CS-834 Intro to Information Retrieval
First Presentation

Seminal and Cited Papers

Plinio H. Vargas

September 15, 2016

Old Dominion University

Table of Contents

Introduction

Pre-Google WWW

In the beginning there was the World Wide Web; and the traffic of knowledge kept increasing, so the number of irrelevant documents recall. Then Google was born with the pure notion of using PageRank to bring order to the **Web**.

Google PageRanking

PageRank Calculation

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right) [1]$$

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) [3]$$

PageRank Calculation Cont.

Internet consisting of only 3 pages.

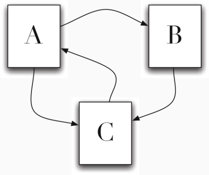


Figure 1: Three Web-pages

Since we do not know any of the pages ranking, we will assume that:

$$PR(A) = PR(B) = PR(C) = \frac{1}{3} \approx 0.33$$

Graph and example taken from textbook[4] chapter 4

PageRank Calculation Cont.

First iteration:

$$PR(C) = \frac{PR(A)}{2} + \frac{PR(B)}{1} = \frac{0.33}{2} + \frac{0.33}{1} = 0.5$$
$$PR(A) = \frac{PR(C)}{1} = \frac{0.33}{1} \approx 0.33$$
$$PR(B) = \frac{PR(A)}{2} = \frac{0.33}{2} \approx 0.17$$

Second iteration:

$$PR(C) = \frac{PR(A)}{2} + \frac{PR(B)}{1} = \frac{0.33}{2} + \frac{0.17}{1} \approx 0.33$$
$$PR(A) = \frac{PR(C)}{1} = \frac{0.5}{1} = 0.5$$
$$PR(B) = \frac{PR(A)}{2} = \frac{0.33}{2} \approx 0.17$$

Graph and example taken from textbook[4] chapter 4

PageRank Calculation Cont.

Third iteration:

$$PR(C) = \frac{PR(A)}{2} + \frac{PR(B)}{1} = \frac{0.5}{2} + \frac{0.17}{1} \approx 0.42$$
$$PR(A) = \frac{PR(C)}{1} = \frac{0.33}{1} \approx 0.33$$
$$PR(B) = \frac{PR(A)}{2} = \frac{0.5}{2} = 0.25$$

After few more iterations:

$$PR(C) = \frac{PR(A)}{2} + \frac{PR(B)}{1} \approx 0.4$$
$$PR(A) = \frac{PR(C)}{1} \approx 0.4$$
$$PR(B) = \frac{PR(A)}{2} \approx 0.2$$

See Table 1 for full iteration calculation.

TextRank

TextRank

A graph-based ranking algorithm of natural language texts with the purpose of:

- Keyword Extraction

TextRank

A graph-based ranking algorithm of natural language texts with the purpose of:

- Keyword Extraction
- Sentence Extraction

TextRank Equation

TextRank modified Google PageRank "random surfer model" equation:

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \tag{1}$$

Taking into account edge weights to compute the score associated with a vertex in the graph:

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \tag{2}$$

TextRank Work Comparison

Although TextRank work is based on equation (2) taken from Google PageRank equation (1), its research innovation is in great deal related to A. Hulth [2] “Improved automatic keyword extraction given more linguistic knowledge”.

TextRank vs Hulth-2003

Comparison between TextRank and Hulth-2003 algorithms:

TextRank

Application

Keyword Extraction
Sentence Extraction

Approach

Unsupervised

Example

Block content.

Hulth, 2003

Application

Keyword Extraction
Sentence Extraction

Approach

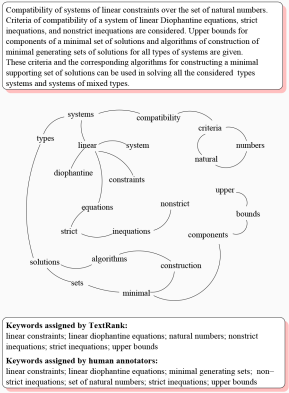
Supervised

Example

Block content.

TextRank Keyword Extraction Example

Figure 2: Keywords Extraction Graph Example



Sample graph built for keyphrase extraction from an Inspec abstract.

