Fusion in Information Retrieval

J. Shane Culpepper & Oren Kurland

RMIT University, Australia

Technion, Israel Institute of Technology

July 08th, 2018

Presenters

- Oren Kurland
  - PhD Computer Science from Cornell University, 2006.
  - **Research Interests**: Information Retrieval
  - `kurland@ie.technion.ac.il`
  - `https://iew3.technion.ac.il/~kurland/`
- Shane Culpepper
  - PhD Computer Science from the University of Melbourne, 2008.
  - **Research Interests**: Information Retrieval, Algorithms and Data Structures, Machine Learning
  - `shane.culpepper@rmit.edu.au`
  - `https://culpepper.io`

Overview

1. Intro and Overview
2. Theoretical Foundations
3. Fusion in Practice
4. Learning and Fusion
5. Applications
6. Conclusions and Future Directions

What is fusion?

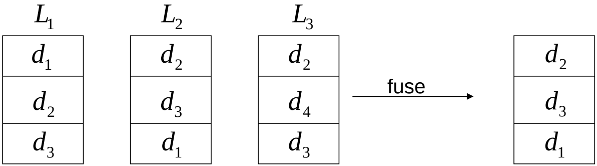
Fusion (IR)

Fusion for Information Retrieval is the the process of combining multiple sources of information so as to produce a single result list in response to a query. This can be accomplished by combining the results from multiple ranking algorithms, different document representations, different representations of the information need, or combinations of all of the above.

Why Should I Care?

- Historically, many of the most competitive systems at evaluation exercises such as TREC, CLEF, FIRE, and NTCIR have been based on fusion.
- There are theoretical and practical connections between fusion and many other fundamental IR techniques, such as pooling in evaluation, ensembles in learning-to-rank, query performance prediction, diversification, and relevance modeling.
- Understanding the fundamentals of fusion models could provide additional tools to help decipher how more complex learned ensembles work. At the very least, it will provide tools to help you build ***better*** learned models.

Basic Notation



L_1	L_2	L_3		
d_1	d_2	d_2		d_2
d_2	d_3	d_4	fuse	d_3
d_3	d_1	d_3		d_1

q: query
d: document
L_i: a document list retrieved in response to q using retrieval method (system) M_i
$r_{L_i}(d)$: d's rank in L_i; the highest ranked document has rank 1
$s_{L_i}(d)$: d's retrieval score in L_i
$F(d; q)$: the fusion score of d

Our Focus: Retrieval over a Single Corpus

We do not cover Federated Search where lists retrieved from different corpora are fused, or on enhancing fusion using external corpora.

1. J. Callan. “Distributed information retrieval”. Advances in information retrieval (edited by B. Croft), chapter 5, pages 127–150.
2. M. Shokouhi and L. Si. “Federated Search”. FNTIR, 5(1), pages 1–102, 2011.

How Does it Work?

- **Skimming effect**: Occurs when systems retrieve different documents. Fusion then just takes the top-*k* documents from each system.
- **Chorus effect**: Occurs when several systems retrieve many of the same documents, so that each document has multiple sources of evidence.
- **Dark Horse effect**: Outlier systems that are unusually good (or bad) at finding unique documents that other systems do not retrieve.

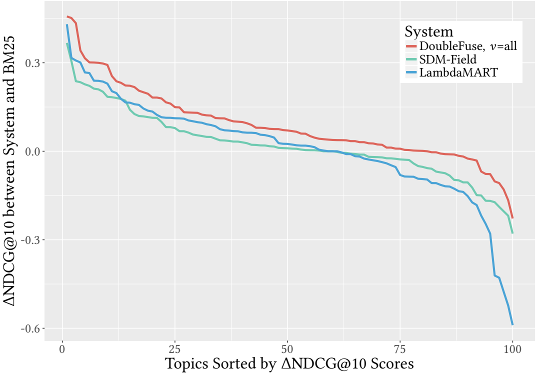
1. C. C. Vogt and G. W. Cottrell: "Fusion via linear combination of scores." Information Retrieval, 1(3) pages 151–173, 1999. (From Diamond T. "Information retrieval using dynamic evidence combination". Unpublished Ph.D. Thesis proposal, School of Information Studies, Syracuse University, 1998.)

Fusion Performance Example

Method	**NDCG@10**	**W/T/L**
BM25	0.212	—/—/—
SDM-Field	0.233	57/3/40
LambdaMART	0.225	59/2/39
DoubleFuse, v=all	0.300‡	80/1/19

Effectiveness comparison of three state-of-the-art ranking methods for the most common query variation for each topic from the ClueWeb12B UQV100 collection. Here ‡ means $p < 0.001$ in a Bonferroni corrected two-tailed t-test. Wins and Losses are computed when the score is 10% greater or less than the BM25 baseline on the original title-only topic run.

Fusion Performance Example



Per topic breakdown comparison of NDCG@10 differences of several state-of-the-art adhoc ranking techniques. The scores shown are the difference between the method and a simple BM25 bag-of-words run. The Double Fusion Technique uses all of the query variations (v=all) for each of the 100 topics, uses RRF Fusion, and combines two systems – SDM-Field and BM25.

Overview

1 Intro and Overview

2 Theoretical Foundations

3 Fusion in Practice

4 Learning and Fusion

5 Applications

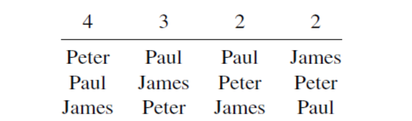
6 Conclusions and Future Directions

Computational Social Choice Theory

- The social choice theory field is mainly concerned with the aggregation of individual preferences so as to produce a collective choice
  - Allocating private commodities fairly and efficiently given the various individual preferences
  - Selecting a public outcome (e.g., candidate) given individual preferences (votes)
- Computational Social Choice is about applying social choice theory in computational problems (e.g., using voting rules for rank aggregation/fusion) and using computational frameworks to analyze and invent social choice mechanisms (e.g., analyzing the computational complexity of computing voting rules)

[1] F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. D. Procaccia. “Handbook of Computational Social Choice”. 2016.

Voting Rules



4	3	2	2
Peter	Paul	Paul	James
Paul	James	Peter	Peter
James	Peter	James	Paul

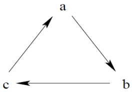
- **Condorcet** winner (Peter): an item that defeats every other item in strict majority sense.
    - A voting rule is a *Condorcet extension* if for each partition of the candidates $(C, \bar{C})$ s.t. for any $x \in C$ and $y \in \bar{C}$ the majority prefers x to y, then x will be ranked above y (Trunchon '98, Dwork et al. '01).
- **Plurality** rule (Paul) (not Condorcet): number of lists where the item is ranked first.
- **Copeland** rule (1951) (Peter) (Condorcet): number of pairwise victories minus number of pairwise defeats.
- **Borda** rule/count (1770) (Peter) (not Condorcet): the score of an item with respect to a list is the number of items in the list that are ranked lower.
    - Scores are summed over the lists.
    - This is a linear fusion method; more details later.

1. F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. D. Procaccia. “Handbook of Computational Social Choice.” 2016.
2. M. Trunchon “An extension of the Condorcet criterion and Kemeny orders.” cahier 98-15 du Centre de Recherche en Économie et Finance Applique ’es, 1998.
3. C. Dwork, R. Kumar, M. Naor and D. Sivakumar. “Rank Aggregation Methods for the Web”. In Proc. WWW, pages 613–622, 2001.

Condorcet Fusion

The Condorcet paradox:

1: a, b, c
1: b, c, a
1: c, a, b



The Condorcet fusion algorithm:

- Graph $G = (V, E)$; V: candidates; $(u, v) \in E$: iff v would receive at least the same number of votes as u in a head-to-head competition.
- Induce a DAG based on strongly connected components.
- Topological sort of the DAG.
- All candidates in the same strongly connected component are scored equally.
- For n candidates and k voters: $O(n^2k)$; can reduce to $O(nk \log n)$ by finding Condorcet paths.
- Weighted Condorcet: each vote is weighted by a weight assigned to the voter.

[1] M. Montague and J. A. Aslam. “Condorcet fusion for improved retrieval”. In Proc. CIKM, pages 427–433, 2001.

Kemeny Rank Aggregation

Input: Ranked lists: $L_1, \ldots, L_m$
Output: Aggregated (fused) list: L_{fuse}
Inter-list distance measure: Kendall's τ (K)

Kemeny (optimal) rank aggregation (Kemeny '59)

$$L_{fuse} \overset{def}{=} \underset{L}{\text{argmin}} \sum_{L_i} K(L, L_i)$$

- Important axiomatic properties
- Maximum likelihood interpretation (Young '88)
- Computing Kemeny is NP-Hard even when $m = 4$ (Dwork et al. '01)
  - Polynomial time approximation using Spearman's footrule distance
- Local Kemenization (Dwork et al. '01)
  - Satisfies extended Condorcet; can be applied on top of any rank aggregation function; polynomial time

The Fusion Hypothesis

Fusing retrieved lists should result in performance superior to that of using each of the lists alone

Early Empirical Evidence

- Combining document representations (Katzer et al. ’82)
- Combining Boolean and free text representations of queries (Turtle&Croft ’91)
- Combining Boolean query representations (Belkin et al. ’93)

1. P. Das-Gupta and J. Katzer. “A Study of the Overlap Among Document Representations”. In Proc. SIGIR, pp 106-114, 1983.
2. N. J. Belkin and C. Cool and W. B. Croft and J. P. Callan. “The effect of multiple query representations on information retrieval system performance”. In Proc. SIGIR, pages 339–346, 1993.
3. H. R. Turtle and W. B. Croft. “Evaluation of an Inference Network-Based Retrieval Model”. ACM Trans. Inf. Syst. 9(3): 187-222, 1991.

"Formal" Support for the Fusion Hypothesis

- The skimming and chorus effects (Diamond '96, Vogt&Cottrell '99)
- The probability ranking principle (Robertson '77)
- Combining experts' opinions (Thompson '90)
- BayesFuse (Aslam&Montague '01)
- The benefits of averaging the decisions of classifiers whose outputs are independent (Tumer&Ghosh '99)
- Croft '00:

$$\log O(H|E, e) = \log O(H|E) + \log L(e|H)$$

  - H, E, e are the hypothesis, history and new evidence, respectively
  - $O(H|E, e) = \frac{P(H|E,e)}{P(\neg H|E,e)}$
  - $O(H|E) = \frac{P(H|E)}{P(\neg H|E)}$
  - $L(e|H) = \frac{P(e|H)}{P(e|\neg H)}$
  - Independence assumption: $P(e|H, E) = p(e|H)$

When is Fusion Effective?

Hypothesis: When the overlap between relevant documents in the retrieved lists is higher than that between the non-relevant documents

- The chorus effect

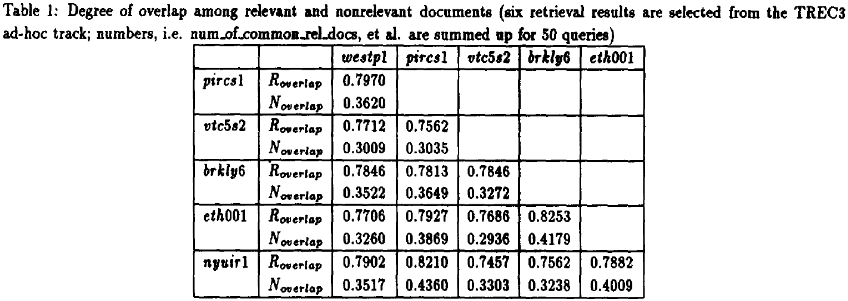


Table 1: Degree of overlap among relevant and nonrelevant documents (six retrieval results are selected from the TREC3 ad-hoc track; numbers, i.e. num_of_common_rel_docs, et al. are summed up for 50 queries)

		westpl	*pircs1*	*vtc5s2*	*brkly6*	*eth001*
pircs1	$R_{overlap}$	0.7970				
	$N_{overlap}$	0.3620				
vtc5s2	$R_{overlap}$	0.7712	0.7562			
	$N_{overlap}$	0.3009	0.3035			
brkly6	$R_{overlap}$	0.7846	0.7813	0.7846		
	$N_{overlap}$	0.3522	0.3649	0.3272		
eth001	$R_{overlap}$	0.7706	0.7927	0.7686	0.8253	
	$N_{overlap}$	0.3260	0.3869	0.2936	0.4179	
nyuir1	$R_{overlap}$	0.7902	0.8210	0.7457	0.7562	0.7882
	$N_{overlap}$	0.3517	0.4360	0.3303	0.3238	0.4009

$$R_{overlap} \overset{def}{=} \frac{2R_{common}}{R_1 + R_2} \qquad N_{overlap} \overset{def}{=} \frac{2N_{common}}{N_1 + N_2}$$

R_{common}: # of shared rel documents; R_1, R_2: # of rel documents in the first and second lists, respectively

1. J. H. Lee. “Analyses of multiple evidence combination”. In Proc. SIGIR, pages 180–188, 1995.

“Disproving” Lee’s Hypothesis?

