Improved probable clustering based on data dissemination for retrieval of web URLs

Sunita1, Vijay Rana2

1Research Scholar Arni University, India
2Assistant Professor, Sant Baba Bhag Singh University, India

Email: sunitamahajan2603@gmail.com, vijay.rana93@gmail.com

Corresponding Author: Sunita

http://doi.org/10.26782/jmcms.2019.10.00021

Abstract

The programmable paradigm in web technologies is evolving into a web service model where services and information can be reused by distinct users. Diverse information is present over the web and the problem of relevant information discovery based on location is a big challenge for web information retrieval system. Lack of Intelligent classification of information compounded the problem further. This paper presents an approach that extends information similarity analysis using probable clustering procedure and introduces specific results based on the current location of the user using Google location services. To capture the similarity of functional text, feature vector techniques are employed. Dissimilar words are classified as stop words and eliminated from the query string to reduce the complexity of search space. Location sensitive mechanism fetches only relevant information belonging to the current location of a user. Experiments were performed to compare classification accuracy with respect to various models used for feature vector extraction and result in emphasis the effectiveness of Semantic similarity extractor location-based web service model.

Keywords: Intelligent service classification, Natural Language Processing, Location sensitive searching.

I. Introduction

Web technologies [I] use the distributed computing paradigm (WTDCP) including fundamental entities known as services as constructive elements of a complex business system. As per WTDCP, the business application comprises of service-centric applications that are known as a service architecture. Service orientation is critical in the field of B2B, B2C, e-governance, and e-commerce. Web services [II] and its applications are one way of achieving service orientation. In user-specific applications, service orientation can help in designing new applications faster.
thus promoting reusability. Reusability thus reduces search space length causing less execution time while information retrieval [III].

For web applications [IV], the developer either create new services or tries to discover existing services for performing the individual task. The process of discovering an existing mechanism for performing an individual task is known as information discovery as per information workflow. Despite remarkable efforts in the simplifying process, information retrieval simplification is still challenging due to a keyword-based searching mechanism for information retrieval. Web services commonly provided by the web service providers on some service portals that use manual categorization that makes it difficult to disclose relevant services for a given task. It is possible that many services are already created by third-party developers that did not even appear within the search due to service violation issues.

The massive web database searching [V] results in ambiguous and vague information that causes loss of interest for business workflow. The locality sensitive information, thus becomes critical for specific and relevant information for the user. To overcome the problem [VI] proposed a locality sensitive hashing based searching mechanism for the massive database. Based on locality sensitive hashing, the identifier is assigned to each record and clusters similar records based on the similarity index. The process is further refined by reducing the length of an identifier using hamming distance measured in the refinement phase.

This paper proposes a probable clustering mechanism that predicts similar information using word extraction and groups it into a similar cluster [VII]. Feature vector extraction [VIII] mechanism is employed to determine similar features. Different feature vector selection techniques are used in this paper in the service document in vector space. To form clusters, User search query [IX] is parsed and extracted words are matched against extracted features. Group distinct words are labelled according to the frequency they possess. URLs fetching employ JOC tool. The fetched URLs are maintained within the Access Database. This mechanism ensures faster search in case the same URL is searched again. Locality sensitive searching helps in providing specific recommendations to the user. The proposed work resolves the problem of adding semantics and machine understanding capabilities to web searching. This novel approach for effectively extracting feature vector and producing probabilistic clusters yield service categorization using locality sensitive searching mechanism [X]. The result generated through this approach is compared against existing machine learning mechanism support vector machine, naïve Bayes, random forest, and maximum entropy approach. The main contribution of this paper is summarized as below

--An approach that broadens the technique proposed in for extracting distinct functional feature vectors for similarity extraction for effective web search.

--Compounded feature selection mechanism to effectively capturing functionally extracted features.

--Probabilistic clustering mechanism to form similarity and frequency based groups.
--Locality sensitive hash search mechanism for location sensitive, specific results.
--Experimental validation of classification accuracy related to various feature extraction mechanisms.

The rest of the paper is organized as under: section 2 presents a literature survey of techniques used in word sense, clustering and locality search section 3 presents a brief analysis of the methodology used, section 4 gives performance analysis and results, section 5 gives a conclusion and future scope, section 6 gives references.

