

Condorcet Fusion: an implementation

Improved Retrieval through Rank Fusion

Marco Alfonso, Davide Martini, Giovanni Mazzocchin
February 9, 2018



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

This project implements some ideas coming from a paper by **Aslam** and **Montague**.

These authors applied *Social Choice Theory* to **Rank Fusion**.

They claim that their algorithm beats the ones by **Fox** and **Shaw** most of the time.

Fox and **Shaw** devised some simple, score-based fusion formulae:

CombMAX

$$\text{fused score} = \max(\text{scores})$$

CombMIN

$$\text{fused score} = \min(\text{scores})$$

CombSUM

$$\text{fused score} = \text{sum}(\text{scores})$$

Fox and **Shaw** devised some simple, score-based fusion formulae:

CombMED

$$\text{fused score} = \text{median}(\text{scores})$$

CombANZ

$$\text{fused score} = \frac{\text{CombSUM}}{\#nonZeroScores}$$

CombMNZ

$$\text{fused score} = \text{CombSUM} \times \#nonZeroScores$$

Are we comparing apples and oranges?



Of course we can't perform any sensible fusion without **normalization**.

Lee came up with two handy formulae:

Max Norm

$$\text{max_norm} = \frac{\text{old_score}}{\text{max_score}}$$

Min Max Norm

$$\text{min_max_norm} = \frac{\text{old_score} - \text{min_score}}{\text{max_score} - \text{min_score}}$$

What about voting systems?



Metaphor: *voters are retrieval systems*
candidates are documents

The algorithm inspired by the **Condorcet method** starts from a document comparator known as *Simple Majority Runoff*.

“Simple Majority Runoff”



Imagine a **contest** between two documents, then the rule below sounds sensible, doesn't it?:

the more times one of the documents ranks above the other one, the more votes it gets.