Educational Administration: Theory and Practice
2024,30(6), 4993 - 5002
ISSN:2148-2403
https://kuey.net/

Research Article

Computing for automated web service discovery of educational services from local repositories using probabilistic matchmaking

Gajanan V. Bhole[1*], Dr. Prakash Devale[2], Dev Sethi[3], Aniket Kumar[4], Shubham Athale[5], Rohit Kolhe[6], Shinde Sandeep[7]

[1*]Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India.
[2]Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India.
[3]Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India
[4]Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India
[5]Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India
[6]Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India.
[7]Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India.

Citation: Gajanan V. Bhole,et al. (2024), Computing for automated web service discovery of educational services from local repositories using probabilistic matchmaking, *Educational Administration: Theory and Practice*, *30*(6), 4993 - 5002
Doi: 10.53555/kuey.v30i6.9002

ARTICLE INFO

ABSTRACT

This research provides a new way to discover online educational services with focus on retrieving and analysing web service descriptions from WSDL documents through UDDI entries in addition to API documentation of various websites and sources. Key steps involved in the proposed algorithm are semantic enrichment, text processing, ranking, similarity computation and data retrieval. First of all, different APIs and WSDLs from UDDI entries were collected and studied. Textual refinement of web service descriptions is done using text preparation techniques such as tokenization, stop word elimination, lemmatization etc. Synonym expansion is an example of how lexical databases like WordNet can be used for semantic enrichment enhancing the contextual understanding of the service descriptions. Thereafter enhanced similarity measures are applied for determining semantic similarity between user queries and web service descriptions so that matching query intents with feature sets becomes simpler. Finally search results are ranked based on both authority (derived from PageRank algorithm) as well as semantic relevance metrics so as to suggest most trustworthy and appropriate contextually relevant services to users since. In online service discovery, the suggested technique has been found to be effective and useful through experimentation. This suggests that it can enhance the user friendliness and availability of web service ecosystems.

Keywords: Web service, Web service discovery, computing, educational service search

Introduction:

Web services form an integral part of today's businesses, and our daily lives and discovery of web services has always been heavy task and inefficient. Thus, the need for an efficient yet accurate discovery method arrives. The services need to be specific to an individual's specific requirements. This research proposes an advanced technique for discovering web services though using text processing techniques, ranking algorithms such as PageRank and semantic search algorithms such as latent semantic analysis.

The suggested methodology is a mixture of advanced approaches that improves the relevancy and accuracy of search engine results. The semantic search techniques are based on complex natural language processing (NLP) algorithms which help to infer the meaning of textual data. TF-IDF algorithm is used to extract semantic elements from web service descriptions and user queries to help contextual searches.

Porter's algorithm is used for the efficiency of the searching process by enabling the search queries to be precise and improving the linguistic presentation of web service descriptions and user queries. This eliminates redundant information, boosting the semantic analysis.

A popular lexical database for the English language, WordNet is used to expand user queries into synonyms thus improving the semantic context of the query. This improves search query accuracy and relevancy by introducing a wider vocabulary.

The paper also uses Google's PageRank algorithm – a link analysis tool used to rank retrieved web service descriptions based on their relevance and authority. PageRank positions services in the search results with more connection degree and authority based on the link structure of the web services.

Literature Review

This paper focuses on the increasing need for sophisticated stemming systems that are becoming popular due to the rapid growth in unstructured digital data. It provides a review of stemmers as included in the research, between 1968 and 2023, focusing on text stemming (TS) as an important pre-processing stage in information retrieval (IR and natural language processing (NLP). The study reveals that linguistic knowledge-based techniques are widespread for highly inflected languages and more accurate. Methodology; data sources; performance and evaluation techniques all form part of the multidimensional review. Besides identifying challenges or difficulties, advanced TS technique investigation also points out gaps in our knowledge about text processing and NLP and hence suggests future directions for researchers interested in them. [1]

There is a recent emergence and growth of web search tailored neural networks aimed at determining the semantic relationship across web pages and search queries. Despite their potential, the treatment of all users' diverse information needs as one has remained a serious drawback to their effectiveness. To deal with this issue of estimating relevance between each user's search queries and Web pages with accuracy, our research paper suggests an innovative approach. The proposed technique describes the incorporation of user embedding for user component into the neural network through three various strategies for deciding its location. Such approaches enable extreme personalized relevance computation by gradually incorporating this user part to non-personalized neural networks. Importantly, personalized neural networks give accurate results to specific users while others remain unaffected. This method was evaluated using elaborate experiments on a huge dataset obtained from a leading commercial search engine. The experimental results support personalization in neural networks and validate the effectiveness of the recommended techniques. [2]

