ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 10, October 2014

Enhancing a Ranking System: Practical Applications to Information Retrieval Systems and BCS Rankings

Kee-cheol Lee

Computer Engineering Department, Hongik University, Seoul, Korea 121-791

Abstract: The rankings of the members fora given domain, if provided, can be thought to be the efficient description of the domain and help us make decisions accordingly. For example, each web search engine like Google retrieves query-relevant documents and shows us highly ranked ones in ascending order of their ranks. However, if we have two or more rankers or ranking systems with their own expertise for one domain, we may be puzzled what to choose or how to fix their differences. In this paper, the issue of how to combine two experts' rankings is treated. We suggest some methods of combining two different rankings. For practical applications, two domains were selected to test and validate our method. First, two different rankings generated by changing some portion of our information retrieval system were selected and our experiments show that the resulting average rank of top ten relevant documents, for example, was considerably improved. The second domain we tried may be thought to be almost improbable. The question is if American college BCS football ranking, officially published from the middle of the football season, can be modified to better predict the four week later BCS ranking itself. Our experiments show that some computer-based ranker help enhance the predictability of the future BCS ranking.

Keywords: ranking, machine learning, information retrieval, search engines, TF-IDF, NCAA football, computer rankings, BCS ranking

I. INTRODUCTION

Ranking is a simplified decision-making method of a given system such that it makes its users easily understand the current situation and behave accordingly. Rank-related research has been conducted in bio-informatics areas, where rank normalization has been applied to replace each observation by its fractional rank(the one divided by the total number of genes) within array[1][2]. With this rank normalization robustness to non-additive noise is achieved at the expense of losing some parametric information of expressions[3]. In information retrieval areas, rank normalization like rank shifting and rank freezing has been studied for relevance feedback of search engines[4], which must show us a small number of highly ranked relevant web pages in the order of relevance, despite the enormous amount of web pages relevant to given queries. The problem is that we may have two or more rankers that decide ranking in different ways, and each of which may have its own expertise. Given two or more experts' opinions of the rankings for the same domain, the question arises if and how a different ranking should be referenced. Recently in economics area, the issue of simultaneous consultation and utilizing two informed experts' opinions for decision making has been studied[5][6][7], but we believe that this issue should be processed mathematically without any human bias to be usable in broader problems like search engines and sports ranking predictions. To reference and utilize another expert's ranking, a combining function should be defined and tested to convert original rankings. If we have an enough past ranking sequence, the appropriate combining functions may be applied such that they better predict future unknown rankings. Suspecting that just averaging two rankings may lose the expertise of each expert ranker, we decided to give another ranker a supporting status to improve the main system ranker. We suggest some combining functions and conditions upon which those functions may be applied, and applied our method to two quite different areas, i.e., re-ranking relevant documents retrieved by search engines, and improving the BCS rankings, the prestigious official American college football rankings, in terms of predicting future BCS rankings better website.

II. COMBINING RANKINS

Just averaging two rankings may possibly blur the expertise hidden in each published ranking system. Therefore to keep the original expertise, combining functions should be used appropriately. We suggest the following basic method.

Let *set1* be the set of elements ranked by *ranker1*, main ranker;

Let *set2* be the set of elements ranked by *ranker2*, auxiliary ranker;

Let *default_rank1*(*i*) be the rank assignable to any element *i* not belonging to *set1*;

Let *rank1*(*i*) be the ranker1's rank of element *i* and *rank2*(*i*) be the ranker2's rank of element *i*;
for each $i \in (set1 \cup set2)$

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 10, October 2014

```
{
if (i∉set1)
rank1(i) = default_rank1(i);
if (i∉set2)
   tmp_rank1(i) = rank1(i);
else
   tmp_rank1(i) = CombineRanks(rank1(i), rank2(i));
} // end of for
```

