

---

# Discovery of ranking functions for Information Retrieval based on Genetic Programming

**Humberto Mossri de Almeida**

**Advisor: Marcos André Gonçalves**

**DCC/UFMG, Brazil, 2007**

---

## Introduction

- Information Retrieval Systems
  - Use ranking functions to sort a set of answer documents to a query based on their estimated relevance to a user's information need
  
- Ranking functions
  - Play a fundamental role in the performance of information retrieval systems

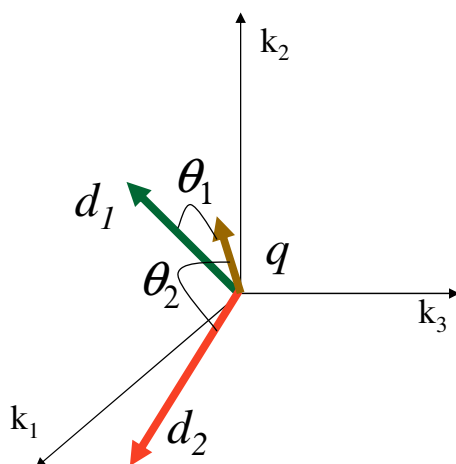
# Introduction

- Ranking functions (Trotman, 2004)
  - Represent the sum of influences of each term  $t$  of a query  $q$  to a document  $d$
  - Are generically expressed as follows

$$W_{dq} = \sum_{t \in q} g(t, d)$$

where  $w_{dq}$  is the weight of a document  $d$  with respect to a query  $q$

# Vector Space Model



$$sim(q, d_j) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{k_i \in q} w_{i,j} \times w_{i,q}}{|\vec{d}_j| \times |\vec{q}|}$$

$$w_{i,j} = 1 + \log(tf_{i,j}) \times \log\left(\frac{N}{n_i}\right)$$

$$w_{i,q} = 0.5 + \frac{0.5 \times tf_{i,q}}{\max tf_{i,q}} \times \log\left(\frac{N}{n_i}\right)$$

$$|\vec{d}_j| = \sqrt{\sum_{i=1}^t w_{i,j}^2}$$

## BM25 (Okapi System)

$$\text{sim}(q, d_j) = \vec{d}_j \bullet \vec{q} = \sum_{k_i \in q} w^{(1)} \times w_{i,j} \times w_{i,q} + k_2 \times \frac{|\vec{d}_j| - |\vec{d}_j|}{|\vec{d}_j| + |\vec{d}_j|}$$

$$w^{(1)} = \log\left(\frac{(r+0.5)/(R-r+0.5)}{(n_i-r+0.5)/(N-n_i-R+r+0.5)}\right)$$

$$w_{i,q} = \frac{(k_3+1) \times \text{tf}_{i,q}}{\text{tf}_{i,j} + k_3}$$

$$w_{i,j} = \frac{k_1 \times \text{tf}_{i,j}}{\text{tf}_{i,j} + k_1 \times \left(1 - b + b \times \frac{|\vec{d}_j|}{|\vec{d}_j|}\right)} \times \log\left(\frac{N - n_i + 0.5}{0.5}\right)$$