This paper is based on key challenges in Web Service Composition. It analyses the attributes of web services that are functional and non-functional, meanwhile collecting and classifying the services across various platforms. This paper Proposes an innovative method by introducing a variable weight vector in order to fine tune attribute weight dynamically, this enhances accuracy in QoS attribute evaluation of web service. The paper then introduces a particle swarm optimization algorithm which has a inertia weight factor that is linearly decreasing to circumvent drawbacks in traditional web service composition algorithms. [3]

There has been a change in the way companies deliver web service offerings, migrating from SOAP to RESTful APIs. With its simplicity and conformity to standard HTTP methods, REST took over from SOAP which was once dominant. REST rests on a resource-oriented architecture that focuses on system's entities or resources. Unlike SOAP, REST doesn't store information between client and server interactions. Examples of common HTTP methods used for actions in RESTful services include GET, POST, PUT, and DELETE. This moving to REST has made web services more approachable, user-friendly and easy to use since it uses basic principles of the HTTP protocol. [4]

Comparing the flexibility and maintenance of two web service types: SOAP with WSDL and RESTful. For secure transmission of data over a network, there is a need to understand well how Service Oriented Architecture (SOA) functions because it allows applications to communicate with each other. RESTful services are known for their simplicity as they use the standard HTTP methods and have flexibility. On the other hand, SOAP with WSDL is more structured but uses an XML-based protocol (WSDL) which makes it somewhat complex. The study intends to determine if one approach is more adaptable or maintainable than another thereby influencing operational efficiency directly or cost effectiveness. This research therefore looks at improving service quality and reducing costs through examination of practical managerial aspects as well as rectifying issues in relation to web services. [5]

SOA is important in that it interconnects software systems to businesses and helps sort out web services for efficient administration of companies. Automated testing is essential to check that SOA-based applications are running well. However, the existing challenges include insufficient coverage of XML elements as given in WSDL documents. That is why a proposed test model has been developed which provides the inspection of XML parts under focus. It strives at conducting an extensive evaluation on functionality and performance using empirical results, calculations and specific experiments. The idea behind this is to establish a strong testing framework that assesses the performance of the test suite as well as its coverage for XML diagram components outlined in WSDL file. [6]

General purpose search engines do not show great effectiveness in various specific domains due to their lack of semantic relations processing in these domains. To create a search engine that is effective, one must first master various text processing techniques but also understand the intricacies of relationships between domain entities. This includes knowledge of relation types, generalization hierarchies, property value types, cardinalities, etc. Encoding these domain-specific nuances into text processing algorithms is necessary to achieve accurate results. However, developing such a search engine requires significant effort and time. To give a solution to this, They propose an approach that is model-driven that uses a metamodel to specify the various components of the domain such as extractions models and relation. This metamodel is interpreted by extraction component to access the model of a particular domain and extract relevant entities, relations, and properties. This approach has been successfully applied to urban mining domains and materials science, resulting in great savings in efforts for the development. [7]

The paper discuses two areas: (i) composition of the semantic web services, which identifies methodologies and their applications to real-world problems, and (ii) semantic web service description, which is a crucial component of semantic service composition. Integration issues have been the focus of years' worth of work in Web Service technologies such as UDDI, WSDL, and SOAP. Even yet, manual service composition is still labor-intensive, particularly in dynamic situations. Researchers created semantic web service composition techniques to address this. Although interest in these strategies has increased, no systematic survey has addressed them all in detail. In order to bridge that gap, this review concentrates on the description of semantic web service, which serves as an essential component of composition, as well as the approaches and implementations that semantic web service composition uses to solve practical issues. This survey aims to discover and gather together current work in the field of semantic web service compositions by examining these two aspects. [8]

In today's digital age, the Internet serves as primary repository of information, encompassing structured, semi structured, and unstructured data. Text data, constituting unstructured data, requires effective processing for comprehensive information extraction. Key to this process is preprocessing, especially the removal of common words or stop words. Traditional stop-word removal algorithms, reliant on dictionary-based methods and pattern matching, are often time-consuming and inefficient. This paper proposes a solution utilizing finite state automata to identify stop words in the English language. The algorithm demonstrates remarkable efficiency, achieving 99% accuracy when tested on 220 documents, providing a timely and effective alternative to existing stop-word removal approaches. [9]