Sort and re-rank all elements in (*set1*∪*set2*) in ascending order of (*tmp_rank1*(*i*), *rank1*(*i*));
Let *new_rank1*(*i*) be the modified new rank of an element *i*, after the above steps; Here, *default_rank1*(*i*) is the rank to be assigned to any element not belonging to *set1*, and its value should be (pessimistically) big enough not to over-affect final rankings. For example, in case of top 25 ranking only predictions, a value much larger than 25, e.g. 35, was used for the experiment. If unranked ones were considered as 26^{th} as in many prediction systems, it could over-affect final rankings. The final sorting is done first by *tmp_rank1*(*i*), and *rank1*(*i*) is used as a tie-breaker. The combining function *Combine Ranks*(*r1*, *r2*) can be regarded to be a kind of an averaging function, and some of the potential candidate functions which may be used are listed below.

```
// arithmetic mean
doubleAri(double r1, double r2)
{ return (r1 + r2)/2.0; }
// average of squared arithmetic mean
doubleAri2(double r1, double r2)
{ returnsqrt((r1*r1 + r2*r2)/2.0); }

// harmonic mean
doubleHar(double r1, double r2)
{ return 2.0/(1.0/r1 + 1.0/r2); }

// average of squared harmonic mean
doubleHar2(double r1, double r2)
{ returnsqrt(2.0/(1.0/(r1*r1)+ 1.0//(r2*r2)); }
```

Har function can be thought to consider the ranks of the ranker2 more than *Ari* function, and *Ari2* and *Har2* are the second order version of *Ari* and *Har*. Therefore, the return values of the four combining methods are in the order of *Har2*, *Har*, *Ari*, and *Ari2*. The return values of the above mentioned combining functions are summarized in TABLE I(a) for some example data set. TABLE I(b) and (c) show the corresponding results of the temporary ranks and the final ranks for the same data set.

TABLE I
RANK COMBINING EXAMPLE

(a) Combined ranks, if applicable

rank1	**rank2**	**CombineRanks**			
		Ari	**Ari2**	**Har**	**Har2**
1	2	NA	NA	NA	NA
2	1	1.5	1.58	1.33	1.26
3	7	NA	NA	NA	NA
4	6	NA	NA	NA	NA
5	5	NA	NA	NA	NA
6	4	5	5.10	4.8	4.71
7	3	5	5.39	4.2	3.90
8	10	NA	NA	NA	NA
9	9	NA	NA	NA	NA
10	8	9	9.06	8.89	8.83

(b) Temporary ranks

rank1	***rank2***	***tmp_rank1***			
		Ari	***Ari2***	***Har***	***Har2***
1	2	1	1	1	1
2	1	1.5	1.58	1.33	1.26
3	7	3	3	3	3
4	6	4	4	4	4
5	5	5	5	5	5
6	4	5	5.10	4.8	4.71
7	3	5	5.39	4.2	3.90
8	10	8	8	8	8
9	9	9	9	9	9
10	8	9	9.06	8.89	8.83

(c) Modified new ranks

rank1	rank2	new_rank1, if these are all data available			
		Ari	Ari2	Har	Har2
1	2	1	1	1	1
2	1	2	2	2	2
3	7	3	3	3	3
4	6	4	4	4	5
5	5	5	5	7	7
6	4	6	6	6	6
7	3	7	7	5	4
8	10	8	8	8	8
9	9	9	9	10	10
10	8	10	10	9	9

III. EHANCING SEARCH ENGINE RANKINGS

We have constructed some search engine system to be used for information retrieval-related class assignments [8][9][10]. Its rankings are based on TF-IDF method, where by TF we mean some measure of a term frequency for a document, and by IDF(inverse document frequency) we mean a measure of a word in the collection, including entropy or noise measure. We are not going to delve into the details of our information retrieval system[11]. For this experiment, TF variations like the following *TF1* and *TF2* were used.

$$TF1_{ij} = \log_2(freq_{ij}+1)/\log_2 t_j \quad (1)$$
$$TF2_{ij} = tf_{ij} \quad (2)$$
$$tf_{ij} = freq_{ij} / maxfreq_j$$

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 10, October 2014

$freq_{ij}$= frequency of term i in document j
$maxfreq_j$= max.frequency of any term in document j
t_j= the number of unique terms in document j

Table II summarizes CACM test collection used for this experiment. It contains 3,204 documents and 52 queries. The experimental results for two rankers which utilize *TF1* and *TF2* are shown in the 2nd and 3rd column of TABLE III. *TF1*, suggested by *Harman*[12], turned out to be better than *TF2*, a simple relative term frequency, in terms of all seen relevant documents and top 10 documents. The question is if and how the search engine performance based on *TF1* can be improved by referencing an inferior ranker(*TF2*-based ranker). The last two columns of TABLE III show us that by combining the ranks of the ranker2 to those of the ranker1 by the arithmetic averaging function(*Ari*) and the harmonic averaging function(*Har*) defined before, more than 5% of improvement has been obtained. This means that referencing and methodically utilizing the ranks of the ranker2, even if they are not overall satisfactory, could possibly improve the ranking quality of the *ranker1*.