$$k_1 = 1.2, k_2 = 0, k_3 = 1000, b = 0.75, r = 0, R = 0$$

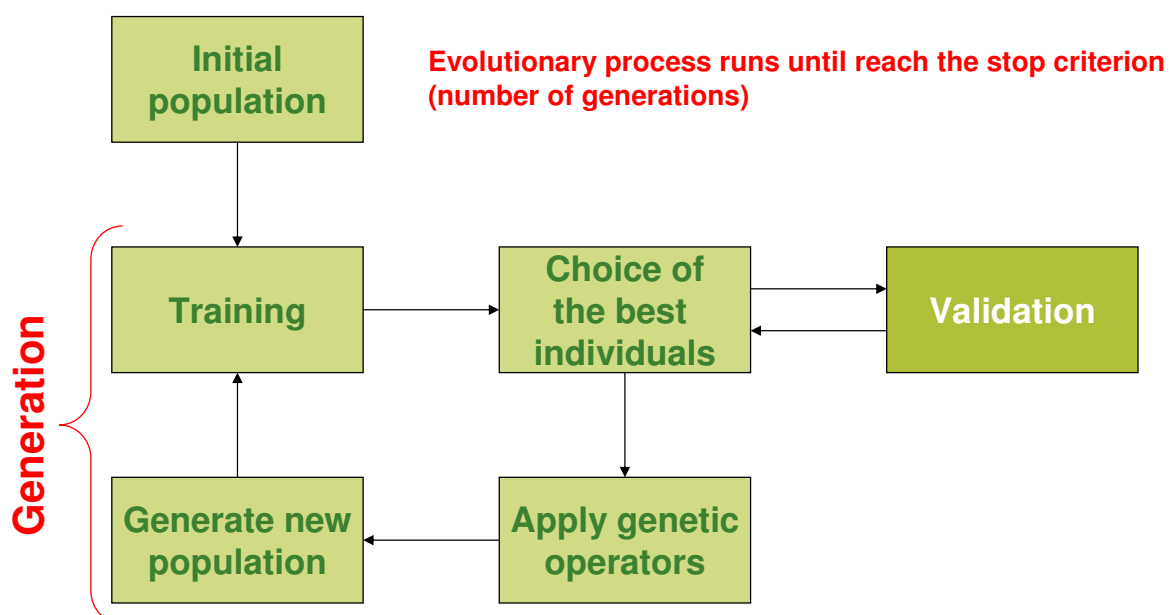
## Motivation

- Thousands of ranking functions have already been proposed and studied. However, no ranking function is consistently good in all collections or domains (Zobel e Moffat, 1998)
- Several works show that ranking functions present inconsistent behavior when applied in different contexts (e.g., collections, queries) (Fan, Gordon e Pathak, 2003)

## Our proposal

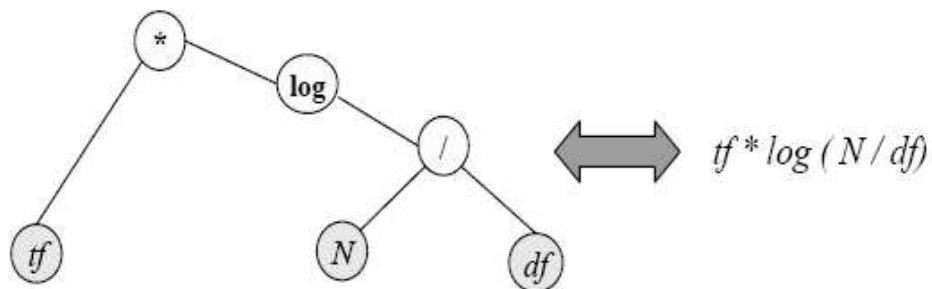
- Generate specific, better tuned, ranking functions to a document collection based on a Genetic Programming framework

## Genetic Programming (GP)



# GP Framework

- Individuals
  - Represent a ranking function



# GP Framework

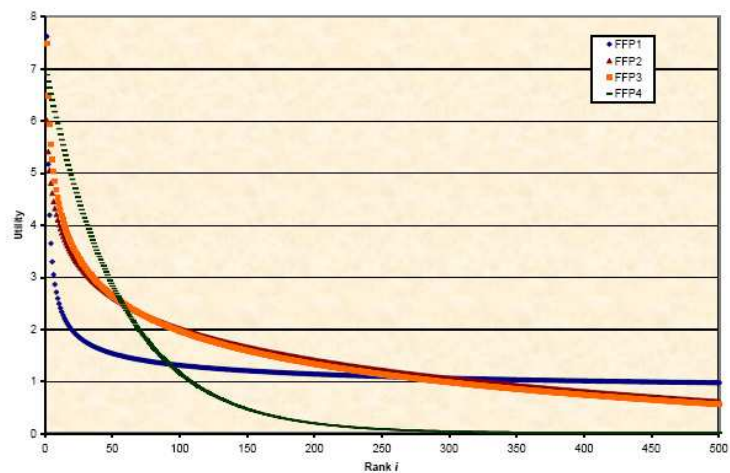
- Set of terminals
  - Characteristics of the collection, of the documents, of the terms, and of the queries
    - Number of documents in the collection
    - Term frequency
    - ...
- Set of functions
  - Combine terminals, and sub-trees
    - +, -, /, \*
    - log, sqrt, exp
    - ...