New hypothesis: Fusion is effective if the lists contain unique relevant documents at top ranks (skimming effect)

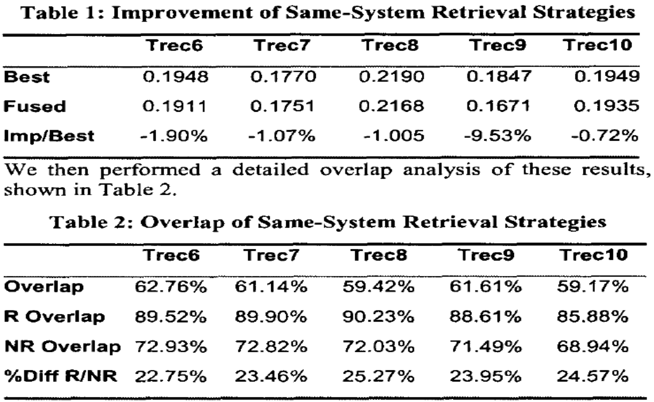


Table 1: Improvement of Same-System Retrieval Strategies

	Trec6	Trec7	Trec8	Trec9	Trec10
Best	0.1948	0.1770	0.2190	0.1847	0.1949
Fused	0.1911	0.1751	0.2168	0.1671	0.1935
Imp/Best	-1.90%	-1.07%	-1.005	-9.53%	-0.72%

We then performed a detailed overlap analysis of these results, shown in Table 2.

Table 2: Overlap of Same-System Retrieval Strategies

	Trec6	Trec7	Trec8	Trec9	Trec10
Overlap	62.76%	61.14%	59.42%	61.61%	59.17%
R Overlap	89.52%	89.90%	90.23%	88.61%	85.88%
NR Overlap	72.93%	72.82%	72.03%	71.49%	68.94%
%Diff R/NR	22.75%	23.46%	25.27%	23.95%	24.57%

[1] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, D. A. Grossman, and N. Goharian. “Disproving the fusion hypothesis: An analysis of data fusion via effective information retrieval strategies”. In Proc. SAC, pages 823–827, 2003.

“Disproving” Lee’s Hypothesis? (contd.)

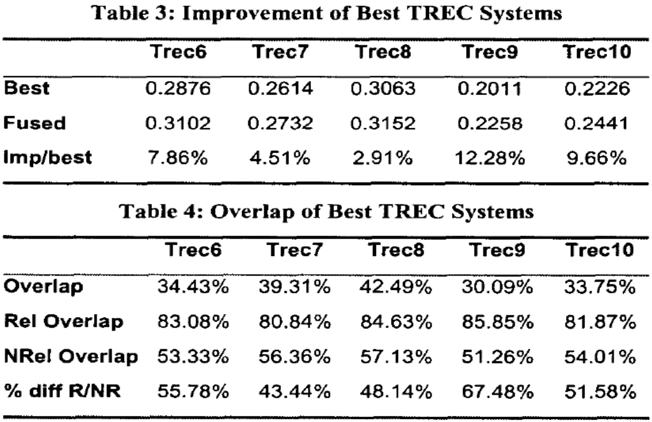


Table 3: Improvement of Best TREC Systems

	Trec6	Trec7	Trec8	Trec9	Trec10
Best	0.2876	0.2614	0.3063	0.2011	0.2226
Fused	0.3102	0.2732	0.3152	0.2258	0.2441
Imp/best	7.86%	4.51%	2.91%	12.28%	9.66%

Table 4: Overlap of Best TREC Systems

	Trec6	Trec7	Trec8	Trec9	Trec10
Overlap	34.43%	39.31%	42.49%	30.09%	33.75%
Rel Overlap	83.08%	80.84%	84.63%	85.85%	81.87%
NRel Overlap	53.33%	56.36%	57.13%	51.26%	54.01%
% diff R/NR	55.78%	43.44%	48.14%	67.48%	51.58%

1. S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, D. A. Grossman, and N. Goharian. “Disproving the fusion hypothesis: An analysis of data fusion via effective information retrieval strategies”. In Proc. SAC, pages 823–827, 2003.

“Disproving” Lee’s Hypothesis? (contd.)

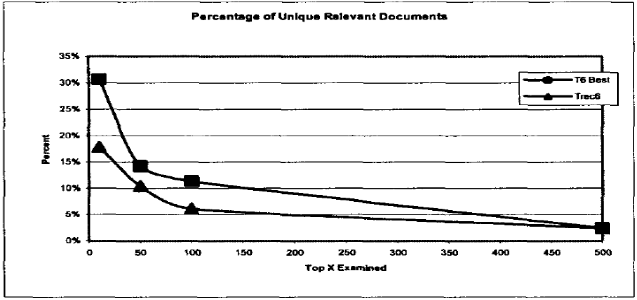


Figure 1: TREC-6 Unique Relevant Document Analysis

1. S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, D. A. Grossman, and N. Goharian. “Disproving the fusion hypothesis: An analysis of data fusion via effective information retrieval strategies”. In Proc. SAC, pages 823–827, 2003.

Fusing Best vs. Randomly Selected TREC Runs

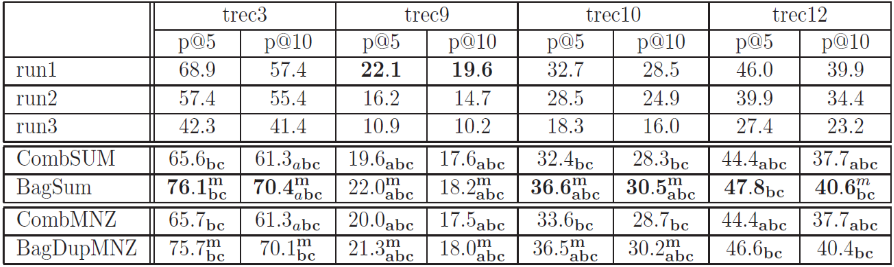
Fusing the best runs

	trec3		trec9		trec10		trec12	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
opt. run	76.0	72.2	60.0	**53.1**	63.2	58.8	54.5	48.6
run1	74.4	72.2	60.0	**53.1**	63.2	58.8	51.1	44.8
run2	72.8	67.6	$45.8^{\mathbf{o}}$	$38.8^{\mathbf{o}}$	54.4	50.2	52.5	48.6
run3	76.0	71.2	$38.3^{\mathbf{o}}$	$34.6^{\mathbf{o}}$	55.6	$46.8^{\mathbf{o}}$	51.5	45.2^{o}
CombSUM	$80.8_{a\mathbf{b}}$	$74.6_{\mathbf{b}}$	52.9_{bc}	$48.5_{\mathbf{bc}}$	$71.2^{o}_{a\mathbf{bc}}$	$\mathbf{61.0}_{\mathbf{bc}}$	53.7	$\mathbf{49.2}_{\mathbf{ac}}$
BagSum	$\mathbf{83.2}^{o}_{\mathbf{ab}c}$	$78.8^{\mathbf{om}}_{\mathbf{abc}}$	$59.6^{m}_{\mathbf{bc}}$	$48.1_{\mathbf{bc}}$	$71.2^{o}_{a\mathbf{bc}}$	$\mathbf{61.0}_{\mathbf{bc}}$	55.4_{ac}	$\mathbf{49.2}_{\mathbf{ac}}$
CombMNZ	$80.8_{a\mathbf{b}}$	$74.6_{\mathbf{b}}$	$55.0_{\mathbf{bc}}$	$48.8_{\mathbf{bc}}$	$71.2^{o}_{a\mathbf{bc}}$	$\mathbf{61.0}_{\mathbf{bc}}$	53.9	$\mathbf{49.2}_{\mathbf{ac}}$
BagDupMNZ	$\mathbf{83.2}_{\mathbf{ab}}$	$\mathbf{79.0}^{\mathbf{om}}_{\mathbf{abc}}$	$\mathbf{60.4}^{\mathbf{m}}_{\mathbf{bc}}$	$47.9_{\mathbf{bc}}$	$\mathbf{72.0}^{o}_{a\mathbf{bc}}$	$\mathbf{61.0}_{\mathbf{bc}}$	$\mathbf{56.6}^{\mathbf{m}}_{\mathbf{a}b\mathbf{c}}$	$49.0_{\mathbf{ac}}$

[1] A. K. Kozorovitzky and O. Kurland. “From ”Identical” to ”Similar””: Fusing Retrieved Lists Based on Inter-Document Similarities”. J. Artif. Intell. Res. 41, pages 267–296, 2011.

Fusing Best vs. Randomly Selected Runs (contd.)

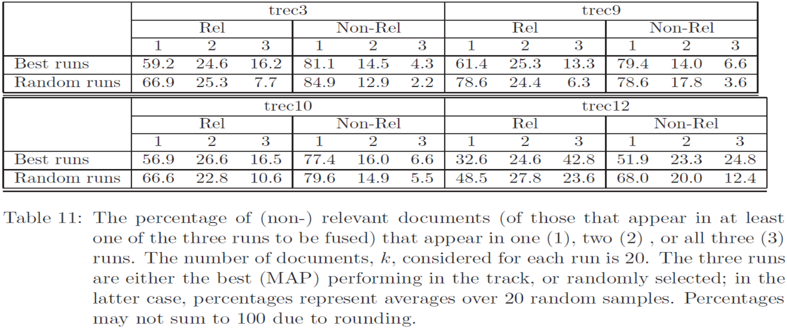
Fusing randomly selected runs



	trec3		trec9		trec10		trec12	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
run1	68.9	57.4	**22.1**	**19.6**	32.7	28.5	46.0	39.9
run2	57.4	55.4	16.2	14.7	28.5	24.9	39.9	34.4
run3	42.3	41.4	10.9	10.2	18.3	16.0	27.4	23.2
CombSUM	$65.6_{\mathbf{bc}}$	$61.3_{a\mathbf{bc}}$	$19.6_{\mathbf{abc}}$	$17.6_{\mathbf{abc}}$	$32.4_{\mathbf{bc}}$	$28.3_{\mathbf{bc}}$	$44.4_{\mathbf{abc}}$	$37.7_{\mathbf{abc}}$
BagSum	$\mathbf{76.1}^{\mathrm{m}}_{\mathbf{bc}}$	$\mathbf{70.4}^{\mathrm{m}}_{a\mathbf{bc}}$	$22.0^{\mathrm{m}}_{\mathbf{abc}}$	$18.2^{\mathrm{m}}_{\mathbf{abc}}$	$\mathbf{36.6}^{\mathrm{m}}_{\mathbf{abc}}$	$\mathbf{30.5}^{\mathrm{m}}_{\mathbf{abc}}$	$\mathbf{47.8}_{\mathbf{bc}}$	$\mathbf{40.6}^{m}_{\mathbf{bc}}$
CombMNZ	$65.7_{\mathbf{bc}}$	$61.3_{a\mathbf{bc}}$	$20.0_{\mathbf{abc}}$	$17.5_{\mathbf{abc}}$	$33.6_{\mathbf{bc}}$	$28.7_{\mathbf{bc}}$	$44.4_{\mathbf{abc}}$	$37.7_{\mathbf{abc}}$
BagDupMNZ	$75.7^{\mathrm{m}}_{\mathbf{bc}}$	$70.1^{\mathrm{m}}_{\mathbf{bc}}$	$21.3^{\mathrm{m}}_{\mathbf{abc}}$	$18.0^{\mathrm{m}}_{\mathbf{abc}}$	$36.5^{\mathrm{m}}_{\mathbf{abc}}$	$30.2^{\mathrm{m}}_{\mathbf{abc}}$	$46.6_{\mathbf{bc}}$	$40.4_{\mathbf{bc}}$

1. A. K. Kozorovitzky and O. Kurland. “From ”Identical” to ”Similar””: Fusing Retrieved Lists Based on Inter-Document Similarities”. J. Artif. Intell. Res. 41, pages 267–296, 2011.

Fusing Best vs. Randomly Selected Runs (contd.)



	trec3						trec9					
	Rel			Non-Rel			Rel			Non-Rel		
	1	2	3	1	2	3	1	2	3	1	2	3
Best runs	59.2	24.6	16.2	81.1	14.5	4.3	61.4	25.3	13.3	79.4	14.0	6.6
Random runs	66.9	25.3	7.7	84.9	12.9	2.2	78.6	24.4	6.3	78.6	17.8	3.6

	trec10						trec12					
	Rel			Non-Rel			Rel			Non-Rel		
	1	2	3	1	2	3	1	2	3	1	2	3
Best runs	56.9	26.6	16.5	77.4	16.0	6.6	32.6	24.6	42.8	51.9	23.3	24.8
Random runs	66.6	22.8	10.6	79.6	14.9	5.5	48.5	27.8	23.6	68.0	20.0	12.4

Table 11: The percentage of (non-) relevant documents (of those that appear in at least one of the three runs to be fused) that appear in one (1), two (2) , or all three (3) runs. The number of documents, k, considered for each run is 20. The three runs are either the best (MAP) performing in the track, or randomly selected; in the latter case, percentages represent averages over 20 random samples. Percentages may not sum to 100 due to rounding.