II. Proposed System

Feature extraction for similarity extraction to reduce the length of search string uses a probable clustering mechanism. The mechanism is described in section 2.1

Similarity function with probable clustering mechanism

The similarity function extracts the similarity values using a sensing mechanism and unique values are listed within clusters. The equation for the probable clustering is represented as

$$\text{sum}_c(a_i, a_j) = \frac{g(a_i, a_j) + \gamma}{\gamma}$$

Equation 1: representing probable clustering matrix formation

Where \(g(a_i, a_j)\) is evaluated as

$$g(a_i, a_j) = \sum_{i=1}^{n} \sum_{j=1}^{n} (S_i == S_j)$$

Equation 2: Representing logical value extraction mechanism for matrix formation

Where \(Si\) and \(Sj\) are similarity matrix vectors. In case \(Si\) and \(Sj\) are equals than \(g(ai, aj)\) will contain logical ‘1’ otherwise it will contain logical ‘0’. The matrix formation is represented with \(\text{sumc}(ai, aj)\). The number of similar words extracted depends upon the sense contained in the database. In case word with similar sense is extracted than matrix contains more than 1 value for that word. ‘ai ‘and ‘aj’ is the probability values.

The matrix formation through probable clustering is given as

|          | F1 | F2 | F3 | F4 | F5 |
|----------|----|----|----|----|----|
| File1    | 0  | 0  | 1  | 2  | 1  |
| File2    | 3  | 3  | 2  | 1  | 3  |
| File3    | 2  | 3  | 0  | 0  | 1  |
| File4    | 1  | 2  | 3  | 2  | 1  |

Table-1: Matrix in the form of frequency

The dimensionality reduction process reduces the dimension without losing any information or without adding any noise information within the original query string. Table 2 indicates the logical matrix corresponds to Table 1
Table-2: Logical Matrix formation

|     | F1 | F2 | F3 | F4 | F5 |
|-----|----|----|----|----|----|
| File1 | 0  | 0  | 1  | 1  | 1  |
| File2 | 1  | 1  | 1  | 1  | 1  |
| File3 | 1  | 1  | 0  | 0  | 1  |
| File4 | 1  | 1  | 1  | 1  | 1  |

The word patterns of “w” is demonstrated as

Table-3: Word patterns of w

|     | X1  | X2  | X3  | X4  |
|-----|-----|-----|-----|-----|
| 0.01| 0.01| 0.01| 0.01| 0.01|
| 0.87| 0.87| 0.87| 0.87| 0.87|

Cluster formation considering Table 3 is given in table 4

Table-4: Cluster formation

|     | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 |
|-----|-----------|-----------|-----------|-----------|
| X1  | 1         | 0         | 1         | 0         |
| X2  | 0         | 1         | 1         | 0         |
| X3  | 0         | 0         | 0         | 1         |
| X4  | 0         | 0         | 0         | 1         |

The tabular formation indicates the dimensionality reduction procedure suitable for clustering and classification process. The similarity function with probable clustering [XI] is used for fetching sense out of the query string in proposed literature.

**Feature Vector mechanism for specific result fetching**

The count vectorization mechanism employed in a proposed system extracts the sense and perform tokenization. This mechanism performs tokenization of the text [XII] document and also builds a vocabulary of known words. The feature vector extraction [XIII] mechanism, first of all, uses encoded vector that is equal to the length of entire documents. The clustering done in section 2.1 is used to count the frequency of occurrence of each word in the document. The form document contains lots of zeros hence it is termed as sparse. The transformation mechanism employed returns sparse matrix. The mechanism used for feature extraction [XIV] is listed as below

**Feature Vector using Count Vectorization**

- Text=File;
- Vectorize=countvectorization(Text)
- Vectorize.fit(Text)
- Print(Vocabulary)
- Vector=Vectorize.transform(text)
- Print(Vector.shape)
This mechanism extracts meaningful words that are to match with keywords for URL extraction [XV]. To be more specific, locality sensitive matching is employed in section 2.3

### Locality sensitive Searching

Locality sensitive mechanism [XVI], [XVII] is used for specific result presentation using Google Location services. The current location of the user is fetched and stored. Result fetched from feature vectorization is matched against the current location. In case the match occurs, the URL is presented to the user. Location sensitive searching mechanism [XVIII]. The proposed mechanism converts documents into small signature using hash function H. in case documents consist of 4 corpora denoted by d then

- H(d) indicates hash key or signature and it is small enough to fit into the memory.
- If similarity associated with keys d1 and d2 is high then probability(H(d1)==H(d2)) is high
- If similarity associated with keys d1 and d2 is low then probability(H(d1)==H(d2)) is low

Once the similarity in terms of URLs is located then match against the location using Google Location services commences [XIX], [XX]. URLs specific to a location is presented hence specific result in the least execution time.