# GP Framework

## ■ Fitness function

- Evaluation function that defines the effectiveness of a ranking function represented by an individual

- Mean average precision (MAP,  $P\_AVG$ )
- Utility functions (FFP1, FFP2, FFP3, FFP4)
- R-precision
- $P@5$
- $P@10$



# GP Framework

## ■ Parameters

- Size of the initial population
- Genetic operations
- Maximum size of the individuals
  - Depth of the trees

## ■ Stop criterion

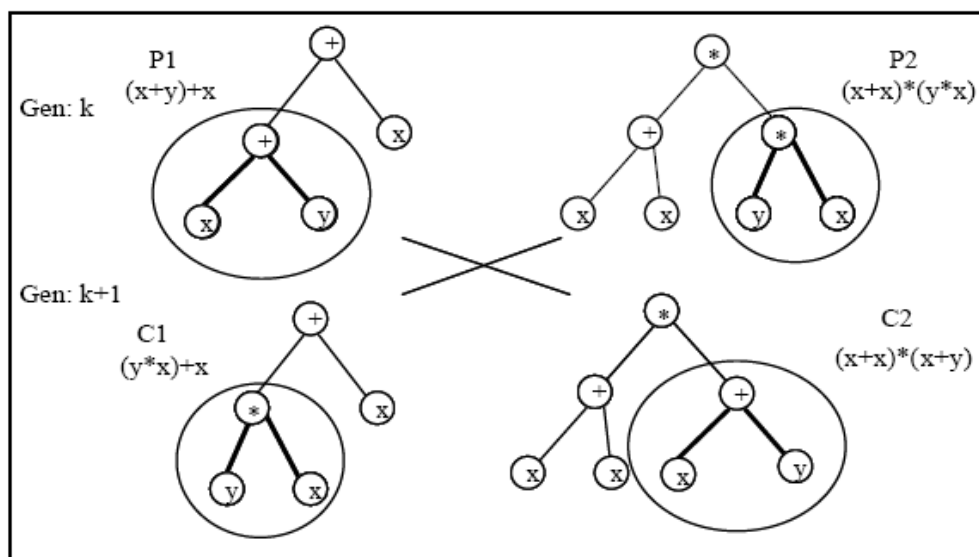
- Maximum number of generations

# GP Framework

- Choice of the best individual
  - The best in the training set
  - The best in the validation set
  - The best average between training and validation sets
  - The best sum of training and validation sets
  - The best average between training and validation sets minus standard deviation
  - The best sum of training and validation sets minus standard deviation

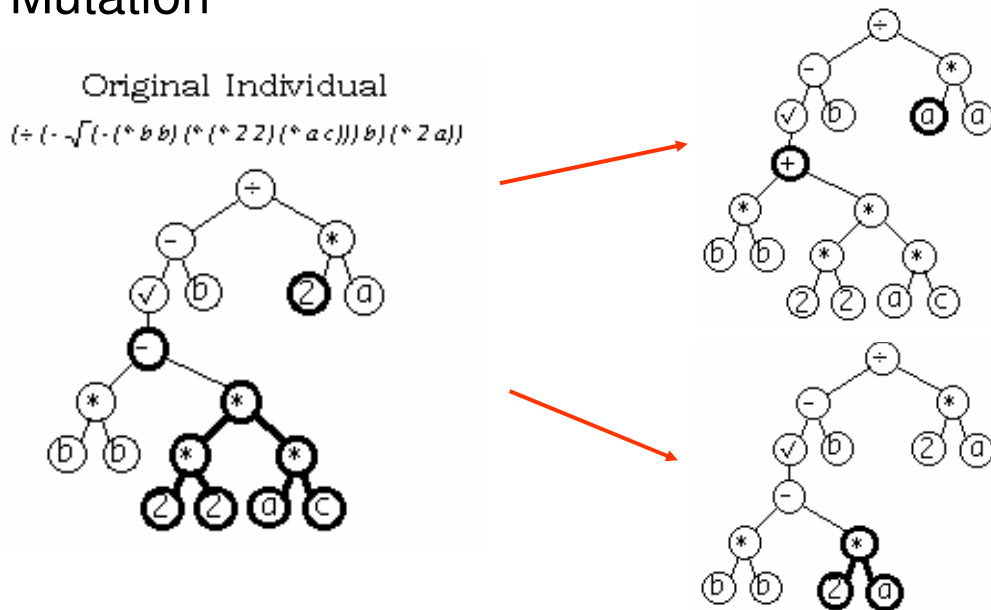
# GP Framework

## ■ Crossover



# GP Framework

## ■ Mutation



## Related work

- Based on the use and combination of statistical evidence of the collection, documents, and queries
  - Fan, Gordon e Pathak (2003, 2004, 2005)
    - tf, tf\_max, tf\_avg, tf\_max\_col, df, df\_max, N, length, length\_avg, n
  - Trotman (2004)
    - Query: length, vector\_length, ...
    - Each search term: df, tf, tf\_query, ...
    - Each document: length, vector\_length, unique\_terms, ...
    - Collection: N, length\_max, tf\_max, df\_max, ...
    - Add baselines to the initial population

