
CondorcetFuse

Condoret voting for run fusion

Evalutation and comparison of an implementation with other fusion strategies

Contents

1	Introduction	2
2	Condorcet fusion	2
2.1	The Condorcet Graph	3
2.2	Condorcet paths	3
3	Implementation	3
4	Evaluation	3
4.1	First Analysis	4
4.1.1	Results Analysis	4
4.2	Further Analysis	4
4.2.1	Analysis over increasing number of inputs	4
4.2.2	Analysis on the indexing and retrieval settings	5
4.3	Results	7
5	Conclusions	7

1 Introduction

In Information Retrieval, data fusion is the combination of the results of independent searches on a document collection into one single output result set.

It has been shown in the past that this can greatly improve retrieval effectiveness over that of the individual results.

The aim of this work is to show a possible implementation of some basic fusion strategies and compare them to an advanced one: Condorcet fusion [3].

In this project the input documents were taken from the TREC TIPSTER Collection using 50 topics (topics 351-400). We used Terrier for retrieval and indexing and Matters for evaluation.

We organized our work as follows:

- **Indexing:** we created four different indexes:
 - Without both stemmer and stop list;
 - only using the Porter stemmer;
 - only using the stop list;
 - using both the Porter stemmer and the stop list;
- **Retrieval:** 10 different retrieval models listed in Table 1;
- **Normalization:** min/max normalization as presented by Lee [1]
- **Fusion strategy:** we compared seven different fusion strategies:
 - 6 basic strategies proposed by Fox and Shawn [2] and listed in Table 2;
 - Condorcet fusion (advanced strategy) as proposed by Montague and Aslam [3]

Retrieval models	Basic fusion methods	New score formula
BB2	CombMNZ	$(\sum_i^N s_i) * (Num\ s_i \neq 0)$
BM25		
DLH13	CombSUM	$\sum_i^N s_i$
Hiemstra_LM		
IFB2	CombMIN	$min_i^N(s_i)$
TF_IDF		
DFIC	CombMAX	$min_i^N(s_i)$
DFIZ		
DirichletLM	CombMED	$median_i^N(s_i)$
InL2		
	CombANZ	$(\sum_i^N s_i) / (Num\ s_i \neq 0)$

Table 1: Retrieval models used

Table 2: Basic fusion methods used

2 Condorcet fusion

The Condorcet Fusion strategy considers the ranking of documents from different systems as an instance of the voting problem where documents are candidates and each input retrieval system is a voter.

The output of each input system is seen as a list of preferences, where the higher ranked document beats the lower ranked ones.

Having the lists of different preferences, Condorcet Fusion orders the documents using the Condorcet voting algorithm, a majoritarian method which specifies that the “winner” of the fusion is the document(s) that beats or ties with every other document in a pair-wise comparison between the input systems (i.e. runs).

2.1 The Condorcet Graph

The Condorcet Graph is used to determine the Condorcet winner.

Given 10 models of retrieval with n documents, the corresponding Condorcet graph $G = (V, E)$ has one vertex for each of the n documents.

For each document pair (x, y) , there exists an edge from x to y (denoted by $x \rightarrow y$) if x would receive at least as many votes as y in a head-to-head contest.

Cycles can simply be viewed as ties.

2.2 Condorcet paths

A Condorcet-consistent hamiltonian path (or condorcet path) is any hamiltonian path through the Condorcet graph.

The goal of Condorcet Fusion is to efficiently find such a path and to output the documents in the order they are visited as nodes of the path.