Next section gives the result generated through the proposed system. In addition result through existing literature also presented for proving worth of study.

### III. Performance Analysis and Results

This section provides a result obtained through the proposed literature. Figure 1 plots the number of features extracted without dimensionality reduction and with dimensionality reduction using probable clustering mechanism. Number of documents considered for feature extraction are 200,300,400,500 and 500 and number of extracted features includes 2360,2789,3125,3360,3425 and 3215.

The number of features extracted through probable clustering is reduced significantly. The comparison of singular valued decomposition without dimensionality is made with probable clustering with dimensionality reduction.
Table-5: Number of document and feature extraction without Dimensionality reduction

| Documents | Before DR |
|-----------|-----------|
| 200       | 2360      |
| 300       | 2789      |
| 400       | 3125      |
| 500       | 3360      |
| 600       | 3425      |
| 700       | 3215      |

Fig 1: Number of features extracted without dimensionality reduction using probable clustering

After pre-processing stop word elimination, the number of features extracted is significantly reduced. The comparison is made with singular valued decomposition for feature extraction. The result is highlighted in the figure 2

Fig 2: Number of features extracted through existing and proposed with probable clustering
Training is in the proposed system is accomplished using a hold out rate of 0.3 and the result is extracted using probabilistic clustering. Dimensionality reduction is compared against distinct classifiers. Number of words before and after the proposed approach is given in plot 3.

![Fig 3: Plots of meaningful words extracted through existing and proposed through probable clustering](image)

The training used in proposed system is exponential in nature. The exponential training is faster and complexity is low. The classification accuracy obtained through the proposed system is significantly higher as compared to the cosine distance and Euclidean distance mechanism. This is depicted in figure 4. The classification accuracy for 400 documents with 300 features using Euclidean and cosine distances are compared against the proposed system.

![Fig 4: Classification accuracy with varying hold out rates](image)
As the hold out rates varied, classification accuracy differs and proportion increases. The holdout ration when increased beyond 0.5, classification accuracy shows less variation. The execution time while fetching URLs with dimensionality reduction also show deviation. The execution time reduces as the dimensionality reduction procedure is in place. The random samples are used for determining the final result. In addition execution time also affected through locality searching. The locality sensitive searching mechanism fetches only relevant URLs and presents it to the user. The execution time thus is reduced considerably.

![Fig 5: Execution time with distinct quantity of URLs](image)

**IV. Our Contribution**

In our proposed technique the probable clustering mechanism is used along with location dependent searching that reduces the execution time and avoids collision. In result section the comparison with Euclidean distance, cosine based techniques is given that shows that it gives optimal results. The proposed system avoids collision that occur using collision handling mechanism mid square. It cluster the preferable URL based on the location so the fetching is faster.

**V. Conclusion**

The probable clustering mechanism with location dependent searching provides specific locatable URLs. The URLs found with the proposed mechanism takes much less time as compared to mechanisms without probable clustering. The location-dependent searching using Google location services are done that reduces the overall execution time. The results that are generated are specific to the location. It is much preferable as compare to other search engines because clustering is presented which is based on the location of a particular user.

This work is further enhanced by using location sensitive searching using hashing that makes searching faster. The location sensitive hashing includes the mid square method which calculates the hash values using the squares of middle values of location gathered and after that searching is performed.
References

I. A. R. Patil, “An Innovative Approach to Classify and Retrieve Text Documents using Feature Extraction and Hierarchical Clustering based on Ontology,” International Conference on Computing, Analytics and Security Trends (CAST) IEEE, pp. 371–376, 2016.