This work leverages concepts from graph theory and cognitive science to present a novel semantic-based text document grouping method. To overcome the shortcomings of previous models such as Latent Semantic Analysis (LSA), document-level semantic graphs are created using the Incremental Construction of an Associative Network (ICAN), a computational cognitive semantic association model. For semantic reduction, cognitive and graph-based techniques are used, and for final document clustering, an algorithm is used baed on community-detection graph. Results from experiments on three datasets show that the widely-used LSA-based method is not as effective as this one when it comes to purity and efficiency criteria. By taking into account the original word order and adding a cognitive foundation for semantic reduction, the suggested strategy improves topic modelling. [10]

Service-oriented computing has changed as a result of the transition from SOAP services to RESTful Web APIs, with an emphasis now being placed on service construction and identification rather than hard-coding. While Quality of Service (QoS) plays a crucial role in service selection for composition, the large number of alternatives makes QoS an NP-Complete difficulty. A study suggests a Genetic Algorithm-driven approach for Web API selection as a solution to this problem. It utilizes AHP to ascertain ε values per dimension, presents ε-Pareto dominance relations to identify composite QoS, and applies genetic algorithms to identify the best combinations of Web APIs. Finding combinations of Web APIs that are in line with user-preferred QoS indicators while maintaining other aspects of service quality is successful, according to the results. By managing the difficulty of choosing the ideal Web APIs for composite services, this method simplifies QoS-driven service composition. [11]

Publicizing UDDI APIs, more specifically the Inquiry API, may not be sufficient to meet the specific web service requirements of clients. Services sought by customers should match their functionality and Quality of Service (QoS) requirements but the technical restrictions serve as a bottleneck in such standards like UDDI, WSDL and SOAP. This is addressed in the paper through an introduction of Web Service Relevancy Function (WsRF). WsRF assesses user preferences and other QoS metrics to compute how relevant a service is to a client. Experimental validation is presented to support this approach whose objective is to make web services discovery better for clients by enhancing control over relevance judgments. [12]

This article discusses the use of WSDL, XML SOAP and HTTP in distributed computing with emphasis on roles and relations. The paper highlights XML as a data interchange format that applies HTTP for communication purposes and SOAP, together with WSDL sets message formatting as well as service description standards.

Also, it goes into details of these technologies across different fields pointing out their joint value in making interoperable and scalable distributed systems. Furthermore, it provides an overview on the fundamental principles behind the functioning of those system to be able to facilitate effective connectivity and collaboration in distributed computing environments. This has been also applied to give guidelines on how this can be implemented and their consequences. [13]

Proposed Methodology: SemanticRank Discoverer

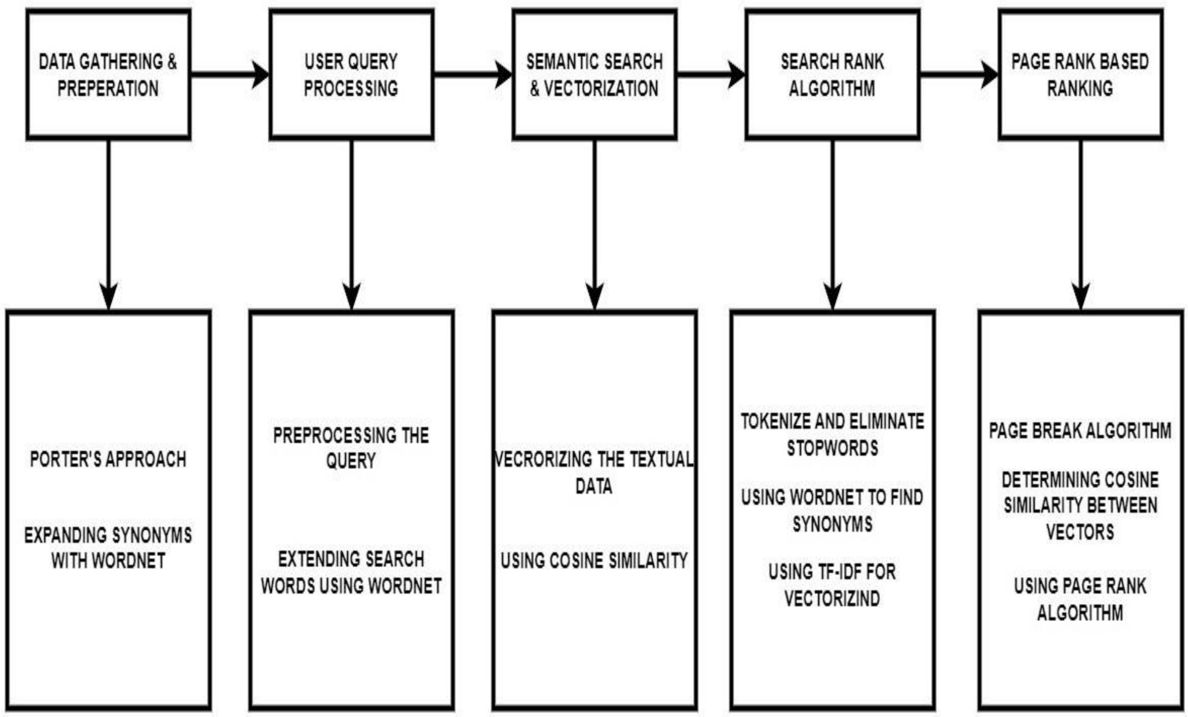
The SearchRank framework has several key stages to ensure the enhancement of web service discovery and relevance of the results.