---

## Our research assumption

- Is it possible to improve the information retrieval task results using meaningful, rich, and proven components extracted from well-known ranking functions?

---

## CCA - Combined Component Approach

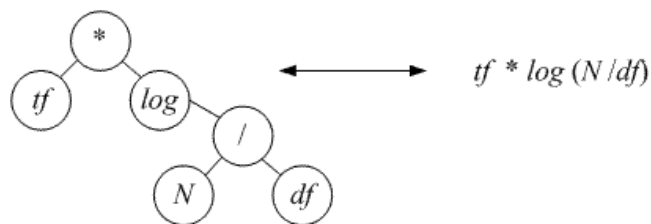
- Use meaningful components extracted from well-known effective ranking functions
- Components
  - Background
    - Term-weighting components (Salton and Buckley, 1988)
    - Exploring the Similarity Space (Zobel and Moffat, 1998)
  - Components
    - Term Frequency Component
    - Collection Frequency Component
    - Normalization Component
  - Thousands of different alternatives

# CCA

- A typical CCA individual



- A typical individual based on statistical information



# CCA Terminals

$tf$ <b>t01</b>	$\log(tf) + 1$ <b>t02</b>	$0.5 + \frac{0.5 + tf}{\max tf}$ <b>t03</b>	$\frac{\log(tf) + 1}{\log(\text{avg}tf) + 1}$ <b>t04</b>
$\frac{(k_1 + 1) \times tf}{(k_1 \times ((1 - b) + b \times dl / \text{avg}dl) + tf)}$ <b>t05</b>	$\log\left(\frac{N}{df}\right)$ <b>t06</b>	$\log\left(\frac{N}{df} + 1\right)$ <b>t07</b>	
$\log\left(\frac{N - df + 0.5}{0.5}\right)$ <b>t08</b>	$w^{(1)} = \log\left(\frac{N - df + 0.5}{df + 0.5}\right)$ <b>t09</b>	$\log\left(\frac{N - df}{df}\right)$ <b>t10</b>	
$\frac{\log((N + 0.5)/df)}{\log(N + 1)}$ <b>t11</b>	$\frac{1}{\sqrt{\sum_{i=1}^l w_{i,j}^2}}$ , onde $w_{i,j}^2$ é $tf \times \log(N/df + 1)$ <b>t12</b>	$dl$ (bytes) <b>t14</b>	
$tfidfnorm = \frac{1}{\sqrt{\sum_{i=1}^l w_{i,j}^2}}$ , onde $w_{i,j}^2$ é $1 + \log(tf) \times \log(N/df + 1)$ <b>t13</b>	$\frac{(k_3 + 1) \times qtf}{k_3 + qtf}$ <b>t19</b>		
$\frac{1}{(k_1 \times ((1 - b) + b \times dl / \text{avg}dl) + tf)}$ <b>t18</b>	$\frac{1}{(1 - slope) \times \text{avg}dl + (slope \times dl)}$ <b>t16</b>		
$\frac{1}{((1 - slope) + slope \times tfidfnorm / \text{avg}tfidfnorm)}$ <b>t15</b>	$\frac{0.5 + (0.5 \times qtf)}{\max qtf}$ <b>t20</b>		
$\frac{1}{((1 - slope) \times pivot + slope \times \# \text{ of unique terms})}$ <b>t17</b>			

---

# Collections

Characteristics	Collection	
	<i>TREC-8</i>	<i>WBR-99</i>
Number of documents	528,155	5,939,061
Number of distinct terms	737,833	2,669,965
Number of topics (queries)	50 (401-450)	49
Query size	10.80	1.88
Collection size	2 (GB)	16 (GB)