TABLE II
INFORMATIONRETRIEVAL TEST COLLECTION AND GENERAL RESULT SUMMARY (3204 CACMDOCUMENTS, 52 QUERIES)

	avg.±σ	median	min	max
terms/doc.	63.0±51.5	38	13	356
uniq.terms/doc.	44.7±32.0	29	10	241
terms/query	19.9±14.7	14	2	68
uniq.terms/query	16.4±11.1	12	2	45
rel.docs/query	11.9±8.7	10	1	38
recall	0.84±0.19	0.89	0.35	1.0
returned docs	524.2±271.3	454.5	21	1109

TABLE III
INFORMATION RETRIEVAL EXPERIMENTAL RESULTS

TF for ranker1		*TF1*	*TF2*	*TF1*	*TF1*
TF for ranker2		-	-	*TF2*	*TF2*
Combining function		-	-	*Ari*	*Har*
all seen rel.docs	avg.rank	73.09	76.79	69.43	68.83
	imp. rate	0	-5.1%	+5.0%	+5.8%
top 10 documents	avg.rank	53.1	55.0	50.4	49.6
	imp. rate	0	-3.6%	+5.1%	+6.6%

IV. IMPROVING NCAA FOOTBALL BCS RANKINGS

The next domain for our experiments is the BCS which is considered to be the most prestigious ranking system for evaluating American college football teams. In 2013 season, for example, BCS rankings were officially published for eight weeks from week 8 to week 15 after the football season began. The BCS ranking is generated based on many factors, which are outside of the scope of this paper. The question here is if we can enhance the current week BCS ranking. One of the difficulties of this domain is that no real ranking exists, so we decided to use the predictability of the future (four week later) BCS ranking to determine the validity of our modified BCS ranking. BCS and other rankings are available on-line[13][14][15], and we summarized BCS ranking together with 6 computer-based rankings and computer-averaged ranking for 8th to 15th football season weeks of the year 2013. TABLE IV is a sample ranking data for 8th football week. The max rank provided for BCS week 8 ranking is 42, but we have just top 25 of 6 computer rankings, and top 27 of the computer average ranking.

TABLE IV
BCS AND COMPUTER NCAAFOOTBALL RANKINGS FOR WEEK 8, 2013

wk8 (Oct. 20, 2013)								
Team	BCS	AH	CM	JS	KM	PW	RB	Cp.Avg
Alabama	1	2	3	2	3	1	1	2
Florida State	2	1	2	1	1	2	5	1
Oregon	3	4	4	4	4	4	2	4
Ohio State	4	5	5	8	8	7	3	5
Missouri	5	3	1	3	2	3	6	3
Stanford	6	6	6	15	6	10	4	6
Miami (Fla.)	7	8	12	12	9	8	21	10
Baylor	8	9	11	14	15	13	11	12
Clemson	9	10	8	13	10	9	7	9
Texas Tech	10	11	10	10	12	11	14	11
Auburn	11	7	7	9	5	6	17	7
UCLA	12	15	19	11	13	12	16	14
LSU	13	14	17	19	11	16	9	15
Virginia Tech	14	13	9	7	7	5		8
Oklahoma	15	12	15	20	19	19	8	16
Texas A&M	16	22	22	22	16	17	18	18
Fresno State	17	16	14	16		14		17
Northern Illinois	18	19	13	5	14	15	10	13
Oklahoma State	19	25						
Louisville	20						15	
South Carolina	21	24			22	23	19	26
Michigan	22	17	16			25	20	21
Central Florida	23	23	23	17		18	13	19
Nebraska	24							
Oregon State	25		24	6	18	21	22	20
Wisconsin	26							
Michigan State	27		21	21	25	24		27
Arizona State	28	18	18		21			25

Georgia	29	20	25	23	20	22	25	22
Notre Dame	30	21	20	25	24		12	22
Ole Miss	31			18	17	20		22
Florida	32				23			
Texas	33						24	
Houston	34							
Ball State	35							
BYU	36						23	
Boise State	37							
Washington	38							
La.-Lafayette	39							
Rutgers	39							
Tennessee	39							
Pittsburgh	42			24				

TABLE V shows us the predictability of 4 week later BCS ranking in terms of average rank errors. As expected, any week *n* computer-based rank predicts week *n*+4 BCS ranking better than week *n* BCS ranking.