1. A. K. Kozorovitzky and O. Kurland. “From ”Identical” to ”Similar””: Fusing Retrieved Lists Based on Inter-Document Similarities”. J. Artif. Intell. Res. 41, pages 267–296, 2011.

Regression Analysis

$p_{i,} J_i$: effectiveness of the retrieved lists

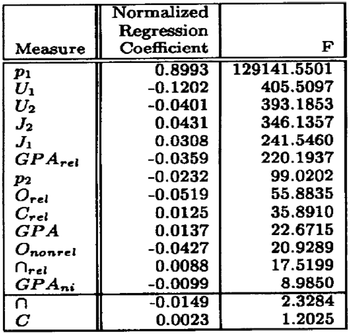
$GPA_{,} GPA_{rel}, GPA_{ni}$: Gutman's Point Alienation between retrieval scores in the lists (for all, relevant and non-relevant documents)

$C, C_{rel,}$: linear correlation between mean-normalized retrieval scores of all and rel docs

U_i: # of unique rel docs contributed by list *i*

O_{rel}, O_{nonrel}: Lee's overlap between rel and non-rel docs in the lists

$\cap_{rel,} \cap_{nonrel}$: # of shared rel and non-rel docs

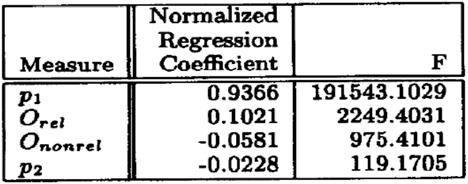


Measure	Normalized Regression Coefficient	F
p_1	0.8993	129141.5501
U_1	-0.1202	405.5097
U_2	-0.0401	393.1853
J_2	0.0431	346.1357
J_1	0.0308	241.5460
GPA_{rel}	-0.0359	220.1937
p_2	-0.0232	99.0202
O_{rel}	-0.0519	55.8835
C_{rel}	0.0125	35.8910
GPA	0.0137	22.6715
O_{nonrel}	-0.0427	20.9289
$\cap_{rel}$	0.0088	17.5199
GPA_{ni}	-0.0099	8.9850
$\cap$	-0.0149	2.3284
C	0.0023	1.2025

Table 3: Results of Linear Regression for Predicting Combination's Average Precision (r^2=0.94)

[1] C. C. Vogt and G. W. Cottrell. "Predicting the performance of linearly combined IR systems". In Proc. SIGIR, pages 190–196, 1998.

Regression Analysis (contd.)



Measure	Normalized Regression Coefficient	F
p_1	0.9366	191543.1029
O_{rel}	0.1021	2249.4031
O_{nonrel}	-0.0581	975.4101
p_2	-0.0228	119.1705

Table 4: Results of Linear Regression for Predicting Combination's Average Precision (r^2=0.94)

Ng&Kantor showed, using linear discriminant analysis, that the ratio of lists' precision values and their dissimilarity (Kendall-τ) can be used to predict fusion effectiveness to a descent extent

1. C. C. Vogt and G. W. Cottrell. "Predicting the performance of linearly combined IR systems". In Proc. SIGIR, pages 190–196, 1998.
2. K. B. Ng and P. P. Kantor. "An investigation of the preconditions for effective data fusion in information retrieval: A pilot study", 1998.

Formal Analysis of Linear Fusion Between Two Lists

Linear fusion of lists L_1 and L_2

$$F_{linear}(d;q) \overset{def}{=} \omega_1 s_{L_1}(d) + \omega_2 s_{L_1}(d) = \sin(\omega) s_{L_1}(d) + \cos(\omega) s_{L_1}(d)$$

Formal analysis which utilizes the mean of retrieval scores of relevant and non-relevant documents in a list

Formal findings that provide support/explanation to

- The chorus (but not skimming) effect
- Empirical finding that fusion is effective if the lists share relevant documents but not non-relevant documents and one of the lists is highly effective

[1] C. C. Vogt and G. W. Cottrell: “Fusion via linear combination of scores.” Information Retrieval, 1(3) pages 151–173, 1999.

Fusion Frameworks

- Evidential reasoning (Lalmas ’02)
- Geometric probabilistic framework (Wu ’07)
- Statistical principles (Wu ’09)
- A probabilistic framework (Anava et al. ’16)
- Learning frameworks (Sheldon et al. ’11 and Lee et al. ’15)
    - To be discussed later

Evidential Reasoning

- Based on Ruspini's ('86) evidential reasoning theory (logic and probability)

Macro-level view

- Symbolizing the knowledge induced from a retrieved list
  - Knowledge: rank positions of documents and their scores, terms in the title and abstract of the documents, etc.
- Combination of knowledge yields a description of the fused list

In practice

- Specific estimates of documents' properties and corresponding probabilities are needed for deriving a specific fusion method

1. M. Lalmas. "A formal model for data fusion". Proc. of FQAS, pages 274–288, 2002.
2. E. H. Ruspini. "The logical foundations of evidential reasoning". Tech. Rep. 408, SRI International, 1986.

Geometric probabilistic framework

- A list is represented as a vector of the relevance probabilities assigned to documents in the list
- Effectiveness of a list is measured using the Euclidean distance from a vector of “true” probabilities
  - The Euclidean distance is connected with p@k
- A centroid of the lists’ vectors is an effective result with respect to individual lists (i.e., CombSUM is effective)
- For CombSUM to be effective, lists should be of equal effectiveness and be quite different from each other (in terms of assigned probabilities)

[1] S. Wu and F. Crestani. “A geometric framework for data fusion in information retrieval”. Inf. Syst., 50, pages 20–35, 2015.

Statistical Principles

- Justification of CombSUM based on the average of a sample being an unbiased estimate for the true mean
- Justification of weighted linear fusion based on stratified sampling

S. Wu. Applying statistical principles to data fusion in information retrieval. Expert Systems with Applications, 36(2):2997–3006, 2009.

A probabilistic framework

- Document d is ranked by its relevance likelihood: $p(d|q,r)$; r is the relevance event
- θ_x: representation of text x
- Key point: a ranked document list retrieved for a query can serve as the query's representation

$$\hat{p}(d|q,r) \overset{def}{=} \int_{\theta_q} p(\theta_d|\theta_q,r)p(\theta_q|q,r)d\theta_q;$$

$$\hat{p}(d|q,r) \approx \sum_{i=1}^{m} p(d|L_i,r)p(L_i|q,r).$$

- Provides formal grounds for many linear fusion methods
- CombMNZ can also be derived

[1] Y. Anava, A. Shtok, O. Kurland and E. Rabinovich. "A Probabilistic Fusion Framework". In Proc. CIKM, pages 1463–1472, 2016.

Overview

1 Intro and Overview

2 Theoretical Foundations

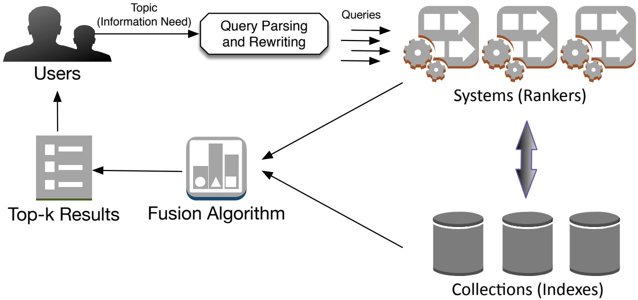
3 Fusion in Practice

4 Learning and Fusion

5 Applications

6 Conclusions and Future Directions

A Taxonomy of Fusion



Fusion can be at the ***collection level***, the ***system level***, or at the ***topic level***. Once a set of ranked items is obtained, they can be combined based on the scores for each item, or by the rank ordering of the items in each list.

System-Based Fusion Example

Topic	Rank	BM25 (Indri)		QL (Indri)		InL2 (Terrier)	
		DocID	Score	DocID	Score	DocID	Score
302	1	FBIS4-67701	22.628	FBIS4-67701	-6.342	LA043090-0036	20.103
302	2	LA043090-0036	22.326	LA043090-0036	-6.556	FBIS4-67701	19.802
302	3	LA013089-0022	16.079	FBIS4-30637	-7.018	LA071590-0110	15.725
302	4	FBIS4-30637	14.978	LA013089-0022	-7.029	FR940126-2-00106	14.725
302	5	LA031489-0032	12.222	LA090290-0118	-7.352	LA013089-0022	14.653

Top five results for the query "`poliomyelitis and post polio`" on the Newswire collection for three different systems. The first two runs are from Indri 5.12 using BM25 and the Language Model. The third run is from Terrier 4.2 using a Divergence from Randomness and Bose-Einstein 1 query expansion model.

Score Normalization

Normalization addresses the problem that relevance scores from different ranking functions / systems for the same item are not directly comparable. Montague and Aslam argue that normalized scores should possess three qualities:

1. **Shift invariant**: Both the shifted and unshifted scores should normalize to the same ordering.
2. **Scale invariant**: The scheme should be insensitive to scaling by a multiplicative constant. For example $e^{s_L(d)}$.
3. **Outlier insensitive**: A single item should not significantly affect the normalized scores for the other items.

1. M. Montague and J. Aslam: "Relevance Score Normalization for Metasearch." In Proc. CIKM, pages 427–433, 2001.

Score Normalization

1. **Min-Max (Standard Norm)** - Normalize the scores between 0 and 1 linearly for each list such that the minimum is shifted to 0, and the maximum is scaled to 1. $s_L^{minmax}(d) = \frac{s_L(d) - \min_{d' \in L} s_L(d')}{\max_{d' \in L} s_L(d') - \min_{d' \in L} s_L(d')}$
2. **Sum normalization (Sum Norm)** Shift the minimum value to 0, and scale the sum to 1. $s_L^{sum}(d) = \frac{s_L(d) - \min_{d' \in L} s_L(d')}{\sum_{d' \in L}(s_L(d') - \min_{d'' \in L} s_L(d''))}$
3. **Zero Mean and Unit Variance** - This method is based on the Z-score statistic. The idea is to shift the mean to 0, and scale the variance to 1. $s_L^{znorm}(d) = \frac{s_L(d) - \mu}{\sigma}$ where $\mu = \frac{1}{|L|} \sum_{d' \in L} s_L(d')$ and $\sigma = \sqrt{\frac{1}{|L|} \sum_{d' \in L}(s_L(d') - \mu)^2}$.

Note: In an implementation, adding a small ϵ to the $n \cdot$th item is not uncommon as originally this item had a non-zero score.

[1] M. Montague and J. Aslam: “Relevance Score Normalization for Metasearch.” In Proc. CIKM, pages 427–433, 2001.

Min-Max Normalization Example

Topic	Rank	BM25 (Indri)		QL (Indri)		InL2 (Terrier)	
		DocID	Score	DocID	Score	DocID	Score
302	1	FBIS4-67701	22.628	FBIS4-67701	-6.342	LA043090-0036	20.103
302	2	LA043090-0036	22.326	LA043090-0036	-6.556	FBIS4-67701	19.802
302	3	LA013089-0022	16.079	FBIS4-30637	-7.018	LA071590-0110	15.725
302	4	FBIS4-30637	14.978	LA013089-0022	-7.029	FR940126-2-00106	14.725
302	5	LA031489-0032	12.222	LA090290-0118	-7.352	LA013089-0022	14.653

Identify the minimum and maximum score for each retrieval list and apply the transform $s_L^{minmax}(d) = \frac{s_L(d) - \min_{d' \in L} s_L(d')}{\max_{d' \in L} s_L(d') - \min_{d' \in L} s_L(d')}$

Min-Max Normalization Example

Topic	Rank	BM25 (Indri)		QL (Indri)		InL2 (Terrier)	
		DocID	Score	DocID	Score	DocID	Score
302	1	FBIS4-67701	22.628	FBIS4-67701	-6.342	LA043090-0036	20.103
302	2	LA043090-0036	22.326	LA043090-0036	-6.556	FBIS4-67701	19.802
302	3	LA013089-0022	16.079	FBIS4-30637	-7.018	LA071590-0110	15.725
302	4	FBIS4-30637	14.978	LA013089-0022	-7.029	FR940126-2-00106	14.725
302	5	LA031489-0032	12.222	LA090290-0118	-7.352	LA013089-0022	14.653

Identify the minimum and maximum score for each retrieval list and apply the transform $s_L^{minmax}(d) = \frac{s_L(d) - \min_{d' \in L} s_L(d')}{\max_{d' \in L} s_L(d') - \min_{d' \in L} s_L(d')}$

The Indri scores are negative. Does that matter?