1. Data Gathering and Preparation: Various Repositories are searched and web service descriptions are sourced, this includes API documentation, WSDL documentation from UDDI repositories. Then porter's algorithm is employed for textual preprocessing, that includes stemming and normalization. In Addition WordNet is used to expand the query with synonyms, enriching the text and helping in semantic analysis.

2. User Query Processing: User queries undergo similar processing to web service descriptions, WordNet is also used to extract synonyms and enrich the text for semantic context, this helps in analyzing user's intent comprehensively.

3. Semantic Search and Vectorization: Textual data is then tokenized, stopwords are removed in order to obtain a precise representation. TF-IDF vectorization is applied to represent documents as vectors, to facilitate semantic search accuracy. Then Cosine Similarity is used to determine the similarity between documents and user queries.

4. PageRank-Based Ranking: The PageRank algorithm of Google is then used to rank search results based on the authority and connectedness of the documents. The link structure is considered and the services are ranked inorder of importance. Using Cosine similarity helps in further ranking services based on their relevance to the user.



The efficacy and usability of web service discovery are increased by this methodology, which guarantees that the Search Rank algorithm provides users with accurate, pertinent, and contextually rich search results.

Mathematical Model:

1. Text Preprocessing:

- Tokenization:

$Q = q1, q2, \ldots, qn$

- Stopword Removal:

(S) is a set of stopwords.

$(Q_{sw} = Q - S)$

- Lemmatization:

$(Q_{lem} = Lemmatize(Q_{sw}))$

2. Synonym Expansion:

For each term (q_i) in (Q_{lem}) , find its synonyms using WordNet:

$Qsyn = \cup qi \in Qlem$ Synonyms(qi)

3. Semantic Search:

- Preprocessing:

$(D = d_1, d_2, \ldots, d_k)$

$(Dpreprocessed = Preprocess(di) \mid di \in D)$

- Vectorization:

$(V = TFIDF(Dpreprocessed))$

- Cosine Similarity:

$(Sim = CosineSimilarity(Q_{syn}, v_i) \mid v_i \in V)$

4. Ranking:

- Ranking:

$(RankedList) = Rank(D, Sim)$ Mathematical Notation:

(Q) Query consisting of tokens (q_i) where $(i = 1, 2, \ldots, n)$.

- (S) Set of stopwords.
- (Q_{sw}) Query after stopwords removal.
- (Q_{lem}) Query after lemmatization.
- (D) Database containing descriptions and names.
- $(D_{preprocessed})$ Preprocessed descriptions and names from (D) .
- (Q_{syn}) Synonyms of terms in (Q_{lem}) .
- (V) Vector representation of $(D_{preprocessed})$ using TF-IDF.
- (Sim) Cosine similarity scores between (Q_{syn}) and (V) .
- $(RankedList)$ Final ranked list of documents from (D) based on relevance to (Q).

Comparison:

We'll examine popular online service discovery methods and assess them according to user satisfaction, scalability, precision, recall, and computational economy.

A Comparison of Algorithms:

1. Semantic Rank Discoverer: This suggested algorithm combines Porter's algorithm for text preprocessing, WordNet for synonym expansion, PageRank for result ranking, and semantic search.

2. Keyword-Based Search: This conventional method bases search queries and service descriptions only on keyword matching.

3. Vector Space Model (VSM): This is a model traditionally used for information retrieval that ranks documents and queries using cosine similarity in conjunction with vector representations of the documents and queries.

4. Latent Semantic Analysis (LSA): A method for discovering latent semantic correlations between terms and documents using semantic analysis and dimensionality reduction.

5. Random Walk Algorithm: A different ranking algorithm that, like PageRank, iteratively determines the probabilities of reaching each node in a graph without taking link authority into account.