3 Implementation

The main project structure is the following:

- **eval/** Contains evaluation scripts and results;
- **results/** Contains input and output runs for fusion;
- **scripts/** Contains indexing and retrieval scripts;
- **src/** Main directory of the application, contains Java source code;
 - **Base/** Contains definition for run, runset and min/max normalization;
 - **Fusion/** Contains implementation of basic fusion methods and Condorcet;
 - **IO/** Manages the input and output of runs;
 - **Application** - the main class;

The implementation of Condorcet uses quicksort, with the following algorithm as comparing function:

```
count = 0
for each of the k search systems do:
    if sys i ranks d1 above d2, count++
    if sys i ranks d2 above d1, count--
    if count > 0, rank d1 better than d2
    else rank d2 better than d1
```

4 Evaluation

The evaluation criteria based on the given pool uses a binary relevance score: Relevant and Non-Relevant. The documents left out from the pool are considered to be Non-Relevant. We decided to use Average Precision to evaluate a run on the set of topics and Mean Average Precision to evaluate between topics. We always work on normalized data.

We first did an analysis considering the 10 models presented in Table 1 running retrieval and indexing with Terrier's default settings (that is, using both the Porter Stemmer and a stop list) and the fusion's given these 10 input systems.

Then, since the paper on Condorcet Fuse performed some analysis to understand the relation between CondFuse's MAP and the number of input systems, we also tried something similar.

4.1 First Analysis

At first, we wanted to see if using a fusion strategy actually improves the best performance of a simple model run with both the stemmer (we used Porter Stemmer) and a stop list. We proceeded as follows:

1. We computed the AP and the MAP for the 10 input models;
2. We computed the AP and the MAP for the 7 fusion models (6 basic and Condorcet);
3. We selected the 5 systems with the best MAP values.

Figure 1 shows the Average Precision for the top 5 best systems, while Table 3 shows the MAP for all of them.

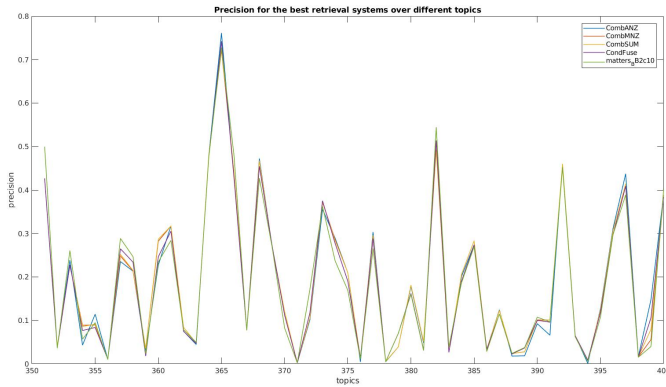


Figure 1: AP over 50 topics for the 5 best systems

Systems	MAP
CombSUM	0.1909
CombMNZ	0.1902
CombANZ	0.1891
CondFuse	0.1883
BB2	0.1881
IFB2	0.1880
DirichletLM	0.1862
InL2	0.1853
CombMED	0.1840
DLH13	0.1829
BM25	0.1827
TF_IDF	0.1821
CombMAX	0.1817
DFIZ	0.1783
DFIC	0.1758
Hiemstra_LM	0.1733
CombMIN	0.1515

Table 3: Retrieval systems sorted by decreasing MAP

4.1.1 Results Analysis

The results of these first tests show that it is generally convenient to use a fusion method. Four out of the top five best systems were fusion systems, with Condorcet Fuse being the fourth best overall.

But, we also noted that the performances of the methods are all very close, and the best and worse topics are the same regardless of the system used.

4.2 Further Analysis

We decided we wanted to test how the performances of the 10 input systems affected the performance of the fusion methods. Furthermore, since Montague and Aslan [3] analyzed the algorithm's performance over the number of input systems, we also tried fusions starting with 2 input systems up to 10.

4.2.1 Analysis over increasing number of inputs

Figure 2 shows a comparison between the MAP for the 7 fusion methods over the number of input systems and with different indexing and retrieval settings.