Min-Max Normalization Example

Topic	Rank	BM25 (Indri)		QL (Indri)		InL2 (Terrier)	
		DocID	Score	DocID	Score	DocID	Score
302	1	FBIS4-67701	22.628	FBIS4-67701	0.00176	LA043090-0036	20.103
302	2	LA043090-0036	22.326	LA043090-0036	0.00142	FBIS4-67701	19.802
302	3	LA013089-0022	16.079	FBIS4-30637	0.00090	LA071590-0110	15.725
302	4	FBIS4-30637	14.978	LA013089-0022	0.00088	FR940126-2-00106	14.725
302	5	LA031489-0032	12.222	LA090290-0118	0.00064	LA013089-0022	14.653

Identify the minimum and maximum score for each retrieval list and apply the transform $s_L^{minmax}(d) = \frac{s_L(d) - \min_{d' \in L} s_L(d')}{\max_{d' \in L} s_L(d') - \min_{d' \in L} s_L(d')}$

The Indri scores are negative. Does that matter?

Since we know that the LM scores produced by Indri are log smoothed (negative cross entropy), we can convert the scores with the transform $e^{s_L(d)}$ before normalization. However, we don't always know, so you can also just work directly with the negative scores.

Min-Max Normalization Example

Topic	Rank	BM25 (Indri)		QL (Indri)		InL2 (Terrier)	
		DocID	Score	DocID	Score	DocID	Score
302	1	FBIS4-67701	1.000	FBIS4-67701	1.000	LA043090-0036	1.000
302	2	LA043090-0036	0.970	LA043090-0036	0.696	FBIS4-67701	0.944
302	3	LA013089-0022	0.370	FBIS4-30637	0.232	LA071590-0110	0.197
302	4	FBIS4-30637	0.265	LA013089-0022	0.214	FR940126-2-00106	0.013
302	5	LA031489-0032	0.000	LA090290-0118	0.000	LA013089-0022	0.000

Identify the minimum and maximum score for each retrieval list and apply the transform $s_L^{minmax}(d) = \frac{s_L(d) - \min_{d' \in L} s_L(d')}{\max_{d' \in L} s_L(d') - \min_{d' \in L} s_L(d')}$

The Indri scores are negative. Does that matter?

Since we know that the LM scores produced by Indri are log smoothed (negative cross entropy), we can convert the scores with the transform $e^{s_L(d)}$ before normalization. However, we don't always know, so you can also just work directly with the negative scores.

Fitting Score Distributions

The score normalization techniques we have seen scale retrieval scores (often to the same range), but ignore the (potentially) different score distributions across lists

Manmatha et al. suggested to model the score distribution of each list and use the average of the relevance posterior probabilities of a document over the lists as a fusion score

- The assumption is that scores of relevant documents follow a Gaussian distribution and scores of non-relevant documents follow an exponential distribution
- The paramaters of a mixture model were learned using the EM algorithm
- Arampatzis and Robertson showed that Gamma-Gamma is the most suitable mixture and that the Gaussian-Exponential is a good approximation

1. R. Manmatha, T. Rath and F. Feng. "Modeling Score Distributions for Combining the Outputs of Search Engines". In Proc. SIGIR, pages 267–275, 2001.
2. A. Arampatzis and Stephen Robertson. "Modeling score distributions in information retrieval". Inf. Retr. 14(1): 26-46 (2011).

Score-based Fusion

$$m \overset{def}{=} |\{L_i : d \in L_i\}|$$

Name	Author	Function	Description
CombSUM	Fox and Shaw (1994)	$\sum_{L_i:d\in L_i} s_{L_i}(d)$	Adds the retrieval scores of documents contained in more than one list and rearranges the order. Also possible to take the minimum, maximum, or median of the scores.
CombMNZ	Fox and Shaw (1994)	$m \cdot \sum_{L_i:d\in L_i} s_{L_i}(d)$	Adds the retrieval scores of documents contained in more than one list, and multiplies their sum by the number of lists where the document occurs.
CombANZ	Fox and Shaw (1994)	$\frac{1}{m} \cdot \sum_{L_i:d\in L_i} s_{L_i}(d)$	Adds the retrieval scores of documents contained in more than one list, and divides their sum by the number of lists where the document occurs.
Linear	Vogt and Cottrell (1999)	$\sum_{L_i:d\in L_i} w_i \cdot s_{L_i}(d)$	Similar to CombSUM, but allows a different weight to be applied to each list.

Rank-based Fusion

$$m \overset{def}{=} |\{L_i : d \in L_i\}|; n \overset{def}{=} |L_i|$$

Name	Author	Function	Description
Borda	Aslam and Montague (2001)	$\sum_{L_i:d\in L_i} \frac{n - r_{L_i}(d) + 1}{n}$	Voting algorithm that sums the difference in rank position from the total number of document candidates in each list.
RRF	Cormack et al. (2009)	$\sum_{L_i:d\in L_i} \frac{1}{\nu + r_{L_i}(d)}$	Discounts the weight of documents occurring deep in retrieved lists using a reciprocal distribution. The parameter ν is typically set to 60.
ISR	Mourao et al. (2014)	$m \cdot \sum_{L_i:d\in L_i} \frac{1}{r_{L_i}(d)^2}$	Inspired by RRF, but discounts documents occurring lower in the ranking more severely.
logISR	Mourao et al. (2014)	$\log m \cdot \sum_{L_i:d\in L_i} \frac{1}{r_{L_i}(d)^2}$	Similar to ISR but with logarithmic document frequency normalization.
RBC	Bailey et al. (2017)	$\sum_{L_i:d\in L_i} (1-\phi)\phi^{r_{L_i}(d)-1}$	Discounts the weights of documents following a geometric distribution, inspired by the RBP evaluation metric.
Markov Chains	Dwork et al. (2001)	stationary distribution	Transition from d to another document randomly selected from those ranked higher than d in the lists it appears in.

Rank-to-Score Transformations

$r_{L_i}(d)$: d's rank in L_i; H_i: the i·th harmonic number; ν is a free parameter

Method	Retrieval Score
Borda 1770	$\lvert L_i\rvert - r_{L_i}(d)$
Lee '97	$1 - \frac{r_{L_i}(d)-1}{\lvert L_i\rvert}$
Cormack et al. '09 (RR)	$\frac{1}{\nu+r_{L_i}(d)}$
Aslam et al. '05 (Measure)	$1 + H_{\lvert L_i\rvert} - H_{r_{L_i}(d)}$

Large-Scale Empirical Study

Datasets: TREC3, TREC7, TREC8, TREC9, TREC10, TREC12, TREC18, TREC19

Linear fusion over 10 randomly selected TREC runs

- Rank to score transformations: RR $>$ Measure $>$ Borda
- Retrieval score normalization: Z-Norm $=$ MinMax $>$ Mean
  - Variants of MinMax and Z-Norm were also evaluated (Markov et. al '12)
- Score vs. rank: In most cases, RR and Measure outperform (statistically significantly) Z-Norm, MinMax and Mean

1. Y. Anava, A. Shtok, O. Kurland and E. Rabinovich. "A Probabilistic Fusion Framework". In Proc. CIKM, pages 1463–1472, 2016.
2. I. Markov, A. Arampatzis and F. Crestani. "Unsupervised linear score normalization revisited". In Proc. SIGIR 2012, pages 1161–1162, 2012.

Query Variations

Topic 304

Title: Endangered Species (Mammals)

Description: Compile a list of mammals that are considered to be endangered, identify their habitat and, if possible, specify what threatens them.

Narrative: Any document identifying a mammal as endangered is relevant. Statements of authorities disputing the endangered status would also be relevant. A document containing information on habitat and populations of a mammal identified elsewhere as endangered would also be relevant even if the document at hand did not identify the species as endangered. Generalized statements about endangered species without reference to specific mammals would not be relevant.

Human Generated Variations: endangered mammals habitat threat; endangered mammals; list endangered mammals; endangered mammals and their habitats; population of endangered mammals; names of endangered mammals; environmental change and endangered mammals

Where do they come from?

- Crowdsourcing (or even you!)
- Query Logs (reformulations in a single session, or clustering).
- Relevance modeling (external resources work very well here)
- Virtual assistants / Conversational IR

Failure / Risk Analysis

- Generally effectiveness is reported as an average over multiple topics, but this often hides important differences when comparing systems.
- In search, our goal is to make systems better for *all* topics, but this rarely happens in practice.
- Several metrics have been proposed recently to measure *risk sensitivity*, and when used in conjunction with a *failure analysis*, important performance trends can be uncovered.
- $\text{URisk}_\alpha = \frac{1}{|Q|}\left[\sum \mathit{Win} - (1+\alpha)\cdot\sum \mathit{Loss}\right]$
- Here *Win* and *Loss* are the number of times a System A is better or worse than System B on a topic by topic basis.
- Inferential risk analysis can be performed using TRisk, which is a generalization of URisk to follow a Studentized *t*-distribution.

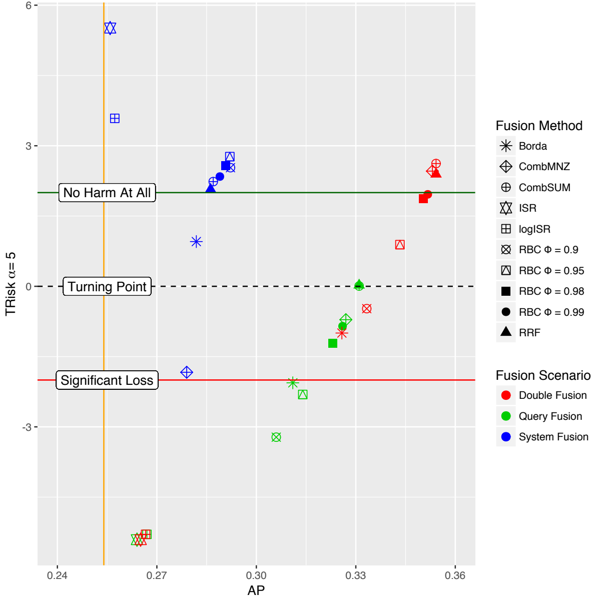
[1] B. T. Dinçer, C. Macdonald, and I. Ounis: "Hypothesis testing for the risk-sensitive evaluation of retrieval systems." In Proc. SIGIR, pages 23–32, 2014.
[2] https://github.com/rmit-ir/trisk

TREC Robust Fusion Experiments (Benham & Culpepper 2017)

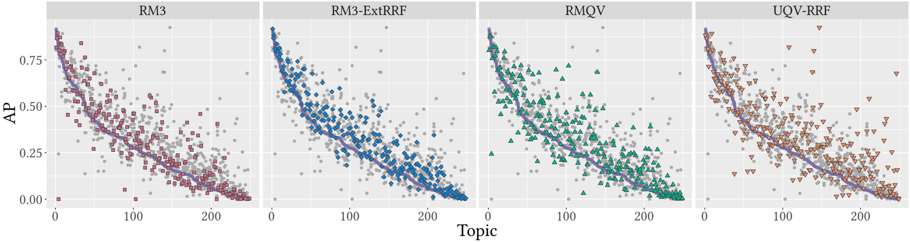
System	AP	Wins	Losses
BM25	0.254	-	-
BM25+QE	0.292 ‡	130	62
FDM	0.264 †	86	66
FDM+QE	0.275 ‡	102	46
BM25+Fuse	0.331 ‡	156	39
BM25+QE+Fuse	0.340 ‡	166	41
FDM+Fuse	0.336 ‡	171	34
FDM+QE+Fuse	0.349 ‡	174	32

Effectiveness comparisons for all retrieval models on Robust04 using BM25 as a baseline. Wins and Losses are computed when the score is 10% greater or less than the BM25 baseline on the original title-only topic run.

TREC Robust Fusion Experiments (Cont'd)



TREC Robust Fusion Experiments



The per-topic AP scores for four different Relevance Modeling and Fusion approaches compared to the BM25 for 250 queries on the TREC 2004 Robust Track. baseline.

1. R. Benham, J. S. Culpepper, L. Gallagher, X. Lu, and J. Mackenzie: “Towards efficient and effective query variation generation.” In Proc. DESIRES, 2018. To appear.

Hands-on Fusion Lab

`https://github.com/jsc/sigir18-fusion-tutorial`

We now walk through a set of scripts and tools that show how to do the following:

- How to fuse system runs.
- How to fuse query variations
- How to perform double and triple fused runs.
- How to to compute t-risk and paired t-tests with Bonferroni correction.