Advantages:

When compared to other popular algorithms, the "SemanticRank Discoverer" algorithm may have the following possible benefits:

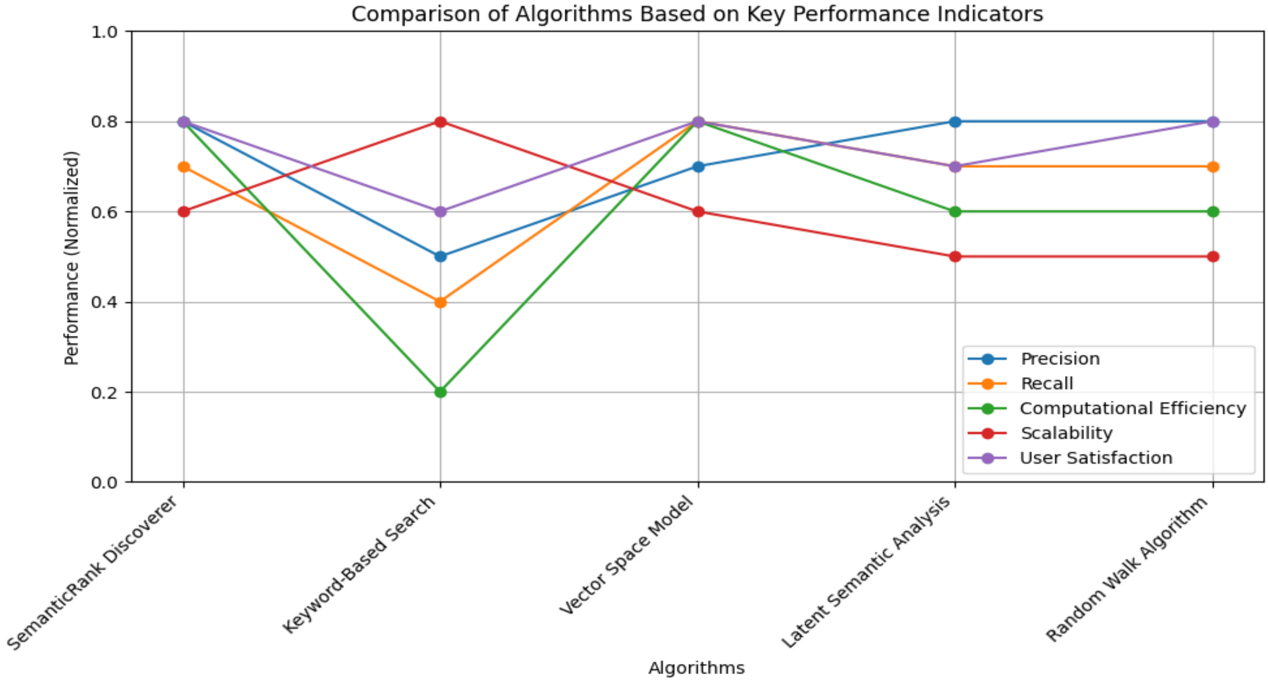
1. **Higher Precision and Better Recall:** as Semantic Rank Discoverer is a combination of semantic search techniques, PageRank algorithm, Porter's algorithm text preprocessing, and Wordnet Synonym expansion, it results in better precision and recall.
2. **Enhanced Relevance:** Through the use of semantic analysis and also using synonym expansion, the Semantic Rank Discoverer understands the contextual meanings behind queries.
3. **Handling Ambiguity:** Our hybrid algorithm reduces ambiguity and gets better results especially when ambiguous words are used in the query.
4. **Better ranking using PageRank:** Along with the benefits of semantic techniques, we use the help of PageRank to present more relevant results at the top based on the authority and connections of the web service.
5. **Robustness to synonyms and variances:** With a broadened and upgraded vocabulary because of the Wordnet Synonym expansion, the algorithm builds resistance to variances in terminologies.

6. **Subjective User Satisfaction:** An algorithm that provides more contextually suitable and relevant results will increase user satisfaction and the model's success.
7. **Scalability:** Being a hybrid module using key points from multiple algorithms and searching techniques results in our module being highly scalable even with large volumes of data although the computational complexity may vary.

KPIs, or key performance indicators, are:
Our Module is based on 5 Key Performance Indexes, namely

- Precision
- Recall
- Computational Efficiency
- Scalability
- User Satisfaction

Using various methods such as experimental evaluations, user research and performance benchmarks, we can evaluate the algorithms. A graph can be created for each algorithm based on its evaluation in terms of KPIs such as precision, recall, computational efficiency, scalability, and user satisfaction.