Source: [3]

TextRank Steps

1. Text is tokenized

TextRank Steps

1. Text is tokenized
2. Edge is added between lexical units

TextRank Steps

1. Text is tokenized
2. Edge is added between lexical units
3. Each vertex is set to initial value of 1

TextRank Steps

1. Text is tokenized
2. Edge is added between lexical units
3. Each vertex is set to initial value of 1
4. TextRank algorithm runs until it converges

Conclusion

Summary

Working on it.

Questions?

Table 1: PageRank Iteration Calculation for Figure 1

Iteration	A	B	C	A-Error	B-Error	C-Error
0	0.333333	0.333333	0.333333	-	-	-
1	0.333333	0.500000	0.166667	0.0000	0.1667	0.1667
2	0.500000	0.333333	0.166667	0.1667	0.1667	0.0000
3	0.333333	0.416667	0.250000	0.1667	0.0833	0.0833
4	0.416667	0.416667	0.166667	0.0833	0.0000	0.0833
5	0.416667	0.375000	0.208333	0.0000	0.0417	0.0417
6	0.375000	0.416667	0.208333	0.0417	0.0417	0.0000
7	0.416667	0.395833	0.187500	0.0417	0.0208	0.0208
8	0.395833	0.395833	0.208333	0.0208	0.0000	0.0208
9	0.395833	0.406250	0.197917	0.0000	0.0104	0.0104
10	0.406250	0.395833	0.197917	0.0104	0.0104	0.0000
11	0.395833	0.401042	0.203125	0.0104	0.0052	0.0052
12	0.401042	0.401042	0.197917	0.0052	0.0000	0.0052
13	0.401042	0.398438	0.200521	0.0000	0.0026	0.0026
14	0.398438	0.401042	0.200521	0.0026	0.0026	0.0000
15	0.401042	0.399740	0.199219	0.0026	0.0013	0.0013
16	0.399740	0.399740	0.200521	0.0013	0.0000	0.0013
17	0.399740	0.400391	0.199870	0.0000	0.0007	0.0007
18	0.400391	0.399740	0.199870	0.0007	0.0007	0.0000
19	0.399740	0.400065	0.200195	0.0007	0.0003	0.0003
20	0.400065	0.400065	0.199870	0.0003	0.0000	0.0003
21	0.400065	0.399902	0.200033	0.0000	0.0002	0.0002
22	0.399902	0.400065	0.200033	0.0002	0.0002	0.0000
23	0.400065	0.399984	0.199951	0.0002	0.0001	0.0001
24	0.399984	0.399984	0.200033	0.0001	0.0000	0.0001

Final result is shown on page 6

Keyword Extraction Results

Table 2: Results for automatic keyword extraction using TextRank or supervised learning (Hulth, 2003)

	Assigned		Correct				
Method	Total	Mean	Total	Mean	Precision	Recall	F-measure
TextRank							
Undirected, Co-occ.window=2	6,784	13.7	2,116	4.2	***31.2***	43.1	***36.2***
Undirected, Co-occ.window=3	6,715	13.4	1,897	3.8	28.2	38.6	32.6
Undirected, Co-occ.window=5	6,558	13.1	1,851	3.7	28.2	37.7	32.2
Undirected, Co-occ.window=10	6,570	13.1	1,846	3.7	28.1	37.6	32.2
Directed, forward, Co-occ.window=2	6,662	13.3	2,081	4.1	31.2	42.3	35.9
Directed, backward, Co-occ.window=2	6,636	13.3	2,082	4.1	31.2	42.3	35.9
Hulth (2003)							
Ngram with tag	7,815	15.6	1,973	3.9	25.2	***51.7***	33.9
NP-chunks with tag	4,788	9.6	1,421	2.8	29.7	37.2	33.0
Pattern with tag	7,012	14.0	1,523	3.1	21.7	39.9	28.1

References I

- S. Brin and L. Page, *The anatomy of a large-scale hypertextual web search engine*, Computer Networks and ISDN Systems, 30 (1998), pp. 107––117.
- A. Hulth, *Improved automatic keyword extraction given more linguistic knowledge*, in Proceedings of the 2003 conference on Empirical methods in natural language processing, Association for Computational Linguistics, 2003, pp. 216–223.
- R. Mihalcea and P. Tarau, *Textrank: Bringing order into texts*, Barcelona, Spain, 2004, Association for Computational Linguistics, pp. 404–401. http://www.aclweb.org/anthology/W04-3252.
- T. S. W.B. Croft, D. Metzler, *Search Engine Information Retrieval in Practice*, Pearson Education, 2015.