Content-based Fusion

So far, all fusion methods have used either rank or retrieval score information. There are fusion methods that utilize the documents' content:

- Lawrence&Giles '98: # of (unique) query terms a document contains and their proximity
- Craswell et al. ('99) used reference term statistics as approximation to corpus statistics, and a term weighting scheme biased to the beginning of the document
- Tsikrika&Lalmas ('01) used title-based and summary-based features for tf-based ranking
  - Applying simple fusion upon lists re-ranked by title and summary based information was most effective
- Beitzel et al. ('05) used title, summary and URL based features; e.g., % of query character n-grams in the title and in the snippet, avg. distance between query terms in the title, URL path depth
  - Title-based features were the most effective
  - The performance was superior to that of rCombMNZ (rank-based CombMNZ)

Fusion Meets the Cluster Hypothesis

The cluster hypothesis (Jardine&van Rijsbergen ’71, van Rijsbergen ’79): Closely associated documents tend to be relevant to the same requests

The basic fusion principle: reward documents that are highly ranked in many of the lists

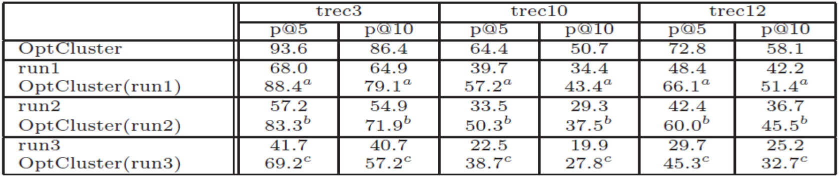
The “revised” fusion principle (Kozorovitzky&Kurland ’09): reward documents that are similar to (many) documents highly ranked in the lists

Methods

- Shou&Sanderson ’02: An in-degree centrality-based approach utilizing documents’ headlines fo fusion over disjoint collections
- Kozorovitzky&Kurland ’09, ’11: A Markov chain approach
- Liang et al. ’18: Efficient manifold-based regularization based on Diaz’s score regularization (’07)

A Cluster-Based Approach

Kozorovitzky&Kurland ’11, Liang et al. ’14



	trec3		trec10		trec12	
	p@5	p@10	p@5	p@10	p@5	p@10
OptCluster	93.6	86.4	64.4	50.7	72.8	58.1
run1	68.0	64.9	39.7	34.4	48.4	42.2
OptCluster(run1)	88.4^a	79.1^a	57.2^a	43.4^a	66.1^a	51.4^a
run2	57.2	54.9	33.5	29.3	42.4	36.7
OptCluster(run2)	83.3^b	71.9^b	50.3^b	37.5^b	60.0^b	45.5^b
run3	41.7	40.7	22.5	19.9	29.7	25.2
OptCluster(run3)	69.2^c	57.2^c	38.7^c	27.8^c	45.3^c	32.7^c

$$F(d;q) \overset{def}{=} (1-\lambda)p(d|q) + \lambda \sum_{c \in clusters} p(c|q)p(d|c)$$

Estimates:

- $p(d|q)$: standard fusion score of d
- $p(d|c)$: average similarity between d and c’s constituent documents
- $p(c|q)$: geometric mean of the standard fusion scores of c’s constituent documents

Retrieval List Selection

Linearly fusing (i) randomly selected lists (**2 Std Dev**), and (ii) lists produced by the methods most effective on a training set (**Best First Schedule**) vs. the list most effective for the test query (**Best Single System**) vs. the list produced by the system most effective on average over all test queries (**Average Single System**)

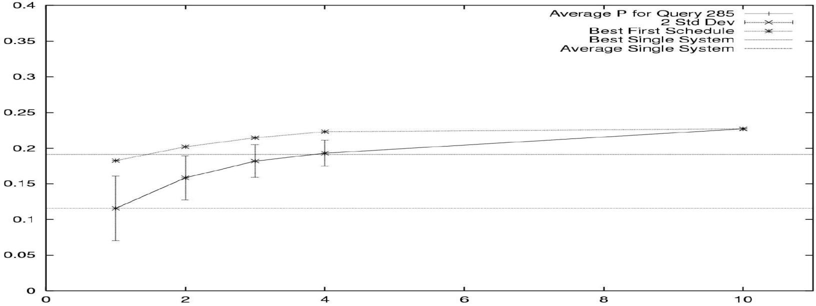


Figure 1: Expected Average **P** of the Combination as Function of the Number of Systems Combined, Compared with "Best First" Schedule, Query 285.

1. C. C. Vogt. How much more is better? Characterising the effects of adding more IR Systems to a combination. In Proc. RIAO, pages 457–475, 2000.

Retrieval List Selection (contd.)

Fusing a subset of the given lists

- Lists most similar to the centroid of all lists (Juárez-González et al. '10)
- A genetic algorithm utilizing past (train) performance of the retrieval systems (Gopalan&Batri '07)
- Weighing the lists using query-performance predictors (Raiber&Kurland '14)

Selecting a single list

- Selective query expansion (Amati et al. '04, Cronen-Townsend et al. '04)
- Selective cluster retrieval (Griffiths et al. '86, Liu&Croft '06, Levi et al. '16)
- Learning to select rankers (Balasubramanian&Allan '10)
- List most similar (in several respects) to the centroid of all lists (Juárez-González et al. '09)

Overview

1. Intro and Overview
2. Theoretical Foundations
3. Fusion in Practice
4. **Learning and Fusion**
5. Applications
6. Conclusions and Future Directions

Supervised Models

Most approaches focus on learning linear models:

$$\hat{p}(d|q,r) \approx \sum_{i=1}^{m} p(d|L_i,r)p(L_i|q,r)$$

- The list L_i was produced by system (retrieval method) M_i in response to the given query q
- A query train set, Q, with relevance judgments
- The document-list association: $s_{L_i}(d)$ is an estimate for $p(d|L_i,r)$
- List effectiveness: $w(L_i)$ is an estimate for $p(L_i|q,r)$

$$F(d;q) \overset{def}{=} \sum_{L_i:d \in L_i} s_{L_i}(d)w(L_i)$$

[1] Y. Anava, A. Shtok, O. Kurland and E. Rabinovich. “A Probabilistic Fusion Framework”. In Proc. CIKM, pages 1463–1472, 2016.

Connection to Learning-To-Rank

$$\hat{p}(d|q,r) \approx \sum_{i=1}^{m} p(d|L_i,r)p(L_i|q,r)$$

If $p(d|L_i,r)$ are given (“feature values”) and $p(L_i|q,r)$ are to be learned (“feature weights”), we get a linear learning-to-rank (LTR) approach

What are the differences in practice between learning linear LTR functions and learning to linearly fuse?

ProbFuse

Uniform list weights ($w(L_i)$)

$$s_{L_i}(d) \overset{def}{=} \frac{1}{k}\frac{1}{|Q|}\sum_{q_j \in Q}\frac{R_{k,q_j}}{R_{k,q_j} + NR_{k,q_j}}$$

k: the number of block in L_i in which d appears

R_{k,q_j} and NR_{k,q_j}: # of relevant (non-relevant) documents in the k·th block of the list retrieved by system M_i for query q_j in the training set

[1] D. Lillis, F. Toolan, R. W. Collier and J. Dunnion. “ProbFuse: a probabilistic approach to data fusion”. In Proc. SIGIR, pages 139–146, 2006.

SegFuse

A variant of ProbFuse with blocks of exponentially rising sizes and a modified fusion score function that also considers the normalized retrieval scores (“normScore”) of documents in the lists

Uniform list weights ($w(L_i)$)

$$s_{L_i}(d) \overset{def}{=} (1 + normScore_{L_i}(d)) \frac{1}{|Q|} \sum_{q_j \in Q} \frac{R_{k,q_j}}{All_{k,q_j}}$$

k: the number of block in L_i in which d appears

R_{k,q_j}, All_{k,q_j}: # of relevant documents and the overall # of documents, respectively, in the $k\cdot$th block of the list retrieved by system M_i for query q_j in the training set

[1] M. Shokouhi. “Segmentation of Search Engine Results for Effective Data-Fusion”. In Proc. ECIR, pages 185–197, 2007.

SlideFuse

Uniform list weights ($w(L_i)$)

PosFuse

$s_{L_i}(d)$ is the fraction of queries in Q for which M_i retrieved a relevant document at rank $r_{L_i}(d)$ (d's rank in L_i)

SlideFuse

$s_{L_i}(d)$ is the average over ranks $x \in [r_{L_i}(d) - a, \ldots, r_{L_i}(d) + b]$ of $s_{L_i}(d_x)$ used in PosFuse where d_x is the document at rank x of L_i; a and b are free parameters

[1] D. Lillis, L. Zhang, F. Toolan and R. W. Collier, D. Leonard and J. Dunnion. “Extending Probabilistic Data Fusion Using Sliding Windows”. In Proc. ECIR, pages 358–369, 2008.

MAPFuse

$w(L_i)$: the MAP of M_i over Q

$s_{L_i}(d) \overset{def}{=} \frac{1}{r_{L_i}(d)}$

1. D. Lillis, L. Zhang, F. Toolan and R. W. Collier, D. Leonard and J. Dunnion. “Estimating Probabilities for Effective Data Fusion”. In Proc. SIGIR, pages 347–354, 2010.

BayesFuse

cf. Thompson's ('90) combination of experts' opinions

$$P(r|d) = P(r|r_{L_1}(d), \ldots, r_{L_m}(d))$$

$$P(\bar{r}|d) = P(\bar{r}|r_{L_1}(d), \ldots, r_{L_m}(d))$$

$$O(r) \overset{rank}{=} \frac{p(r_{L_1}(d), \ldots, r_{L_m}(d)|r)}{p(r_{L_1}(d), \ldots, r_{L_m}(d)|\bar{r})}$$

$$O(r) \overset{rank}{=} \sum_{i=1}^{m} \log \frac{p(r_{L_i}(d)|r)}{p(r_{L_i}(d)|\bar{r})}$$

$p(r_{L_i}(d)|r)$ and $p(r_{L_i}(d)|\bar{r})$ are estimated using a query train set similarly to ProbFuse and SegFuse

[1] J. A. Aslam and M. Montague. "Models for metasearch". In Proc. SIGIR, pages 276–284, 2001.

[2] P. Thompson. "A Combination of Expert Opinion Approach To Probabilistic Information Retrieval, PART 1: The Conceptual Model". Information Processing and Management, 26(3):371382, 1990

Empirical Comparison

- SlideFuse slightly outperforms SegFuse; both outperform ProbFuse
- Adding list effectiveness measures to ProbFuse, SlideFuse and SegFuse results in substantial improvements

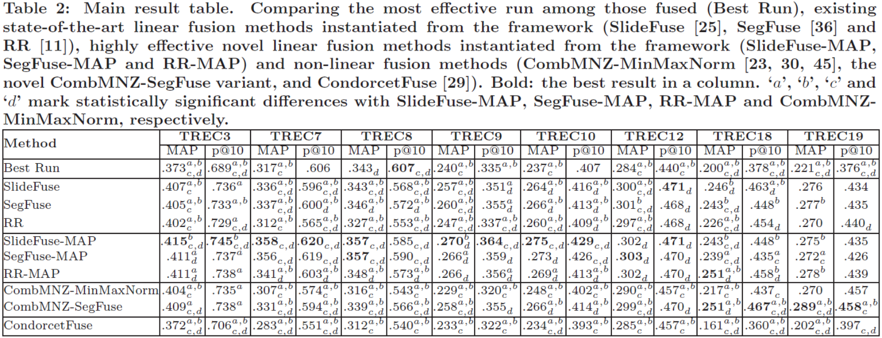


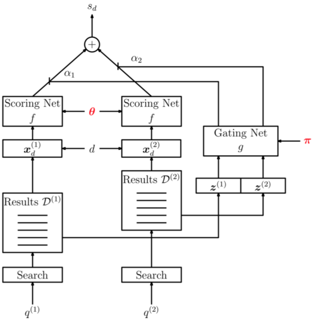
Table 2: Main result table. Comparing the most effective run among those fused (Best Run), existing state-of-the-art linear fusion methods instantiated from the framework (SlideFuse [25], SegFuse [36] and RR [11]), highly effective novel linear fusion methods instantiated from the framework (SlideFuse-MAP, SegFuse-MAP and RR-MAP) and non-linear fusion methods (CombMNZ-MinMaxNorm [23, 30, 45], the novel CombMNZ-SegFuse variant, and CondorcetFuse [29]). Bold: the best result in a column. '*a*', '*b*', '*c*' and '*d*' mark statistically significant differences with SlideFuse-MAP, SegFuse-MAP, RR-MAP and CombMNZ-MinMaxNorm, respectively.