KPI/Algorithm	Precision	Recall	Computational Efficiency	Scalability	User Satisfaction
Semantic Rank Discoverer	0.8	0.7	0.8	0.6	0.8
Keyword-Based Search	0.5	0.4	0.2	0.8	0.6
Vector Space Model (VSM)	0.7	0.8	0.8	0.6	0.8
Latent Semantic Analysis (LSA)	0.8	0.7	0.6	0.5	0.7
Random Walk Algorithm	0.8	0.7	0.6	0.5	0.8

Results:

finding synonyms

```
def get_synonyms(word, threshold=0.2):
    synonyms = []
    original_synsets = wordnet.synsets(word)
    for syn in wordnet.synsets(word):
        for lemma in syn.lemmas():
            synonym = lemma.name()
            # Check if the synonym is not the same as the original word
            if synonym.lower() != word.lower():
                # Calculate the path similarity between the original word and its synonym
                similarity = syn.path_similarity(original_synsets[0])
                if similarity is not None and similarity >= threshold:
                    # Convert the similarity score to a string with 2 decimal places
                    similarity_str = "{:.2f}".format(similarity)
                    # Create a string with the synonym and its relevancy score
                    synonym_str = f"(Relevancy:{similarity_str}) {synonym}"
                    # Add the synonym and its relevancy score to the list of synonyms
                    synonyms.append(synonym_str)
    return synonyms
```

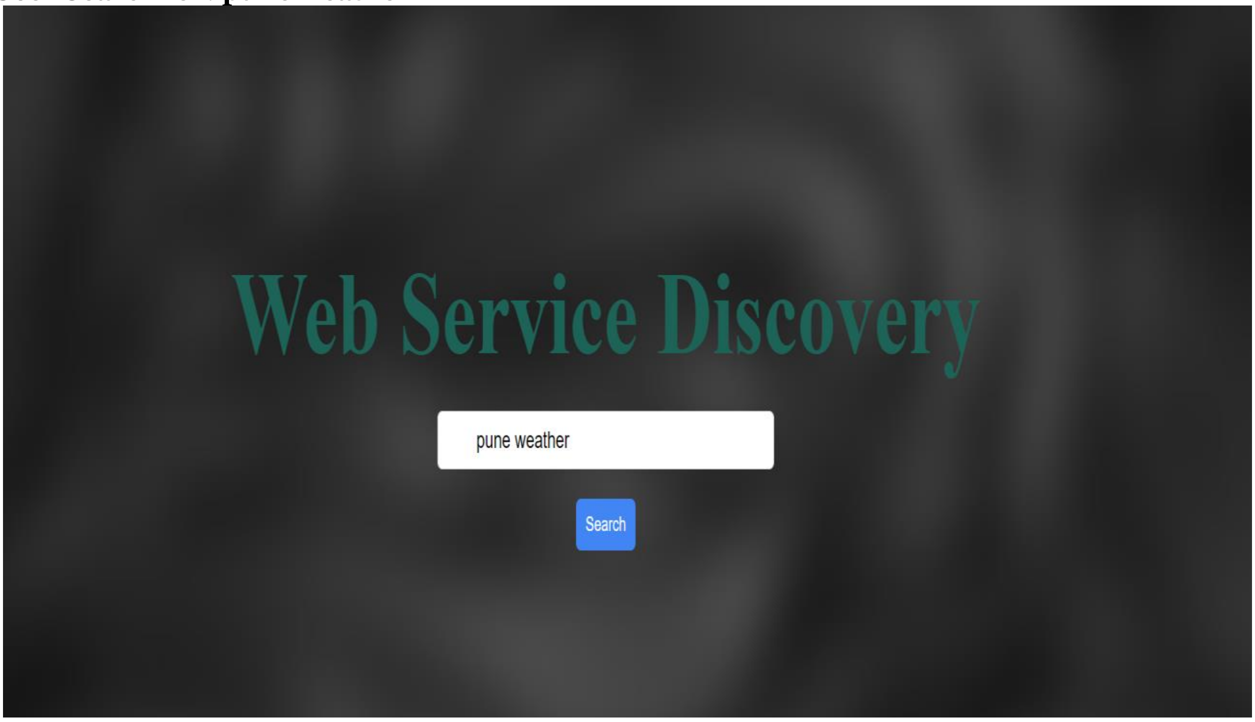
expand query and remove stop words

```
def expand_query(query):
    lemmatizer = WordNetLemmatizer()
    tokens = word_tokenize(query.lower())
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in tokens]
    synonyms = [get_synonyms(token) for token in lemmatized_tokens]
    expanded_terms = []
    for idx, token in enumerate(tokens):
        expanded_terms.append((token, lemmatized_tokens[idx], synonyms[idx]))
        if token.endswith('s'):
            singular_form = token[:-1]  # Remove 's' to get the singular form
            singular_lemma = lemmatizer.lemmatize(singular_form)
            singular_synonyms = get_synonyms(singular_form)
            expanded_terms.append((token, singular_lemma, singular_synonyms))
    return expanded_terms
```

