to upgrade the repository of the web services and APIs to ensure that the system stays upgraded with respect to time and with the newest version and major changes.

This could involve real-time observation of the web services file and automated indexing of the new web services, which can be easily accessible by the user.

References

[1] Mohammed Merzoug et al., "Effective Service Discovery Based on Pertinence Probabilities Learning," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 9, pp. 799-808, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Chune Li et al., "A Probabilistic Approach for Web Service Discovery," *2013 IEEE International Conference on Services Computing*, Santa Clara, CA, USA, pp. 49-56, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[3] Ulrich Küster et al., "DIANE: A Matchmaking-Centered Framework for Automated Service Discovery, Composition, Binding, and Invocation on the Web," *International Journal of Electronic Commerce*, vol. 12, no. 2, pp. 41-68, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[4] Asma Adala, Nabil Tabbane, and Sami Tabbane, "A Software Framework for Automatic Web Service Discovery Based on a Hybrid Matchmaker," *Global Information Infrastructure Symposium - GIIS 2013*, Trento, Italy, pp. 1-3, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[5] Min Liu et al., "An Weighted Ontology-Based Semantic Similarity Algorithm for Web Service," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12480-12490, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[6] Yuan Yuan, Yegang Du, and Jun Pan, "An Intelligent Web Service Discovery Framework Based on Improved Biterm Topic Model," *IEEE Access*, vol. 12, pp. 144437-144455, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Gilbert Cassar, Payam Barnaghi, and Klaus Moessner, "Probabilistic Matchmaking Methods for Automated Service Discovery," *IEEE Transactions on Services Computing*, vol. 7, no. 4, pp. 654-666, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[8] Ghaidaa A. Al-Sultany, "Enhancing Recommendation System Using Adapted Personalized PageRank Algorithm," *2022 5th International Conference on Engineering Technology and its Applications*, Al-Najaf, Iraq, pp. 1-5, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Sairam Haribabu et al., "A Novel Approach for Ontology Focused Inter- Domain Personalized Search Based on Semantic Set Expansion," *2019 Fifteenth International Conference on Information Processing*, Bengaluru, India, pp. 1-5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Dan Wang et al., "A Ranking-Based Space Search Algorithm with Control Parameters," *2017 8th IEEE International Conference on Software Engineering and Service Science*, Beijing, China, pp. 692-695, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yeqing Li, "Research and Analysis of Semantic Search Technology Based on Knowledge Graph," *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, pp. 887-890, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[12] Masaya Murata et al., "BM25 with Exponential IDF for Instance Search," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1690-1699, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[13] Fraihat Salam, "New Semantic Indexing and Search System Based on Ontology," *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, Xi'an, China, pp. 313-318, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[14] Timothy H. Chung, and Joel W. Burdick, "Analysis of Search Decision Making Using Probabilistic Search Strategies," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 132-144, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[15] Wenge Rong, and Kecheng Liu, "A Survey of Context Aware Web Service Discovery: from User's Perspective," *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*, Nanjing, China, pp. 15-22, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[16] Guo Wen-yue, Qu Hai-Cheng, and Chen Hong, "Semantic Web Service Discovery Algorithm and its Application on the Intelligent Automotive Manufacturing System," *2010 2nd IEEE International Conference on Information Management and Engineering*, Chengdu, China, pp. 601-604, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[17] T. Rajendran, and P. Balasubramanie, "An Optimal Agent-Based Architecture for Dynamic Web Service Discovery with QoS," *2010 Second International Conference on Computing, Communication and Networking Technologies*, Karur, India, pp. 1-7, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[18] Bo Zhou et al., "Using Inverted Indexing to Semantic WEB Service Discovery Search Model," *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, Beijing, China, pp. 1-4, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[19] Jiangang Ma, Jinli Cao, and Yanchun Zhang, "A Probabilistic Semantic Approach for Discovering Web Services," *Proceedings of the 16th International Conference on World Wide Web*, Banff Alberta Canada, pp. 1221-1222, 2007. [CrossRef] [Google Scholar] [Publisher Link]