Method	TREC3		TREC7		TREC8		TREC9		TREC10		TREC12		TREC18		TREC19	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
Best Run	$.373^{a,b}_{c,d}$	$.689^{a,b}_{c,d}$	$.317^{a,b}_{c}$	.606	$.343_{d}$	$\mathbf{.607}_{c,d}$	$.240^{a,b}_{c}$	$.335^{a,b}$	$.237^{a,b}_{c}$	.407	$.284^{a,b}_{c}$	$.440^{a,b}_{c}$	$.200^{a,b}_{c,d}$	$.378^{a,b}_{c,d}$	$.221^{a,b}_{c,d}$	$.376^{a,b}_{c,d}$
SlideFuse	$.407^{a,b}_{c}$	$.736^{a}$	$.336^{a,b}_{c,d}$	$.596^{a,b}_{c,d}$	$.343^{a,b}_{c,d}$	$.568^{a,b}_{c,d}$	$.257^{a,b}_{c,d}$	$.351^{a}_{d}$	$.264^{a,b}_{d}$	$.416^{a,b}_{d}$	$.300^{a,b}_{c,d}$	$\mathbf{.471}_{d}$	$.246^{b}_{d}$	$.463^{a,b}_{d}$	.276	.434
SegFuse	$.405^{a,b}_{c}$	$.733^{a,b}$	$.337^{a,b}_{c,d}$	$.600^{a,b}_{d}$	$.346^{a,b}_{d}$	$.572^{a,b}_{d}$	$.260^{a,b}_{c,d}$	$.355^{a}_{d}$	$.266^{a,b}_{d}$	$.413^{a,b}_{d}$	$.301^{b}_{c,d}$	$.468_{d}$	$.243^{b}_{c,d}$	$.448^{b}$	$.277^{b}$	.435
RR	$.402^{a,b}_{c}$	$.729^{a}_{c,d}$	$.312^{a,b}_{c}$	$.565^{a,b}_{c,d}$	$.327^{a,b}_{c,d}$	$.553^{a,b}_{c,d}$	$.247^{a,b}_{c,d}$	$.337^{a,b}_{c,d}$	$.260^{a,b}_{c,d}$	$.409^{a,b}_{d}$	$.297^{a,b}_{c,d}$	$.468_{d}$	$.226^{a,b}_{c,d}$	$.454_{d}$	.270	$.440_{d}$
SlideFuse-MAP	$\mathbf{.415}^{b}_{c,d}$	$\mathbf{.745}^{b}_{c,d}$	$\mathbf{.358}_{c,d}$	$\mathbf{.620}_{c,d}$	$\mathbf{.357}_{c,d}$	$.585_{c,d}$	$\mathbf{.270}^{b}_{d}$	$\mathbf{.364}_{c,d}$	$\mathbf{.275}_{c,d}$	$\mathbf{.429}_{c,d}$	$.302_{d}$	$\mathbf{.471}_{d}$	$.243^{b}_{c,d}$	$.448^{b}$	$.275^{b}$	.435
SegFuse-MAP	$.411^{a}_{d}$	$.737^{a}$	$.356_{c,d}$	$.619_{c,d}$	$\mathbf{.357}_{c,d}$	$.590_{c,d}$	$.266^{a}_{d}$	$.359_{d}$	$.273_{d}$	$.426_{c,d}$	$\mathbf{.303}_{d}$	$.470_{d}$	$.239^{a}_{c,d}$	$.435^{a}_{c}$	$.272^{a}_{c}$	.426
RR-MAP	$.411^{a}_{d}$	$.738^{a}$	$.341^{a,b}_{d}$	$.603^{a,b}_{d}$	$.348^{a,b}_{d}$	$.573^{a,b}_{d}$	$.266_{d}$	$.356^{a}_{d}$	$.269^{a}_{d}$	$.413^{a,b}_{d}$	$.302_{d}$	$.470_{d}$	$\mathbf{.251}^{a,b}_{d}$	$.458^{b}_{d}$	$.278^{b}$	.439
CombMNZ-MinMaxNorm	$.404^{a,b}_{c}$	$.735^{a}$	$.307^{a,b}_{c}$	$.574^{a,b}_{c}$	$.316^{a,b}_{c}$	$.543^{a,b}_{c}$	$.229^{a,b}_{c}$	$.320^{a,b}_{c}$	$.248^{a,b}_{c}$	$.402^{a,b}_{c}$	$.290^{a,b}_{c}$	$.457^{a,b}_{c}$	$.217^{a,b}_{c}$	$.437_{c}$	.270	.457
CombMNZ-SegFuse	$.409^{a}_{c,d}$	$.738^{a}$	$.331^{a,b}_{c,d}$	$.594^{a,b}_{c,d}$	$.339^{a,b}_{c,d}$	$.566^{a,b}_{c,d}$	$.258^{a,b}_{c,d}$	$.355_{d}$	$.266^{a,b}_{d}$	$.414^{a,b}_{d}$	$.299^{a,b}_{c,d}$	$.470_{d}$	$\mathbf{.251}^{a,b}_{d}$	$\mathbf{.467}^{a,b}_{c,d}$	$\mathbf{.289}^{a,b}_{c,d}$	$\mathbf{.458}^{a,b}_{c}$
CondorcetFuse	$.372^{a,b}_{c,d}$	$.706^{a,b}_{c,d}$	$.283^{a,b}_{c,d}$	$.551^{a,b}_{c,d}$	$.312^{a,b}_{c}$	$.540^{a,b}_{c}$	$.233^{a,b}_{c}$	$.322^{a,b}_{c}$	$.234^{a,b}_{c,d}$	$.393^{a,b}_{c}$	$.285^{a,b}_{c}$	$.457^{a,b}_{c}$	$.161^{a,b}_{c,d}$	$.360^{a,b}_{c,d}$	$.202^{a,b}_{c,d}$	$.397_{c,d}$

1. Y. Anava, A. Shtok, O. Kurland and E. Rabinovich. "A Probabilistic Fusion Framework". In Proc. CIKM, pages 1463–1472, 2016.

LambdaMerge

A linear fusion method: $\hat{p}(d|q,r) \approx \sum_{i=1}^{m} p(d|L_i,r)p(L_i|q,r)$
The basic idea: simultaneously learn $p(d|L_i,r)$ and $p(L_i|q,r)$.



- Issue m query formulations to a search engine, generated with a random walk over a click graph using several months of a Bing query log.
- Generate document-list features $\boldsymbol{x}_d^{(k)}$ – Score, Rank, isTopN, NormScore.
- Add gating features $\boldsymbol{z}^{(k)}$ covering "drift" and $D^{(k)}$ – Difficulty (List mean, skew, std, Clarity, RewriteLen, RAPP) and Drift (IsRewrite, RewriteRank, RewriteScore, Overlap@N).
- Learn θ (scoring) and π (gating) with LambdaRank to produce a weighted fusion score $F(d;q)$.
- Compare against RAPP(Ω) which is an oracle selection of the "best" list by NDCG@5.

1. D. Sheldon, M. Shokouhi, M. Szummer, and N. Craswell: "LambdaMerge: merging the results of query reformulations." In Proc. WSDM, pages 795–804, 2011.

Deep Structured Learning

- Lee at al. proposed a derivative of LambdaMerge for collection-based fusion using a Deep Neural Network (DNN).
- The key addition was features that capture the quality of verticals – vmScore, vmCo, and VRatio.
- Other features were query-document (RRF, MNZ, Exist, isTopN, Score-based) and query-list (List mean, mean top-*k*, Ratio of MNZ, Ratio of Documents Returned.
- For TREC FedWeb 2013 and 2014 are a bit better than RRF or RankNet / LambdaMART over similar combinations of features.

1. C. J. Lee, Q. Ai, W. B. Croft, and D. Sheldon: “An optimization framework for merging multiple result lists.” In Proc. CIKM, pages 303–312, 2015.

Overview

1. Intro and Overview
2. Theoretical Foundations
3. Fusion in Practice
4. Learning and Fusion
5. **Applications**
6. Conclusions and Future Directions

Diversification

- Diversification is a common task in web search where queries are often imprecise (“`jaguar`”).
- Liang et al. proposed a fusion-based solution for this problem that achieve some of the best-known results on the TREC WebTrack Diversification tasks for diversity-based metrics such as Prec-IA, MAP-IA, α-NDCG, and ERR-IA.
- Their solution was unsupervised and does not require faceted queries to be pre-defined.
- They also show several other variations on the CombX family of fusion methods, all of which improve diversified effectiveness when combined with common diversification methods such as PM-2 [2] and MMR [3].

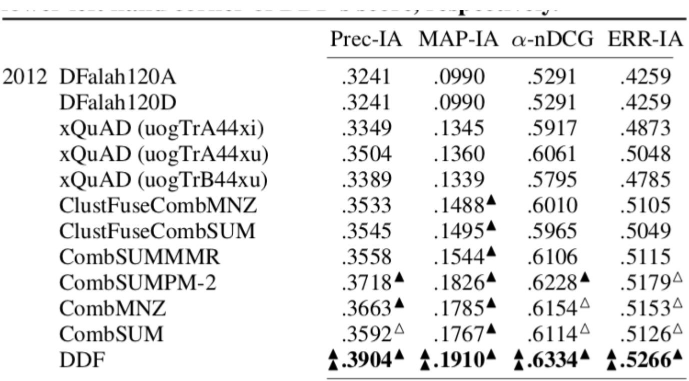
1. S. Liang, Z. Ren, and M. de Rijke: “Fusion helps diversification.” In Proc. SIGIR, pp. 303–312, 2014.
2. V. Dang and W. B. Croft: “Diversity by proportionality: An election-based approach to search result diversification.” In Proc. SIGIR, pp 65–74, 2012.
3. J. Carbonell and J. Goldstein: “The use of MMR, diversity-based reranking for reordering documents and producing summaries.” In Proc. SIGIR, pp 335336, 1998.

Diversification

The algorithm Diversified Data Fusion (DDF) worked in three stages:

1. Use CombSUM on k component runs submitted to TREC.
2. Integrate fusion scores into an LDA topic model to infer a multinomial distribution of facets.
3. Use modification to PM-2 [2] to diversify the results. The key idea was to use fusion scores from CombSUM to compute the aspect probabilities.

Diversification



		Prec-IA	MAP-IA	α-nDCG	ERR-IA
2012	DFalah120A	.3241	.0990	.5291	.4259
	DFalah120D	.3241	.0990	.5291	.4259
	xQuAD (uogTrA44xi)	.3349	.1345	.5917	.4873
	xQuAD (uogTrA44xu)	.3504	.1360	.6061	.5048
	xQuAD (uogTrB44xu)	.3389	.1339	.5795	.4785
	ClustFuseCombMNZ	.3533	.1488▲	.6010	.5105
	ClustFuseCombSUM	.3545	.1495▲	.5965	.5049
	CombSUMMMR	.3558	.1544▲	.6106	.5115
	CombSUMPM-2	.3718▲	.1826▲	.6228▲	.5179△
	CombMNZ	.3663▲	.1785▲	.6154△	.5153△
	CombSUM	.3592△	.1767▲	.6114△	.5126△
	DDF	**▲.3904▲**	**▲.1910▲**	**▲.6334▲**	**▲.5266▲**

Diversified Fusion results for the TREC 2012 Web Track. Reproduced directly from Liang et al [1].

[1] S. Liang, Z. Ren, and M. de Rijke: “Fusion helps diversification.” In Proc. SIGIR, pp. 303–312, 2014.

Expert Search

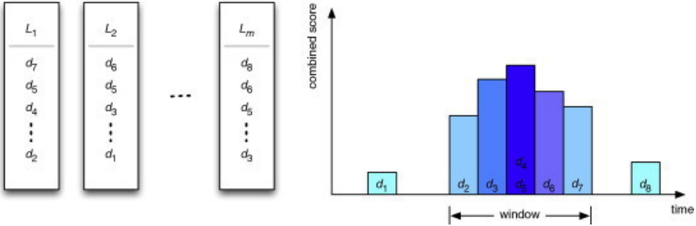
Expert Search

An expert search is a targeted search where a user's information need is a person who has relevant expertise for a specific topic of interest.

- There are normally at least three components in an expert search corpora – queries, documents, and user profiles.
- Macdonald and Ounis [1] showed that RRF and CombX-based fusion techniques can be used to improve expert search effectiveness.
- The key idea is to let each user's expertise implicitly be a set of documents associated to them based on their expertise.
- Now each ranked document returned by retrieval system for a query that is in their "expert" profile is counted as a vote for that document.
- The final fused results can then either be computed by rank position or by renormalized scores.

1. C. Macdonald and I. Ounis: "Voting for candidates: adapting data fusion techniques for an expert search task." In Proc. CIKM, pp 387-396, 2006.

Burst-aware Fusion



Posts that are published in a similar time frame should be promoted in the final list. The *m* ranked lists of posts for a query are on the left. The distribution of the publication timestamps of the documents is on the right, and the vertical axis indicates the combined scores. (Adapted from Liang and de Rijke [1]).

[1] S. Liang and M. de Rijke: "Burst-aware data fusion for microblog search." IPM 51(2): pp 89–113, 2015.

Burst-aware Fusion

Liang and de Rijke [1] propose BurstFuseX to solve this problem, which works in in three stages:

1. Compute the fusion scores using a method such as CombSUM.
2. Detect bursts based on the timestamps and scores.
3. Compute a new fusion score which incorporates three components: $p(d|q)$ (relevance of the document for the query), $p(b|q)$ (how likely a set of posts are relevant to the query), and $p(d|b)$ (how likely the document belongs to the "burst").

$$F(d;q) = (1-\mu)\cdot p(d|q) + \mu \sum_{b \in B} p(d|b)\cdot p(b|q)$$

[1] S. Liang and M. de Rijke: "Burst-aware data fusion for microblog search." IPM 51(2): pp 89–113, 2015.