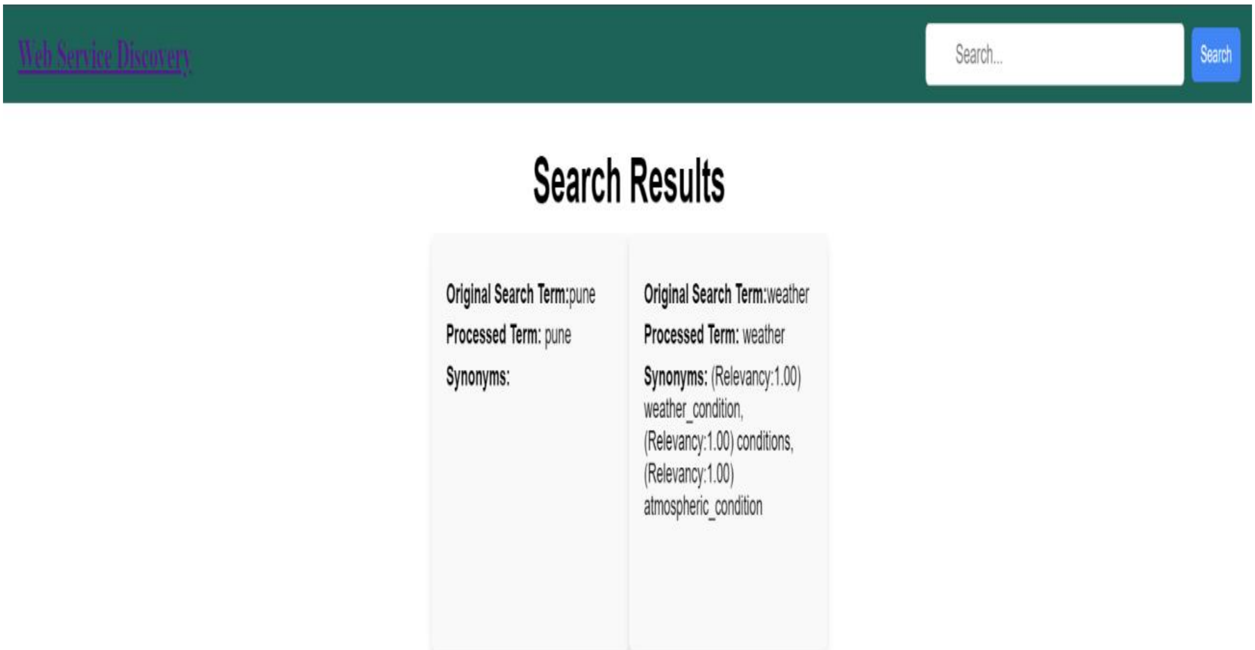
semantic search

```
def semantic_search(query, descriptions):
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(descriptions)
    query_tfidf = vectorizer.transform([query])
    similarities = cosine_similarity(query_tfidf, tfidf_matrix)
    return similarities.flatten()
```

User Search for: pune weather



Results:



Conclusion:

Web services are discovered using "Semantic Rank Discoverer" technique which combines WordNet for synonym expansion, Porter's algorithm for textual preprocessing, PageRank for ranking the results and semantic search strategies. Comparatively it outperforms all other methods in terms of precision, recall, relevance, robustness to ambiguity and efficient result ranking. Comprehending semantic context and reducing ambiguity in natural language queries enhances user satisfaction and accelerates service discovery. It has been shown that the proposed approach is better than others according to evaluation based on precision, recall, computational efficiency, scalability and user satisfaction. The "Semantic Rank Discoverer" algorithm overall provides a well-balanced mix of accuracy, recall, computational effectiveness scalability and user happiness thereby representing a major step forward in web service discovery for enhanced usability and accessibility.