The reason might be that each system has a different kind of expertise and that for computer rankings just top 25 are used for prediction, and the rest universities not ranked were assigned very pessimistic rank(*i.e.*35) for experiments.

The fact that the 4 week later BCS predictability of computer average ranking is comparatively better than those of other computer rankings just reflects that the average computer ranking itself is actually one important factor in deciding the BCS ranking.

TABLE V
FOUR WEEK LATER BCSRANKING PREDICTION PERFORMANCE OF THE CURRENT BCS AND VARIOUS COMPUTER RANKINGS (ORIGINAL PERFERMANCE):AVERAGE RANK ERROR

ranker1(*)	**BCS**	**AH**	**CM**	**JS**	**KM**	**PW**	**RB**	**CAvg**
***8→BCS12**	5.84	6.8	7.08	7.92	7.92	6.32	8.92	6.44
***9→BCS13**	5.76	7.12	7.68	7.04	7.56	6.72	8.44	6.64
***10→BCS14**	5	5.6	6.04	5.96	5.76	6.04	6.4	4.76
***11→BCS15**	4.2	4.84	5.56	5.92	6.12	5.88	4.76	5.32
dev. avg.	5.20	6.09	6.59	6.71	6.84	6.24	7.13	5.79
dev. σ	0.66	0.92	0.84	0.83	0.92	0.32	1.66	0.78
enh.%	0	-17.1	-26.7	-29.0	-31.5	-20.0	-37.1	-11.3

TABLE VI summarizes the 4 week predictability of the modified BCS ranking in terms of rank errors by utilizing each computer ranker as an auxiliary expert to modify the current week BCS ranking. This experiment has been conducted for the data for eight weeks (i.e., week 8 to week 15) of the 2013 football season. Surprisingly,

TABLE VI
FOUR WEEK LATER BCSRANKING PREDICTION PERFORMANCE OF THE CURRENT BCSRANKING MODIFIED BY REFERENCING COMPUTER RANKINGS USING *ARI*COMBINING FUNCTION:AVERAGE RANK ERROR

BCS'	ranker1	BCS						
	ranker2	-	AH	CM	JS	KM	PW	RB	CAvg
	Comb. fcn.	-	Ari	Ari	Ari	Ari	Ari	Ari	Ari
BCS'8→BCS12		5.84	5.88	5.72	5.88	5.84	5.72	6.16	5.88
BCS'9→BCS13		5.76	5.8	5.64	5.88	5.6	5.8	5.88	5.88
BCS'10→BCS14		5	5.08	5	5.12	4.92	4.96	5.16	4.92
BCS'11→BCS15		4.2	4.2	4.12	4.4	4.04	4.16	4.28	4.2
dev. avg.		5.20	5.24	5.12	5.32	5.10	5.16	5.37	5.22
dev. σ		0.66	0.68	0.64	0.62	0.70	0.66	0.73	0.71
enh.%		0	-0.77	1.54	-2.31	1.92	0.77	-3.27	-0.38

utilizing CM and KM computer rankers improves the predictability by 1.54% and 1.92%, respectively. This improvement is amazing because it is hard to assume that any other ranking than current BCS can better predict 4 week later BCS ranking itself, considering that the method of calculating current BCS remains the same as that of calculating other week BCS. The rationale is that if we modify the current BCS ranking closer to the true ranking, it may be eventually reflected in the future. BCS utilizes the average of all computer rankings as a factor in deciding its ranking, but our method may suggest how and which computer ranking should be considered for ranking better.

