A.1 COMPLEXITY ANALYSIS OF CBR CALCULATOR ALGORITHM

The process of context based relevance calculator used to compute the context score of web page in respect to various contextual senses of words presented in this chapter is already been discussed in Chapter-4. The algorithm to compute the context score is as follows:

A.1.2 Algorithm: CBR Calculator

1. For \( i = 1 \) to no. of web pages (WP)

2. Extract the keywords and their frequency in various HTML tags say list ListWP\(_i\)

3. for \( k = 1 \) to no. of keywords in ListWP

4. Compute \( \text{Wt(Kwp)} = \text{wt of title tag} \times \text{freq(k) in title} + \text{wt of body tag} \times \text{freq(k) in body} + \text{wt of link tag} \times \text{freq(k) in link} + \text{wt of head tag} \times \text{freq(k) in head} \)

5. Total \( \text{Wt (WP)} = \text{Wt(WP)} + \text{Wt(Kwp)} \)

6. end of step 3 loop

7. Extract the top ‘20’ highest weighted words

8. For \( j = 1 \) to 20

9. Extract the contextual senses of word ‘j’

10. For \( l = 1 \) to number of CSs say (q)

11. Extract the number of distinct words in Wordforms and CS definition say (p)

12. Compute the context score as \( \text{CScore(CS/WP)} \)

13. End of step 10 loop

14. Extract the maximum computed CScore for each word ‘j’

15. End of step 8 loop
16. End of step 1 loop

A.1.2 Illustration

Let Page A consist the list of keywords extracted after applying stemming and stop words removal. Then each word is passed to contextual sense extractor that extracts the various Wordforms and Contextual senses (CSs) of the word from WordNet dictionary. Let the Wordform and CS definition of keyword in Page A is a list of distinct keywords present in Wordforms and CSs. Using step 12, context score is computed as the probability of occurrence of CS in web page A as shown in figure A.1

Figure A.1: Web pages and CS association
Thus by computing the probability of keyword present in CS definition to be occurring in web page; lead to compute the probability of occurring CS in web page. In other words it can be said that probability that a page serve the CS is computed.

Assuming,

\[ m = \text{number of keywords in a web page,} \]
\[ n = \text{number of contextual senses of each keyword and} \]
\[ p \text{ is the number of common keywords then,} \]

**Complexity of algorithm is:** \[ m \times n + p \]

**A.2 COMPLEXITY ANALYSIS OF BACK-LINK EXTRACTION ALGORITHM**

The process of back-link extraction is discussed in Chapter-6. The proposed algorithm to extract the number of back-links from the repository of downloaded documents is as follows:

**A.2.1 Algorithm: Extraction of Back-Links**

1. For each URL ‘i’ in the database, URL table is searched to find a match with hyperlink say ‘j’.

2. If a match is found the corresponding (i, j) row is selected where ‘i’ will be the back-link of ‘j’.

3. From the ordered pair (i, j), the value ‘j’ is searched recursively in the hyperlink field till results a match

4. All the ordered pair (i, j) entries of URL and corresponding hyperlinks thus obtained are stored and then passed to crawl worker to download if it has not been downloaded earlier

5. Go to step 1

**A.2.2 Illustration**

Let A1, A2, A3 etc. are the hyperlinks present in a Page A and B1, B2, B3, C etc. are the hyperlinks present in Page B.
Similarly let Page A1 has hyperlinks A1.1, A1.2, A1.3 and B so on as shown in Figure A.2.

![Hyperlinked Structure Diagram]

**Figure A.2: Hyperlinked Structure**

Then the hyperlinked relation shown in figure A.2 is stored in tabular form as shown in Table A.1

**Table A.1: Hyperlinked Structure**

<table>
<thead>
<tr>
<th>Page (URL )</th>
<th>Hyperlink</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A1</td>
</tr>
<tr>
<td></td>
<td>A2</td>
</tr>
<tr>
<td></td>
<td>A3</td>
</tr>
<tr>
<td></td>
<td>A4</td>
</tr>
<tr>
<td></td>
<td>A5</td>
</tr>
<tr>
<td>A1</td>
<td>A1.1</td>
</tr>
</tbody>
</table>
Assuming the back-link extraction algorithm is called to extract the back-links of URL ‘C’. Then, using the structure shown in table A.1 it will returns

\[ C \rightarrow B \rightarrow A1 \rightarrow A \ldots . \]

That shows URLs ‘B’, ‘A1’ and ‘A’ are the back-links of URL ‘C’ found in the local database

**A.2.3 COMPLEXITY ANALYSIS**

Let us assume for an instance if there are ‘m’ number of different web pages downloaded in local database and say ‘n’ different hyperlinks present in them. The total search space to get the back-links of a web page will be as follows:

\[ \text{Total Search Space} = m \times n \]

For a particular web page to gets its back-links recursively search is carried out and assumed that for a particular web page search is converged in say ‘k’ iterations. Then,

\[ \text{The Total complexity is:} \; m \times n \times k \]
A.3 COMPLEXITY ANALYSIS OF BACK-LINK RELEVANCE EVALUATION