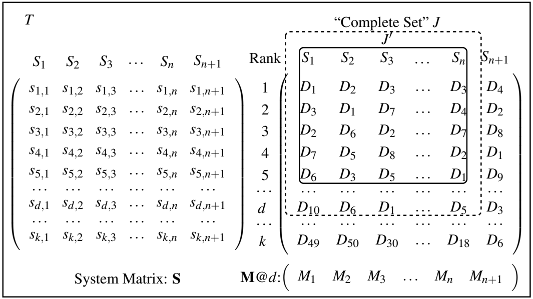
Evaluation

- Most Evaluation campaigns (TREC, NTCIR, CLEF, FIRE) today are based in the Cranfield methodology for collection construction.
    - A large collection of documents.
    - A set of queries, often including a description / narrative of the information need.
    - A set of human relevance judgments (binary or graded) which tell us which documents are relevant in the collection for each query.
- Researchers can then develop a new “system” to test their ideas.
- Once the collection exists, the systems can be compared using some combination of precision and recall-based metrics.

Collection Limitations

- Collection size is increasingly causing problems with offline evaluation.
- If we use a recall-based metric, we must be able to identify every relevant document in the collection for every query.
- If we use a modest sized collection (GOV2), there are 26 million documents.
- For a single person to judge all of the documents for **one** query, it would take more than 9,000 days at a rate of 1 document every 30 seconds, 24 hours a day, 7 days a week.
- There is often a fixed budget available to pay for relevance judgments as well (this seems to be shrinking in today's economy too).

Pooling



T

$$\begin{array}{cccccc} S_1 & S_2 & S_3 & \dots & S_n & S_{n+1} \\ \end{array}$$

$$\left(\begin{array}{cccccc} s_{1,1} & s_{1,2} & s_{1,3} & \dots & s_{1,n} & s_{1,n+1} \\ s_{2,1} & s_{2,2} & s_{2,3} & \dots & s_{2,n} & s_{2,n+1} \\ s_{3,1} & s_{3,2} & s_{3,3} & \dots & s_{3,n} & s_{3,n+1} \\ s_{4,1} & s_{4,2} & s_{4,3} & \dots & s_{4,n} & s_{4,n+1} \\ s_{5,1} & s_{5,2} & s_{5,3} & \dots & s_{5,n} & s_{5,n+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ s_{d,1} & s_{d,2} & s_{d,3} & \dots & s_{d,n} & s_{d,n+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ s_{k,1} & s_{k,2} & s_{k,3} & \dots & s_{k,n} & s_{k,n+1} \end{array}\right)$$

System Matrix: $\mathbf{S}$

"Complete Set" J

J'

$$\begin{array}{c|cccccc} \text{Rank} & S_1 & S_2 & S_3 & \dots & S_n & S_{n+1} \\ \hline 1 & D_1 & D_2 & D_3 & \dots & D_3 & D_4 \\ 2 & D_3 & D_1 & D_7 & \dots & D_4 & D_2 \\ 3 & D_2 & D_6 & D_2 & \dots & D_7 & D_8 \\ 4 & D_7 & D_5 & D_8 & \dots & D_2 & D_1 \\ 5 & D_6 & D_3 & D_5 & \dots & D_1 & D_9 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ d & D_{10} & D_6 & D_1 & \dots & D_5 & D_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ k & D_{49} & D_{50} & D_{30} & \dots & D_{18} & D_6 \end{array}$$

$$\mathbf{M}@d: \begin{pmatrix} M_1 & M_2 & M_3 & \dots & M_n & M_{n+1} \end{pmatrix}$$

To circumvent this problem, Sparck-Jones and van Rijsbergen proposed the idea of pooling. A pool is constructed by collecting the top *k* documents from *n* systems.

[1] J. Spärck Jones and C. J. van Rijsbergen:"Report on the need for and provision of an 'ideal' information retrieval test collection", British Library Research and Development Report 5266, Cambridge, 2018.

Pooling

- Recall the possible effects described by Vogt and Cottrell – chorus, skimming, and dark horse.
- Pooling is cost efficient as many of the best documents will be found by multiple systems.
- Pooling works best when there is diversity in the systems.
- Pool quality can be greatly improved by including manual runs.
- Documents not in the pool are treated as non-relevant when evaluating systems not in the original pool.
- If the size of the collection is tractable, the systems are diverse, and *k* is deep enough, then fixed cutoffs seem to be sufficient (Robust 2004).

Pooling

- Aslam et al. attempted to capture the relationship between fusion (metasearch) and pooling to construct more concentrated documents sets for assessment.
  - Use BordaFuse [1] to order documents for judging. NTCIRPool uses a similar approach.
  - Hedge [2,3] based approach which uses online learning to favour systems that rank the documents judged relevant previously.
- Move-to-Front (MTF) [4] maintains a priority score for each run. The highest priority run is selected, and the highest-ranked, unjudged documents are scored until a non relevant document is found.
- Multi-Armed bandit (reinforcement learning) approaches [5] can also be applied.

[1] J. Aslam and M. Montague: “Models for metasearch.” In Proc. SIGIR, pages 276–284, 2001.

[2] J. Aslam, V. Pavlu, and R. Savell: “A unified model for metasearch, pooling, and system evaluation.” In Proc. CIKM, pages 484–491, 2003.

[3] Y. Freund and R. E. Schapire: “A decision-theoretic generalization of on-line learning and an application to boosting.” JCSS, 55(1):119–139, 1997.

[4] G. Cormack, C. Palmer, and C. Clarke: “Efficient construction of large test collections.” In Proc. SIGIR, pages 282-289, 1998.

[5] D. E. Losada, J. Parapar, and A. Barreiro: “Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems.” IPM, 53(5), 1005-1025, 2017.

Query Performance Prediction

The query performance prediction (QPP) task is to estimate retrieval effectiveness with no relevance judgments (Carmel&Yom Tov '10).

Pre-retrieval predictors utilize information induced from the query and the corpus.

Post-retrieval predictors utilize also information induced from the retrieved list.

Fusion and QPP

- The similarity between the retrieved list at hand and the centroid (i.e., CombSUM fusion) of other retrieved lists was used as a predictor (Aslam&Pavlu '07, Diaz '07, Shtok et al. '16)
  - The idea goes back to Soboroff et al . '01 who evaluated search systems by the similarity of their retrieved lists with a centroid of all retrieved lists
- There is a fundamental formal (and consequently empirical) connection between QPP using a reference list and fusion of the list at hand with the reference list (Shtok et al. '16)

Relevance Feedback

Interactive Fusion (Aslam et al. ’03)

- Using the online learning Hedge algorithm (Freund&Schapire ’97): linear (reciprocal) rank-based fusion
    - At each iteration, a document that would maximize the loss if it were non-relevant is selected
    - A list is penalized based on the number and ranks of non-relevant documents it contains

Utilizing Feedback for the Fused List (Rabinovich&Kurland ’14)

- Relevance feedback is provided for the final fused list
- Feedback is used to (i) create a relevance model and (ii) re-fuse the lists by assigning them infAP/AP weights based on the minimal judgments (feedback)

1. J. Aslam, V. Pavlu, and R. Savell: “A unified model for metasearch, pooling, and system evaluation.” In Proc. CIKM, pages 484–491, 2003.
2. E. Rabinovich, O. Rom and O. Kurland. “Utilizing relevance feedback in fusion-based retrieval”. In Proc. SIGIR, pages 313–322, 2014.

Overview

1. Intro and Overview
2. Theoretical Foundations
3. Fusion in Practice
4. Learning and Fusion
5. Applications
6. Conclusions and Future Directions

Conclusions

- We have focused on the challenge of fusing document lists retrieved in response to a query from the same corpus
  - Lists could be retrieved by using different document representations, query representations and/or ranking functions
- We demonstrated the incredible effectiveness of (simple) fusion approaches
- We surveyed work that tried to explain why and when fusion would be effective
- We discussed a few formal frameworks for fusion
- We presented numerous fusion approaches: supervised vs. unsupervised; rank-based vs. retrieval-score-based
- We discussed various applications for which fusion has been applied: diversification, expert search, evaluation, query performance prediction, relevance feedback

Future Directions

- Developing more rigorous formal frameworks for fusion that can be used for deriving non-linear fusion methods and that will help to explain the conditions for effective fusion
- Predicting (on a per-query basis) whether fusion will be effective
- The list-selection (weighing) challenge: given a few retrieved lists, which subset should be used for fusion? which list weights should be used for weighted linear fusion?
- Selective query expansion (Amati et al. '04, Cronen-Townsend et al. '04)
- Selective cluster-based document retrieval (Liu&Croft '04, Levi et al. '16)
- The optimal cluster question (Kozorovitzky&Kurland '11): finding clusters of similar documents, created from documents across the lists to be fused, that contain a high percentage of relevant documents
- Devising additional non-linear learning-based approaches for fusion
- Predicting which fusion approach will perform best for a given query
- Fusion as an approach for promoting fairness?

Questions?

References I

[1] J. Allen. HARD track overview in TREC 2003: High accuracy retrieval from documents. In *Proc. TREC*, pages 24–37, 2003.

[2] G. Amati, C. Carpineto, and G. Romano. Query difficulty, robustness, and selective application of query expansion. In *Proc. SIGIR*, pages 127–137, 2004.

[3] Y. Anava, A. Shtok, O. Kurland, and E. Rabinovich. A probabilistic fusion framework. In *Proc. CIKM*, pages 1463–1472, 2016.

[4] A. Arampatzis and S. Robertson. Modeling score distributions in information retrieval. *Inf. Retr.*, 14(1):26–46, 2011.

[5] J. A. Aslam and M. Montague. Models for metasearch. In *Proc. SIGIR*, pages 276–284, 2001.

[6] J. A. Aslam and V. Pavlu. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *Proc. ECIR*, pages 198–209, 2007.

[7] J. A. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch and the efficient evaluation of retrieval systems via the hedge algorithm. In *Proc. SIGIR*, pages 393–394, 2003.

References II

[8] J. A. Aslam, V. Pavlu, and E. Yilmaz. Measure-based metasearch. In *Proc. SIGIR*, pages 571–572, 2005.

[9] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. UQV100: A test collection with query variability. In *Proc. SIGIR*, pages 725–728, 2016.

[10] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. Retrieval consistency in the presence of query variations. In *Proc. SIGIR*, pages 395–404, 2017.

[11] N. Balasubramanian and J. Allan. Learning to select rankers. In *Proc. SIGIR*, pages 855–856, 2010.

[12] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, D. A. Grossman, and N. Goharian. Disproving the fusion hypothesis: An analysis of data fusion via effective information retrieval strategies. In *Proc. SAC*, pages 823–827, 2003.

[13] S. M. Beitzel, E. C. Jensen, O. Frieder, A. Chowdhury, and G. Pass. Surrogate scoring for improved metasearch precision. In *Proc. SIGIR*, pages 583–584, 2005.

[14] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. The effect of multiple query representations on information retrieval system performance. In *Proc. SIGIR*, pages 339–346, 1993.

References III

[15] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining evidence of multiple query representation for information retrieval. *Inf. Proc. & Man.*, 31(3):431–448, 1995.

[16] R. Benham and J. S. Culpepper. Risk-reward trade-offs in rank fusion. In *Proc. ADCS*, pages 1:1–1:8, 2017.

[17] R. Benham, J. S. Culpepper, L. Gallagher, X. Lu, and J. Mackenzie. Towards efficient and effective query variation generation. In *Proc. DESIRES*, 2018. To appear.

[18] R. Benham, L. Gallagher, J. Mackenzie, T. T. Damessie, R.-C. Chen, F. Scholer, A. Moffat, and J. S. Culpepper. RMIT at the TREC 2017 CORE Track. In *Proc. TREC*, 2017.

[19] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[20] C. Buckley, D. Dimmick, I. Soboroff, and E. M. Voorhees. Bias and the limits of pooling for large collections. *Inf. Retr.*, pages 491–508, 2007.

[21] C. Buckley and J. Walz. The TREC-8 query track. In *Proc. TREC*, 1999.

[22] C. Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

References IV

[23] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proc. SIGIR*, pages 63–70, 2007.

[24] J. Callan. Distributed information retrieval. In W. Croft, editor, *Advances in information retrieval*, chapter 5, pages 127–150. Kluwer Academic Publishers, 2000.

[25] J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. SIGIR*, pages 335–336, 1998.

[26] D. Carmel and E. Yom-Tov. *Estimating the Query Difficulty for Information Retrieval*. Synthesis lectures on information concepts, retrieval, and services. Morgan & Claypool, 2010.

[27] B. Carterette, V. Pavlu, E. Kanoulas, J. A. Aslam, and J. Allan. Evaluation over thousands of queries. In *Proc. SIGIR*, pages 651–658, 2008.

[28] R.-C. Chen, L. Gallagher, R. Blanco, and J. S. Culpepper. Efficient cost-aware cascade ranking in multi-stage retrieval. In *Proc. SIGIR*, pages 445–454, 2017.

References V

[29] F. M. Choudhury, Z. Bao, J. S. Culpepper, and T. Sellis. Monitoring the top-m rank aggregation of spatial objects in streaming queries. In *Proc. ICDE*, pages 585–596, 2017.