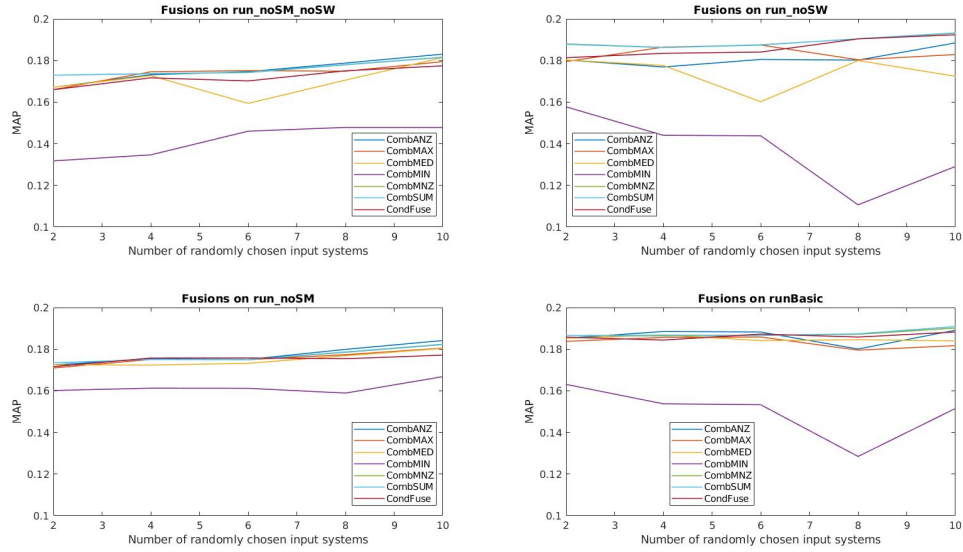


Figure 2: MAP depending in the number of input system and indexing and retrieval settings

We noted that the number of input system doesn't much affect the performances of the fusion methods. The main difference was given by the indexing and retrieval settings. In particular, having neither the stemmer or the stoplist was similar to having just the stop list, while having both stemmer and stoplist was more similar to just having the stemmer.

We expected the stop list to be more defining than the stemmer, so we performed some further analysis focusing on what affected more the fusions.

4.2.2 Analysis on the indexing and retrieval settings

We wanted to see how the indexing and retrieval settings affect the fusions' performances. Furthermore, we wanted to understand the relationship between the models' and the fusions' performances. Since generally the best fusions were the ones performed over 10 input systems, we just focused on them.

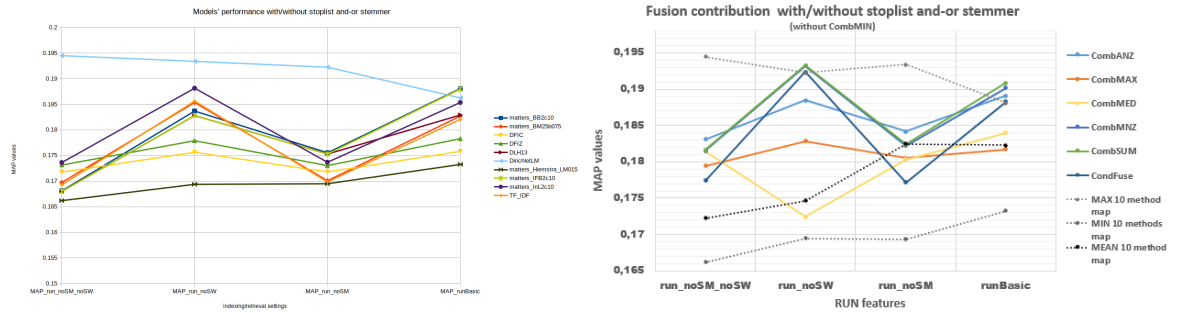


Figure 3: MAP for the retrieval models over different indexing and retrieval settings

Figure 4: MAP for the fusions over different indexing and retrieval settings

4.2.3 Analysis on the performance's improvement using fusions

Grafici che mostrano i miglioramenti/peggioramenti

5 Conclusions

References

- [1] Lee, Joon Ho. "Combining multiple evidence from different properties of weighting schemes." Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1995.
- [2] Fox, Edward A., and Joseph A. Shaw. "Combination of multiple searches." NIST special publication SP 243 (1994).
- [3] Montague, Mark, and Javed A. Aslam. "Condorcet fusion for improved retrieval." Proceedings of the eleventh international conference on Information and knowledge management. ACM, 2002.