II. A. I. Pratiwi, “On the Feature Selection and Classification Based on Information Gain for Document Sentiment Analysis,” Applied Computational Intelligence and Soft Computing, pp. 33-37, 2018.

III. A. Cocos, M. Apidianaki, and C. Callison-burch, “Word Sense Filtering Improves Embedding-Based Lexical Substitution,” In Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications. pp. 110–119, 2017.

IV. C. Xiong and K. Lv, “An Improved K-means Text Clustering Algorithm By Optimizing Initial Cluster Centers,” 7th International Conference on Cloud Computing and Big Data (CCBD) IEEE. pp. 272–275, 2016.

V. D. Sumeet and P. Chowriappa, “Feature Selection and Extraction Strategies in Data Mining,” Data Mining for Bioinformatics, CRC Press, pp. 113–144, 2012.

VI. D. Li, W. Zhang, S. Shen, and Y. Zhang, “SES-LSH: Shuffle-Efficient Locality Sensitive Hashing for Distributed Similarity Search,” International Conference on Web Services (ICWS) IEEE, pp. 822–827, 2017.

VII. F. T. Garc, J. Garc, A. Lucila, S. Orozco, F. Dami, and T. Kim, “Locating Similar Names Through Locality Sensitive Hashing and Graph Theory,” Multimedia Tools and Applications Springer, vol. 10, no. 12, pp. 1-14, 2018.

VIII. H. Shen, T. Li, Z. Li, and F. Ching, “Locality Sensitive Hashing Based Searching Scheme for a Massive Database,” Third International Conference on Digital Telecommunications (icdt 2008) IEEE, vol. 47, no. 52, IEEE, pp. 0–5, 2008.

IX. H. A. Atabay, “A Clustering Algorithm based on Integration of K-Means and PSO,” 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), IEEE. pp. 59–63, 2016.

X. J. K. Mandal, Advanced Computing and Communication Technologies Springer, vol. 452, pp. 494, 2016.

XI. J. Singh Chouhan and A. Gadwal, “Improving Web Search User Query Relevance using Content based Page-Rank,” IEEE Int. Conf. Comput. Commun. Control. IC4, pp. 1-5, 2016.

XII. J. G, “RKE-CP: Response-based Knowledge Extraction from Collaborative Platform of Text-based Communication,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 8, no. 5, pp. 93–98, 2017.

XIII. K. Mishina, “Word Sense Disambiguation of Adjectives using Dependency Structure and Degree of Association Between Sentences,” International Conference on Asian Language Processing (IALP), IEEE. pp. 342–345, 2017.

XIV. M. Lapata and F. Keller, “Web-based Models for Natural Language Processing,” Transactions on Speech and Language Processing (TSLP) ACM, vol. 2, no. 1, pp. 1–30, 2005.
XV. M. Kaur, “Text Classification using Clustering Techniques and and PCA,” *Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, IEEE, pp. 642-646, 2015.

XVI. M. Aydar and S. Ayvaz, “An Improved Method of Locality-Sensitive Hashing for Scalable Instance Matching,” *Knowledge and Information Systems*, vol.58, no.2, pp. 275-294, 2018.

XVII. R. Collobert, J. Weston, and M. Karlen, “Natural Language Processing from Scratch,” *Transactions on Speech and Language Processing ACM*, vol. 1, pp. 1–34, 2000.

XVIII. S. Sharma, Sunita, A. Kumar, and V. Rana, “An Optimum Approach for Preprocessing of Web User Query,” *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 7, no. 1, pp. 8–12, 2018.

XIX. Sunita, and V. Rana,”Removing Ambiguity Problem Based on Clustering in a Web Search,” *First International Conference on Secure Cyber Computing and Communication (ICSCCC) IEEE*, pp. 9-12, 2018.

XX. Z. Jin, Y. Lai, J. Y. Hwang, S. Kim, and A. J. Teoh, “Ranking Based Locality Sensitive Hashing Enabled Cancelable Biometrics : Index - of - Max Hashing,” *Transactions on Information Forensics and Security IEEE*, vol. 60, no.13, pp. 393-407, 2018.

XXI. Z. Lu, Q. Liao, and D. Li, “Locality Sensitive Hashing Based Deepmatching for Optical Flow Estimation,” *International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE*, pp. 1472–1476, 2017.