V. CONCLUSION

If the member rankings are provided for a given domain, they can be efficiently used for their users' decision making processes. However, when we are given different rankings for the same domain, and if each of the ranking producing systems or experts has its own expertise, we are puzzled how to react. In this paper, we suggest a general paradigm for combining conflicting rankings. For that purpose, ranking combining functions were suggested, and two domains were selected for testing and validating our method. First, the issue of handling two different rankings produced by selecting different term-frequency definitions was treated. The test results for that domain is very encouraging, especially in terms of the average rank of top ten relevant documents for given queries. The second domain we chose is the BCS ranking for the American college football. By utilizing computer-based ranking systems, we experimented to seethe possibility of enhancing the predictability of the current BCS ranking. We found some computer ranking may help enhance BCS ranking in terms of predicting four week later BCS ranking itself, which is a very encouraging result.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 10, October 2014

ACKNOWLEDGMENT

This work was supported by 2014 Hongik University Research Fund.

REFERENCES

[1] A. Tsodikov, A. Szabo, and D. Jones, "Adjustment and Measures of Differential Expression for Microarray Data," Bioinformatics 13(2), pp.251-260, 2002.
[2] A. Szabo, K. Boucher, W. Carroll, L. Klebanov, A. Tsodikov, and A. Yakovlev,"Variable Selection and Pattern Recognition with Gene Expression Data Generated by the Microarray Technology," Math Biosci 176, pp.71-98, 2002.
[3] Xing Qiu,Hulin Wu, and Rui Hu, "The Impact of Quantile and Rank Normalization Procedures on the Testing Power of Gene Differential Expression Analysis,"BMC Bioinformatics 14:124 http://www.biomedcentral.com/1471-2105/14/124, pp.1-10, 2013.
[4] XiangyuJin, James French, and Jonathan Michel, "Toward Consistent Evaluation of Relevance Feedback Approached in Multimedia Retrieval," AMR 2005, LNCS 3877, pp.191-206, Springer-Verlag Berlin Heidelberg,2006.
[5] Ming Lee, "Advice from Multiple Experts: A Comparison of Simultaneous, Sequential, and Hierarchical Communication," The B.E. Journal Theoretical Economics, 10(1), Article 18, 2010.
[6] Guangsong Lu and Guochang Li, "Analysis of Prerequisite for Experts Acquiring and Reporting Informatively," ICEE International Conference on E-Business and E-Government, pp. 5094-5079, Guangzhou, China, 7-9 May 2010.
[7] Li Ming and KristofMadarasz, "When Mandatory Disclosure Hurts: Expert Advice and Conflicting Interests," Journal of Economic Theory, pp. 47-74, 2008.
[8] W.B. Croft, D.Metzler, and T. Strohman, "Ranking with Indexes," Search Engines Information Retrieval in Practice, pp.125-186, Pearson Education Inc., 2010.
[9] D. Harman, E. Fox, R. Baeza-Yates, and W. Lee, "Inverted Files," In W. Frakes and R. Baeza-Yates, editors, Information Retrieval: Data Structures & Algorithms, chapter 3, pp.28-43, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.
[10] Christopher D. Manning, PrabhakarRaghavan, and HinrichSchutze, "Computing Scores in a Complete Search System," Introduction to Information Retrieval, chapter 7, pp.124-138, Cambrige University Press, 2008.
[11] K. C. Lee and H. Kim, "Analyzing and Combining TF-IDF based Text Retrieval Systems," GESTS Int. Trans. Computer Science and Engr., Vol.67, No.1, 2012.
[12] D. Harman, "Ranking Algorithms," In W.B. Frakes and R. Baeza-Yates, editors, Information Retrieval: Data Structures & Algorithms, pp.363-392, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.
[13] ESPN football rankings, espn.go.com/college-football/rankings, 2013.
[14] BCS rankings, www.bcsguru.com/bcs_standings.htm, 2013.
[15] Legend Poll, www.legendschannel.com/legends-poll, 2013.

BIOGRAPHY

Kee-cheol Lee He was born in Seoul, Korea in 1955. He received a BS degree in electronic engineering from Seoul National University in 1977, an MS degree in computer science from Korea Advanced Institute of Science in 1979, and a Ph.D in electrical and computer engineering from University of Wisconsin-Madison in 1987. Since 1989, he has been on the faculty of computer engineering department, Hongik University, Seoul, Korea, and currently he is a professor. His academic and research interests cover the fields of artificial intelligence, machine learning, and information retrieval.