The back-link relevance evaluation is done as explained in Chapter-6 by computing the probability of back-link to be similar to the page to which it point to. It computes the relevance score for each back-link for a web page. The algorithm to analyse the relevance of a back-link is as follows:

Algorithm: Relevance Evaluation of Back-links

1. For i = 1 to no. of web pages (WB)
2. Extract the keywords and their frequency in various HTML tags say list ListWBi
3. for k = 1 to no. of keywords in ListWB
4. Compute Wt(Kwb) = wt of title tag * freq (k) in title + wt of body tag * freq(k) in body + wt of link tag * freq(k) in link + wt of head tag * freq(k) in head
5. Total Wt (WB)= Wt(WB) + Wt(Kwb)
6. end of step 3 loop
7. extract the back-links of web page (WB) ‘i’
8. for j = 1 to no. of back-links (BL)
9. Extract the keywords and their frequency in various HTML tags say list ListBL
10. for k = 1 to no. of keywords in ListBLj
11. Compute Wt(Kbl) = wt of title tag * freq (k) in title + wt of body tag * freq(k) in body + wt of link tag * freq(k) in link + wt of head tag * freq(k) in head
12. end of step 10 loop
13. compare the listWBi and listBLj, get the row count for the same say ‘m’
14. For L= 1 to no of matched rows/or records say ‘m’
15. extract the weight of ListBLi[j] from listWB
16. wt BLi=wt BLi + wt of keywords ‘j’ in list WB
17. end of step 14 loop
18. compute RS(BLi) = Wt BLi / Wt WB (computed in step 5)
19. end of step 8 loop
20. end of step 1 loop
A.3.1 Illustration

Let page A is parsed to have list of keywords after applying stemming and stop words removal. Let the pages A1, A2, A3 are the back-link to page A, i.e. they point to Page A. To compute the similarity probability of Page A1 with page A, we can decompose it to keyword level. The joint probability of keywords in page A1 to be occurred in Page A, gives the probability of page A1 to be similar to page A. The Figure A.3 shows the relation between page and its back-links.

![Diagram showing the relation between back-link page with the page it point to](image)

Based on the Navie Bayes Calculations, assuming page ‘A’ consist of keywords K1, K2 and K3 and back-link page A1 consist of K2. Similarly if back-link page A2 consist of K3 and so on. Then, probability of A1 to be occur in page A can be given as equation (1)

$$P(A1 / A) = \frac{P(A / A1)P(A1)}{P(A)} \quad \ldots (1)$$

That further can be represented as equation (2)

$$i.e. = P(A / A1)P(A1) \quad \ldots (2)$$
As Page A1 is having keyword K2 the equation (2) can be represented as equation (3)

\[ i.e. = P(K2/A1)P(A1) \]  (3)

On similar basis the probability of occurrence of back-link Page A2 to be similar to Page A can be represented as

\[ P(A2/A) = P(K3/A2)P(A2) \]

Thus, assuming \( m \) = number of web pages
\( n \) = number of back-link pages and
\( p \) = the number of common keywords

**Complexity of the algorithm is:** \( m \times n + p \)
B.1 COMPARISON OF PROPOSED CONTEXT BASED RANKING WITH PAGE RANK

The ranking order of proposed context based ranking algorithm is compared with the ordering done by page rank score. For this purpose, the list of URLs is downloaded from Google on various topics (query keyword) and is analysed using Context based calculator mechanism. Based on the computed context score the documents are selected in a particular contextual sense of query keyword. The selected URLs are then ordered using context score. The page rank score is also computed for the same list of URLs. The URLs are again ordered using Page rank score. The two orders of list of URLs are compared to see which heuristics (score) placed the contextually more related URLs at top positions.

In this appendix the comparison graph for topics ‘Mouse’, ‘Lion’, ‘Java’, ‘Crane’ and ‘Colt’ are shown in subsequent sections. For comparison purpose the colour code to represent the URL is kept same. Various URLs are represented at x-axis and their context score is represented at y-axis. The height of the bars shows the relatedness of that URL to the selected sense.
B.1.1 Comparison of URLs on topic ‘Mouse’ in sense ‘Computer Mouse’

![Graph showing Context score based ranking of URLs on topic “Computer Mouse”](image1)

**Figure B.1 (a):** Context score based ranking of URLs on topic ‘Computer Mouse’

![Graph showing Page Rank based ranking of URLs on topic “Computer Mouse”](image2)

**Figure B.1 (b):** Page Rank based ranking of URLs on topic ‘Computer Mouse’
B1.2 Comparison of URLs on topic ‘Mouse’ in sense ‘Mouse Rodent’

Figure B.2 (a): Context score based ranking of URLs on topic ‘Mouse Rodent’

Figure B.2 (b): Page Rank based ranking of URLs on topic ‘Mouse Rodent’
B 1.3 Comparison of URLs on topic ‘Crane’ in sense ‘Crane Bird’

Figure B.3 (a): Context score based ranking of URLs on topic ‘Crane Bird’

Figure B.3 (b): Page Rank based ranking of URLs on topic ‘Crane Bird’
B 1.4 Comparison of URLs on topic ‘Crane’ in sense ‘Crane Machine’

Figure B.4 (a): Context score based ranking of URLs on topic ‘Crane Machine’

Figure B.4 (b): Page Rank based ranking of URLs on topic ‘Crane Machine’
B 1.5 Comparison of URLs on topic ‘Java’ in sense ‘Java Programming Language’

Figure B.5 (a): Context score based ranking of URLs on topic ‘Java Lang.’