---

# Experiments

- Number of generations
  - 30
- Genetic operators
  - Crossover: 0.9
  - Mutation: 0.05
  - Reproduction: 0.05
- Maximum depth
  - Range from 3 to 12
- Initial population
  - 200
- Functions
  - \* / + log

---

# Experiments

Number of queries	Collection	
	<i>TREC-8</i>	<i>WBR-99</i>
Training set	20	20
Validation set	10	10
Test set	20	19

---

# Experiments

- Fitness function
  - MAP (P\_AVG)
  - FFP4
- Choice of the best individual
  - The best average between training and validation sets minus standard deviation
  - The best sum of training and validation sets minus standard deviation

# Baselines

## ■ TREC

- BM25 (Robertson et al, 1995)
- FAN-GP (Fan et al, 2003)

## ■ WBR-99

- TF-IDF (Salton et al, 1988)
- FAN-GP (Fan et al, 2003)

# TREC: Results

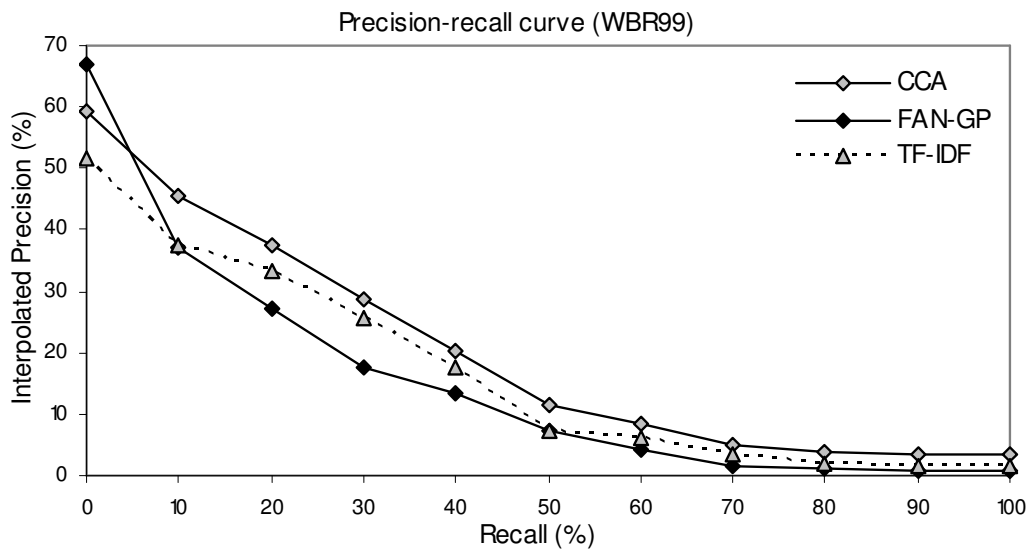
Interpolated precision	BM25	FAN-GP	CCA_D5	Gain over BM25	Gain over FAN-GP	CCA_D7	Gain over BM25	Gain over FAN-GP
At 5 docs	27,000	27,000	32,000	18,52%	18,52%	34,000	25,93%	25,93%
At 10 docs	29,000	25,500	31,500	8,62%	23,53%	30,500	5,17%	19,61%
At 15 docs	25,333	24,667	27,000	6,58%	9,46%	27,000	6,58%	9,46%
At 20 docs	23,750	22,750	25,000	5,26%	9,89%	26,000	9,47%	14,29%
At 30 docs	20,167	21,333	22,167	9,92%	3,91%	22,333	10,74%	4,69%
At 100 docs	15,050	15,150	16,050	6,64%	5,94%	16,050	6,64%	5,94%
At 200 docs	11,900	11,775	12,025	1,05%	2,12%	12,675	6,51%	7,64%
At 500 docs	7,410	7,160	7,650	3,24%	6,84%	7,770	4,86%	8,52%
At 1000 docs	4,910	4,505	5,000	1,83%	10,99%	5,170	5,30%	14,76%
R-Precision	16,116	17,703	20,449	26,89%	15,51%	20,721	28,57%	17,05%
Average Precision	11,643	14,388	16,402	40,87%	14,00%	16,297	39,97%	13,27%