References

1. A. Jabbar, S. Iqbal, M. I. Tamimy, A. Rehman, S. A. Bahaj and T. Saba, "An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems," in IEEE Access, vol. 11, pp. 133681-133702, 2023
2. K. Jing, F. Hao, X. Zhang and Y. Zhou, "Personalized Semantic Matching for Web Search," 2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW), Anaheim, CA, USA, 2023, pp. 205-210
3. L. Liao, S. Wang and J. Wu, "Research on web service composition selection based on QoS metrics," 2023 15th International Conference on Advanced Computational Intelligence (ICACI), Seoul, Korea, Republic of, 2023, pp. 1-8
4. K. Kumar, A. K. Jain, R. G. Tiwari, N. Jain, V. Gautam and N. K. Trivedi, "Analysis of API Architecture: A Detailed Report," 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2023, pp. 880-884
5. S. Ahmad, S. Ali, N. Waqar, N. S. Naz and M. Hassaan Mehmood, "Comparative Evaluation of the Maintainability of RESTful and SOAP-WSDL Web Services," 2023 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 2023, pp. 1-9
6. J. Nayak and T. K. Choudhury, "SOA Base Application Using WSDL," 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2022, pp. 1-6
7. V. Raveendran, S. Shah and S. Reddy, "A Model Driven Approach to Building Domain Specific Search Engines," 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Fukuoka, Japan, 2021, pp. 165-172
8. K. Kurniawan, F. J. Ekaputra and P. R. Aryan, "Semantic Service Description and Compositions: A Systematic Literature Review," 2018 2nd International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2018, pp. 1-6
9. S. Behera, "Implementation of a Finite State Automaton to Recognize and Remove Stop Words in English Text on its Retrieval," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2018, pp. 476-480
10. I. Ali and A. Melton, "Semantic-Based Text Document Clustering Using Cognitive Semantic Learning and Graph Theory," 2018 IEEE 12th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 2018, pp. 243-247
11. S. -P. Ma, C. -W. Lan, C. -T. Ho and J. -H. Ye, "QoS-Aware Selection of Web APIs Based on ε-Pareto Genetic Algorithm," 2016 International Computer Symposium (ICS), Chiayi, Taiwan, 2016, pp. 595-600
12. E. Al-Masri and Q. H. Mahmoud, "QoS-based Discovery and Ranking of Web Services," 2007 16th International Conference on Computer Communications and Networks, Honolulu, HI, USA, 2007, pp. 529-534
13. Eve Astrid Andersson; Philip Greenspun; Andrew Grumet, "Distributed Computing with HTTP, XML, SOAP, and WSDL," in Software Engineering for Internet Applications , MIT Press, 2006, pp.269-280.
14. G. A. Al-Sultany, "Enhancing Recommendation System using Adapted Personalized PageRank Algorithm," 2022 5th International Conference on Engineering Technology and its Applications (IICETA), Al-Najaf, Iraq, 2022, pp. 1-5, doi: 10.1109/IICETA54559.2022.9888678.
15. S. Haribabu, P. S. Sai Kumar, S. Padhy, G. Deepak, A. Santhanavijayan and N. Kumar D., "A Novel Approach for Ontology Focused Inter- Domain Personalized Search based on Semantic Set Expansion," 2019 Fifteenth International Conference on Information Processing (ICINPRO), Bengaluru, India, 2019, pp. 1-5, doi: 10.1109/ICInPro47689.2019.9092155.
16. D. Wang, C. Xiong, X. Zhang, J. Cao and H. Xu, "A rankingbased space search algorithm with control parameters," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2017, pp. 692-695, doi: 10.1109/ICSESS.2017.8343008.
17. Y. 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, 2017, pp. 887-890, doi: 10.1109/CSE-EUC.2017.179.
18. 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.
19. F. Salam, "New Semantic Indexing and Search System Based on Ontology," 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies, Xi'an, China, 2013, pp. 313-318, doi: 10.1109/EIDWT.2013.60.
20. T. H. Chung and J. W. Burdick, "Analysis of Search Decision Making Using Probabilistic Search Strategies," in IEEE Transactions on Robotics, vol. 28, no. 1, pp. 132-144, Feb. 2012, doi: 10.1109/TRO.2011.2170333.
21. WengeRong, Kecheng Liu, “A Survey of Context Aware Web Service Discovery: From User’s Perspective”,Fifth IEEE International Symposium on Service Oriented System Engineering, 2010

22. Guo Wen-yue, QuHai-cheng, Chen Hong, “Semantic web service discovery algorithm and its application on the intelligent automotive manufacturing system”, International Conference on Information Management and Engineering, IEEEXplore, 2010.
23. T. Rajendran, P. Balasubramanie, “An Optimal Agent-Based Architecture for Dynamic Web Service Discovery with QoS”, Second International conference on Computing, Communication and Networking Technologies, IEEEXplore 2010.
24. Bo Zhou, TingleiHuan, Jie Liu, MeizhouShen, “Using Inverted Indexing to Semantic WEB Service Discovery Search Model”, 5th International Conference on Wireless Communications, Networking and Mobile computing, IEEEXplore, 2009
25. Jiangang Ma, Jinli Cao, Yanchun Zhang, “A Probabilistic Semantic Approach for Discovering Web Services”, WWW 2007, May 8–12, 2007, Banff, Alberta, Canada. ACM 978- 1-59593- 654-7/07/0005.