[30] G. V. Cormack, C. L. A. Clarke, and S. Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proc. SIGIR*, pages 758–759, 2009.

[31] G. V. Cormack, C. R. Palmer, and C. L. A. Clarke. Efficient construction of large test collections. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 282–289, 1998.

[32] N. Craswell, D. Hawking, and P. B. Thistlewaite. Merging results from isolated search engines. In *Proc. ADC*, pages 189–200, 1999.

[33] W. B. Croft. Combining approaches to information retrieval. chapter 1, pages 1–36.

[34] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. A language modeling framework for selective query expansion. Technical Report IR-338, Center for Intelligent Information Retrieval, University of Massachusetts, 2004.

References VI

[35] V. Dang and W. B. Croft. Diversity by proportionality: an election-based approach to search result diversification. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 65–74, 2012.

[36] J. C. de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)*, 1784.

[37] T. Diamond. *Information retrieval using dynamic evidence combination*. PhD thesis, Syracuse University, 1998. unpublished.

[38] F. Diaz. Regularizing query-based retrieval scores. *Inf. Retr.*, 10(6):531–562, 2007.

[39] B. T. Dinçer, C. Macdonald, and I. Ounis. Risk-sensitive evaluation and learning to rank using multiple baselines. In *Proc. SIGIR*, pages 483–492, 2016.

[40] B. T. Dinçer, C. Macdonald, and I. Ounis. Hypothesis testing for the risk-sensitive evaluation of retrieval systems. In *Proc. SIGIR*, pages 23–32, 2014.

[41] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proc. WWW*, pages 613–622, 2001.

References VII

[42] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proc. TREC*, 1994.

[43] H. D. Frank and I. Taksa. Comparing rank and score combination methods for data fusion in information retrieval. *Inf. Retr.*, 8(3):449–480, 2005.

[44] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[45] L. Gallagher, J. Mackenzie, R. Benham, R.-C. Chen, F. Scholer, and J. S. Culpepper. RMIT at the NTCIR-13 We Want Web task. In *Proc. NTCIR*, 2017.

[46] N. P. Gopalan and K. Batri. Adaptive selection of top-*m* retrieval strategies for data fusion in information retrieval. *Intl. J. of Soft Computing*, 2(1), 2007.

[47] A. Griffiths, H. C. Luckhurst, and P. Willett. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science*, 37(1):3–11, 1986.

[48] S. Huo, M. Zhang, Y. Liu, and S. Ma. Improving tail query performance by fusion model. In *Proc. CIKM*, pages 559–658, 2014.

[49] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217–240, 1971.

References VIII

[50] K. Jones, C. Van Rijsbergen, B. L. Research, and D. Department. *Report on the Need for and Provision of an Ideal Information Retrieval Test Collection*. 1975.

[51] A. Juárez-González, M. Montes-y-Gómez, L. V. Pineda, and D. O. Arroyo. On the selection of the best retrieval result per query - an alternative approach to data fusion. In *Proc. FQAS*, pages 111–121, 2009.

[52] A. Juárez-González, M. Montes-y-Gómez, L. V. Pineda, D. P. Avendaño, and M. A. Pérez-Coutiño. Selecting the n-top retrieval result lists for an effective data fusion. In *Proc. CICLing*, pages 580–589, 2010.

[53] J. Katzer, M. McGill, J. Tessier, W. Frakes, , and P. Daegupta. A study of the overlap among document represent ations. *Information Technology: Research and Development*, 1:261, 1982.

[54] J. Kemeny. Mathematics without numbers. *Daedalus*, 88, 1959.

[55] Y. Kim, J. Callan, J. S. Culpepper, and A. Moffat. Efficient distributed selective search. *Inf. Retr.*, 20(3):221–252, 2017.

[56] A. K. Kozorovitzky and O. Kurland. From ”identical” to ”similar”: Fusing retrieved lists based on inter-document similarities. In *Proc. ICTIR*, pages 212–223, 2009.

References IX

[57] A. K. Kozorovitzky and O. Kurland. Cluster-based fusion of retrieved lists. In *Proc. SIGIR*, pages 893–902, 2011.

[58] A. K. Kozorovitzky and O. Kurland. From ”identical” to ”similar”: Fusing retrieved lists based on inter-document similarities. *J. of AI Res.*, 41, 2011.

[59] M. Lalmas. A formal model for data fusion. In *Proc. FQAS*, pages 274–288, 2002.

[60] S. Lawrence and C. L. Giles. Inquirus, the NECI meta search engine. *Computer Networks*, 30(1-7):95–105, 1998.

[61] C. Lee, Q. Ai, W. B. Croft, and D. Sheldon. An optimization framework for merging multiple result lists. In *Proc. CIKM*, pages 303–312, 2015.

[62] J. H. Lee. Analyses of multiple evidence combination. In *Proc. SIGIR*, pages 267–276, 1997.

[63] O. Levi, F. Raiber, O. Kurland, and I. Guy. Selective cluster-based document retrieval. In *Proc. CIKM*, pages 1473–1482, 2016.

[64] S. Liang and M. de Rijke. Burst-aware data fusion for microblog search. *Inf. Proc. & Man.*, 51(2):89–113, 2015.

[65] S. Liang and M. de Rijke. Burst-aware data fusion for microblog search. *Inf. Proc. & Man.*, 51(2):89–113, 2015.

References X

[66] S. Liang, M. de Rijke, and M. Tsagkias. Late data fusion for microblog search. In *Proc. ECIR*, pages 743–746, 2013.

[67] S. Liang, I. Markov, Z. Ren, and M. de Rijke. Manifold learning for rank aggregation. In *Proc. WWW*, pages 1735–1744, 2018.

[68] S. Liang, Z. Ren, and M. de Rijke. Fusion helps diversification. In *Proc. SIGIR*, pages 303–312, 2014.

[69] S. Liang, Z. Ren, and M. de Rijke. The impact of semantic document expansion on cluster-based fusion for microblog search. In *Proc. ECIR*, pages 493–499, 2014.

[70] D. Lillis, F. Toolan, R. W. Collier, and J. Dunnion. Probfuse: a probabilistic approach to data fusion. In *Proc. SIGIR*, pages 139–146, 2006.

[71] D. Lillis, F. Toolan, R. W. Collier, and J. Dunnion. Extending probabilistic data fusion using sliding windows. In *Proc. ECIR*, pages 358–369, 2008.

[72] D. Lillis, L. Zhang, F. Toolan, R. W. Collier, D. Leonard, and J. Dunnion. Estimating probabilities for effective data fusion. In *Proc. SIGIR*, pages 347–354, 2010.

References XI

[73] D. E. Losada, J. Parapar, and A. Barreiro. Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Inf. Proc. & Man.*, 53(5):1005–1025, 2017.

[74] X. Lu, A. Moffat, and J. S. Culpepper. The effect of pooling and evaluation depth on IR metrics. *Inf. Retr.*, 19(4):416–445, 2016.

[75] X. Lu, A. Moffat, and J. S. Culpepper. Modeling relevance as a function of retrieval rank. In *Proc. AIRS*, pages 3–15, 2016.

[76] X. Lu, A. Moffat, and J. S. Culpepper. Can deep effectiveness metrics be evaluated using shallow judgment pools? In *Proc. SIGIR*, pages 35–44, 2017.

[77] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006*, pages 387–396, 2006.

[78] J. Mackenzie, F. M. Choudhury, and J. S. Culpepper. Efficient location-aware web search. In *Proc. ADCS*, pages 4.1–4.8, 2015.

References XII

[79] R. Manmatha, T. M. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pages 267–275, 2001.

[80] I. Markov, A. Arampatzis, and F. Crestani. Unsupervised linear score normalization revisited. In *Proc. SIGIR*, pages 1161–1162, 2012.

[81] G. Markovits, A. Shtok, O. Kurland, and D. Carmel. Predicting query performance for fusion-based retrieval. In *Proc. CIKM*, 2012.

[82] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *Proc. CIKM*, pages 538–548, 2002.

[83] M. H. Montague and J. A. Aslam. Relevance score normalization for metasearch. In *Proc. CIKM*, pages 427–433, 2001.

[84] A. Mourao, F. Martins, and J. Magalhaes. Inverse square rank fusion for multimodal search. In *Proc. CBMI*, pages 1–6, 2014.

[85] K. B. Ng and P. P. Kantor. An investigation of the preconditions for effective data fusion in information retrieval: A pilot study, 1998.

References XIII

[86] D. Parikh and R. Polikar. An ensemble-based incremental learning approach to data fusion. *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):437–450, 2007.

[87] T. Qin, X. Geng, and T. Liu. A new probabilistic model for rank aggregation. In *Proc. NIPS*, pages 1948–1956, 2010.

[88] E. Rabinovich, O. Rom, and O. Kurland. Utilizing relevance feedback in fusion-based retrieval. In *Proc. SIGIR*, pages 313–322, 2014.

[89] F. Radlinski and N. Craswell. A theoretical framework for conversational search. pages 117–126, 2017.

[90] F. Raiber and O. Kurland. Query-performance prediction: setting the expectations straight. In *Proc. SIGIR*, pages 13–22, 2014.

[91] M. E. Renda and U. Straccia. Web metasearch: Rank vs. score based rank aggregation methods. In *Proc. SAC*, pages 841–846, 2003.

[92] S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, pages 294–304, 1977. Reprinted in K. Sparck Jones and P. Willett (eds), *Readings in Information Retrieval*, pp. 281–286, 1997.

[93] E. H. Ruspini. The logical foundations of evidential reasoning. Technical report, SRI International, 1986.

References XIV

[94] M. Sanderson. Test collection based evaluation of information retrieval systems. *Found. Trends in Inf. Ret.*, 4(4):247–375, 2010.

[95] D. Sheldon, M. Shokouhi, M. Szummer, and N. Craswell. LambdaMerge: Merging the results of query reformulations. In *Proc. WSDM*, pages 795–804, 2011.

[96] M. Shokouhi. Segmentation of search engine results for effective data-fusion. In *Proc. ECIR*, pages 185–197, 2007.

[97] M. Shokouhi and L. Si. Federated search. *Found. Trends in Inf. Ret.*, 5(1):1–102, 2011.

[98] X. M. Shou and M. Sanderson. Experiments on data fusion using headline information. In *Proc. SIGIR*, pages 413–414, 2002.

[99] A. Shtok, O. Kurland, and D. Carmel. Query performance prediction using reference lists. *ACM Trans. Inf. Sys.*, 34(4):19:1–19:34, 2016.

[100] M. Truchon. An extension of the condorcet criterion and kemeny orders. *conomie et Finance Appliqueés*, 1998.

[101] T. Tsikrika and M. Lalmas. Merging techniques for performing data fusion on the Web. In *Proc. CIKM*, pages 127–134, 2001.

References XV

[102] K. Tumer and J. Ghosh. Linear and order statistics combiners for pattern classification. *CoRR*, cs.NE/9905012, 1999.

[103] H. R. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.*, 9(3):187–222, 1991.

[104] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, second edition, 1979.

[105] C. C. Vogt. How much more is better? Characterising the effects of adding more IR systems to a combination. In *Proc. RIAO*, pages 457–475, 2000.

[106] C. C. Vogt and G. W. Cottrell. Predicting the performance of linearly combined IR systems. In *Proc. SIGIR*, pages 190–196, 1998.

[107] C. C. Vogt and G. W. Cottrell. Fusion via linear combination of scores. *Inf. Retr.*, 1(3):151–173, 1999.

[108] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. The collection fusion problem. In *Proc. TREC*, 1994.

[109] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press, 2005.

[110] E. M. Voorhees and D. K. Harman. *TREC: Experiments and evaluation in information retrieval*. The MIT Press, 2005.

References XVI

[111] W. Webber, A. Moffat, and J. Zobel. The effect of pooling and evaluation depth on metric stability. In *Proc. EVIA*, pages 7–15, 2010.

[112] S. Wu. Applying statistical principles to data fusion in information retrieval. *Expert Syst. Appl.*, 36(2):2997–3006, 2009.

[113] S. Wu and F. Crestani. A geometric framework for data fusion in information retrieval. *Inf. Syst.*, 50:20–35, 2015.

[114] S. Wu, F. Crestani, and Y. Bi. Evaluating score normalization methods in data fusion. In *Proc. AIRS*, pages 642–648, 2006.

[115] S. Wu and C. Huang. Search result diversification via data fusion. In *Proc. SIGIR*, pages 827–830, 2014.

[116] M. Yasukawa, J. S. Culpepper, and F. Scholer. Data fusion for Japanese term and character *n*-gram search. In *Proc. ADCS*, pages 10.1–10.4, 2015.

[117] H. P. Young. Condorcet's theory of voting. *American Political Science Review*, 82(4):1231–1244, 1988.

[118] K. Zhou, X. Li, and H. Zha. Collaborative ranking: improving the relevance for tail queries. In *Proc. CIKM*, pages 1900–1904, 2012.