Confidence level

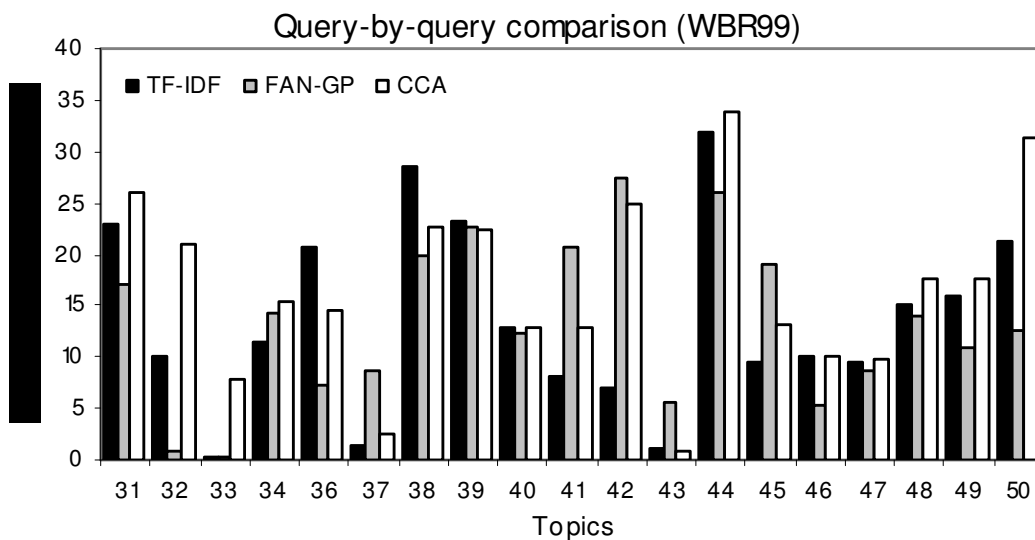
over TF-IDF 98,39%  
over FAN-GP 98,19%

98,95%  
93,63%

# TREC: Precision x Recall



# TREC: Query-by-query comparison



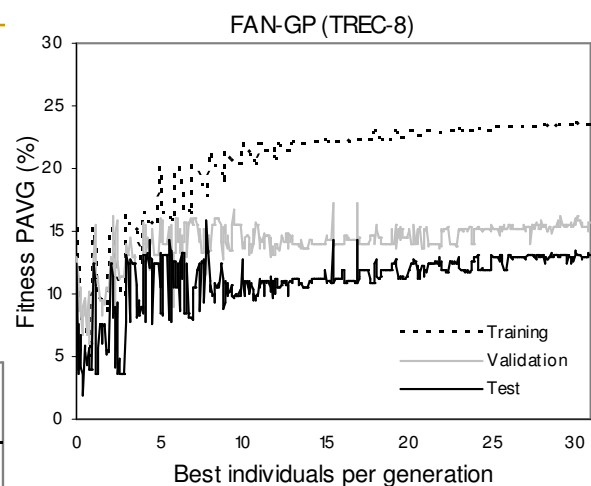
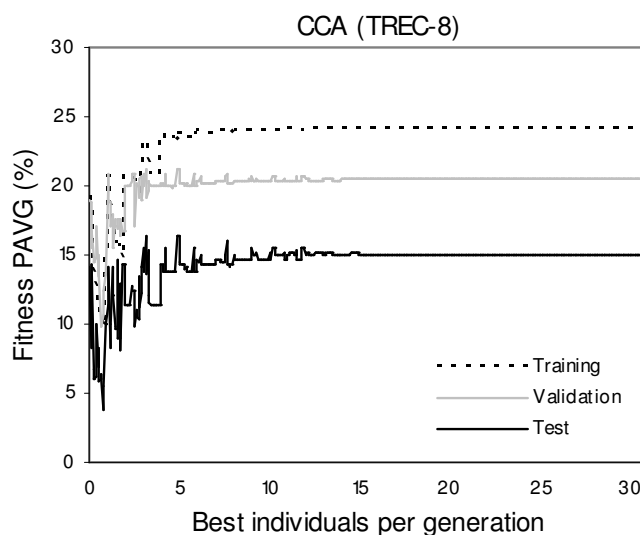
# TREC: The best individuals - CCA

## ■ CCA\_MAP\_D5

```
(*  
  (* (log t08) (+ t05 t07))  
  (+ (+ (* (+ t19 t05) (+ t07 t06))  
      (* (+ t06 t02) (* t18 t18)))  
  (/ t07 t19))  
)
```