Figure B.5 (b): Page Rank based ranking of URLs on topic ‘Java Lang.’
B.1.6 Comparison of URLs on topic ‘Java’ in sense ‘Java Island’

Figure B.6 (a): Context score based ranking of URLs on topic ‘Java Island’

Figure B.6 (b): Page Rank based ranking of URLs on topic ‘Java Island’
B 1.7 Comparison of URLs on topic ‘Java’ in sense ‘Java Coffee’

Figure B.7 (a): Context score based ranking of URLs on topic ‘Java Coffee’

Figure B.7 (b): Page Rank based ranking of URLs on topic ‘Java Coffee’
Figure B.8 (a): Context score based ranking of URLs on topic ‘Lion Animal’

Figure B.8 (b): Page Rank based ranking of URLs on topic ‘Lion Animal’
B.2 COMPARISON OF PROPOSED CONTEXT BASED RANKING WITH Google's RANKING

The comparison of ranking by proposed context based ranking algorithm is done with the ranking by Google search engine. For this purpose, the list of URLs is downloaded from Google on various topics (query keyword). The list of URLs is analysed in different senses. For example, URLs are analysed on topic ‘Java’ for 3 different queries; java language, java island and java coffee. The list of URLs is stored and the position at which these links are displayed is also noted as the position suggested by Google. The URLs in the list are analysed using Context based calculator mechanism. Based on the computed context score the documents are re-ordered and their position is decided in the descending order of their context score. The two orders of list of URLs are compared to see which heuristics (score) placed the contextually more related URLs at top positions. Top results decided by Context score are selected and for the same URLs their display order in Google results are shown in corresponding second graph. The results show that the context score based ordering placed the contextually more related documents at top positions.

In this appendix the comparison graph for topics ‘Java Language’, ‘Java Island’, ‘Crane Machine’, ‘Crane Bird’, ‘Lion Animal’ and ‘Colt Young Horse’ are shown in subsequent sections. For comparison purpose the colour code to represent the URL is kept same. Various URLs are represented at x-axis and their position in result list as decided by Google and Context score is represented at y-axis.

Illustration of Results in subsequent graphs:

The order of URLs by context based ranking placed the contextually highly related web document at top position i.e. at first. This implies the order in context based graph shows the position of URLs in descending order of their computed context score thus first position represent the URL i.e. highly related and last position in graph shows that the corresponding URL is less relevant to that topic.

The comparison is done on the basis of the change in position number i.e. the height of bars.
B.2.1 comparison of URLs positions on topic ‘Java Prog. Lang.’

Figure B.9 (a): Context score based ranking of URLs on topic ‘Java Lang.’

Figure B.9 (b): Google’s ranking of URLs on topic ‘Java Lang.’
B 2.2 comparison of URLs on topic ‘Java Island’

Figure B.10 (a): Context score based ranking of URLs on topic ‘Java Island’

Figure B.10 (b): Google’s ranking of URLs on topic ‘Java Island’
B2.3 Comparison of URLs on topic ‘Java Coffee’

Figure B.11 (a): Context score based ranking of URLs on topic ‘Java Coffee’

Figure B.11 (b): Google’s ranking of URLs on topic ‘Java Coffee’
B 2.4 Comparison of URLs on topic ‘Crane Bird’

Figure B.12 (a): Context score based ranking of URLs on topic ‘Crane Bird’

Figure B.12 (b): Google’s ranking of URLs on topic ‘Crane Bird’
B 2.5 Comparison of URLs on topic ‘Crane Machine’

Figure B.13 (a): Context score based ranking of URLs on topic ‘Crane Machine’

Figure B.13 (b): Google’s ranking of URLs on topic ‘Crane Machine’
B 2.6 Comparison of URLs on topic ‘Lion Animal’

Figure B.14 (a): Context score based ranking of URLs on topic ‘Lion Animal’

Figure B.14 (b): Google’s ranking of URLs on topic ‘Lion Animal’
B 2.7 Comparison of URLs on topic ‘Colt Horse’

Figure B.15 (a): Context score based ranking of URLs on topic ‘Colt Young Horse’

Figure B.15 (b): Google’s ranking of URLs on topic ‘Colt Young Horse’
LIST OF AUTHOR’S PUBLICATIONS