- t07: Alternative IDF
- t05: Okapi BM25 term-frequency factor
- t19: Okapi BM25 query factor

# TREC: Overfitting



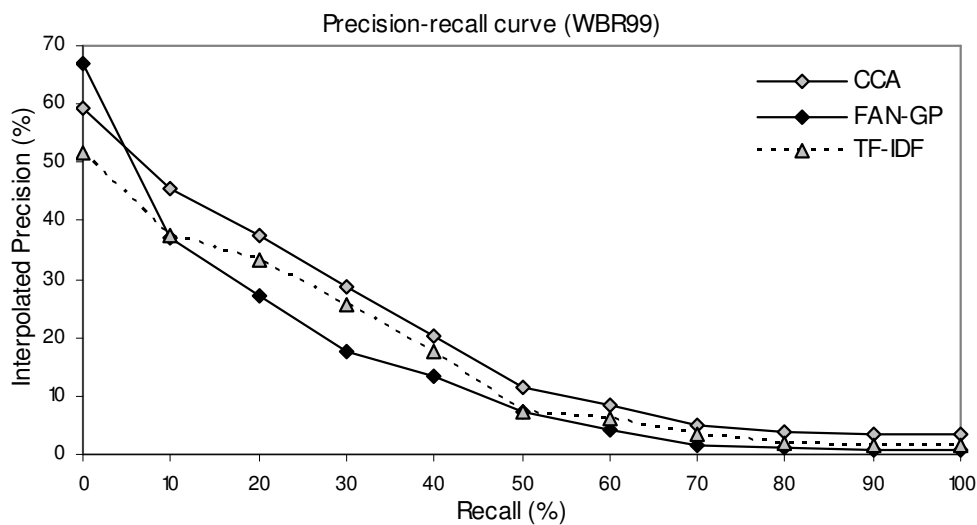
## WBR: Results

Interpolated precision	TF-IDF	FAN-GP	CCA-8	Gain over TF-IDF	Gain over FAN-GP	CCA-11	Gain over TF-IDF	Gain over FAN-GP
At 5 docs	23,157	35,790	36,842	59,10%	2,94%	31,579	36,37%	-11,76%
At 10 docs	26,315	32,632	32,632	24,00%	0,00%	32,105	22,00%	-1,61%
At 15 docs	30,175	29,474	29,123	-3,49%	-1,19%	33,333	10,46%	13,09%
At 20 docs	32,105	26,842	27,368	-14,75%	1,96%	31,053	-3,28%	15,69%
At 30 docs	25,263	21,579	25,965	2,78%	20,33%	27,018	6,94%	25,20%
At 100 docs	13,421	10,737	13,632	1,57%	26,96%	13,526	0,78%	25,98%
At 200 docs	9,000	7,553	8,790	-2,34%	16,38%	9,000	0,00%	19,16%
At 500 docs	3,726	3,516	3,737	0,28%	6,29%	3,779	1,41%	7,48%
At 1000 docs	1,968	1,879	2,005	1,87%	6,73%	2,016	2,41%	7,29%
R-Precision	20,607	19,830	22,294	8,19%	12,43%	22,378	8,59%	12,85%
Average Precision	13,710	13,361	16,681	21,67%	24,85%	17,200	25,45%	28,73%

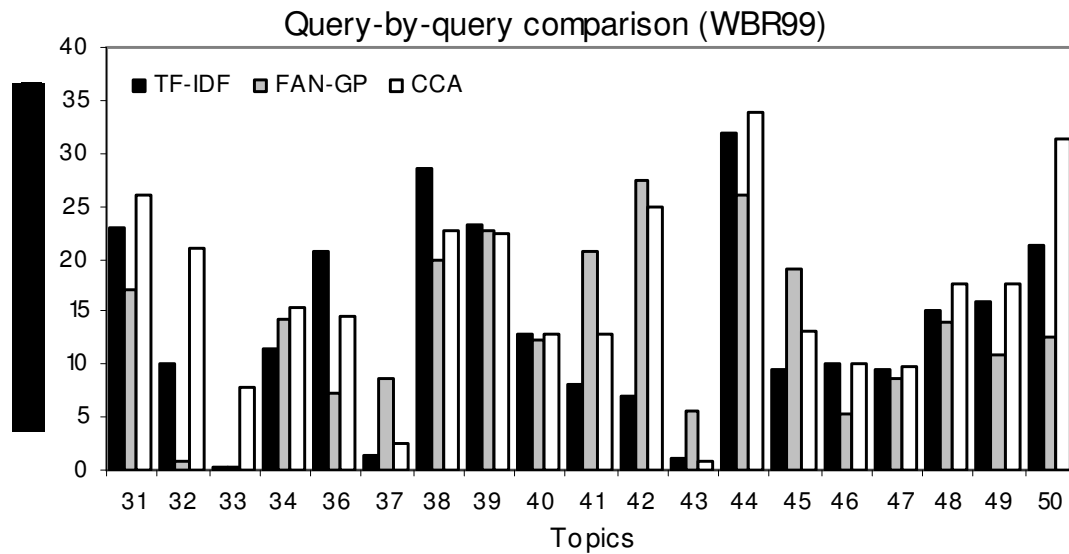
Confidence level  
 over TF-IDF 98,26%  
 over FAN-GP 96,30%

96,96%  
 96,04%

## WBR: Precision x Recall



# WBR: Query-by-query comparison



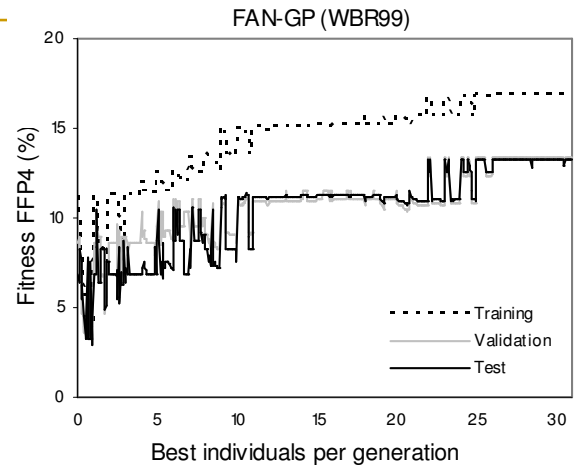
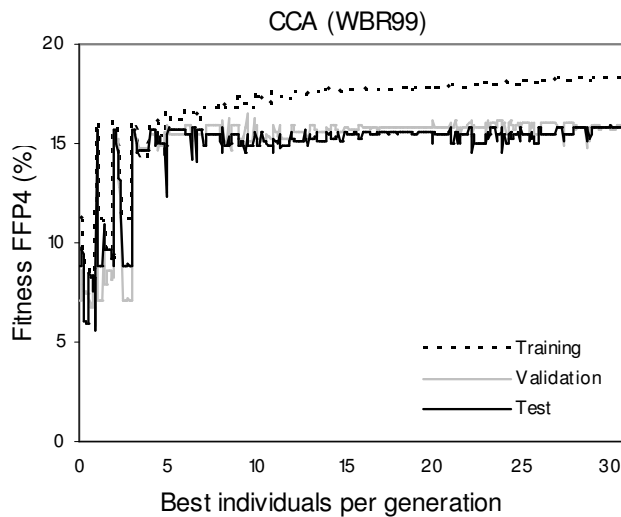
## WBR: The best individual - CCA

### ■ CCA\_FFP4\_D8

```
(+ (+ (+ 99.09 t11)
  (+ (* (* t07 t10)
    (* t05 (* (+ (* t07 t10) (+ t08 t10)) (* t12 t01))))
  (* (* t07 t10)
    (* t05 (* (+ (* t02 t04) (+ t08 t10)) (* t12 t01))))))
(+ (* t12 t01)
  (* (* t07 t10)
    (* t05 (* (+ (/ t08 t20) (+ t08 t10)) (* t12 t01))))))
)
```

- t01: term-frequency
- t10: a probabilistic inverse collection frequency
- t12: cosine normalization

# WBR: Overfitting



## Conclusions

- Is it possible to improve the results using meaningful, rich, and proven components extracted from well-known ranking functions?
  - Meaningful components have presented better results than statistical information of a collection
  - CCA ranking functions have converged faster than FAN-GP
  - Our approach also have reduced overfitting

---

## Future work

- Investigate new meaningful evidences for our Combined Component Approach
- Explore structure for Web Search (e.g., title, anchor text, etc)
- Explore link information for Web search
- Investigate statistical information of the collection based on the subset of documents obtained relative to the query
- Explore termsets (Set-Based Model) for correlation among terms

