Information Retrieval for HR

Ismael Belghiti, CTO @ Hiresweet

Meetup NLP #6 – July 25, 2018

HIRESWEET

Information Retrieval in the NLP World

Speech Recognition

Name Entity Recognition

Parsing

Text to Speech

Information Retrieval

Translation

Topic Modelling

Relationship Extraction

Automatic Summarization

Question Answering

Information Retrieval in the NLP World

Speech Recognition

Name Entity Recognition

Parsing

Text to Speech

Information Retrieval

Translation

Topic Modelling

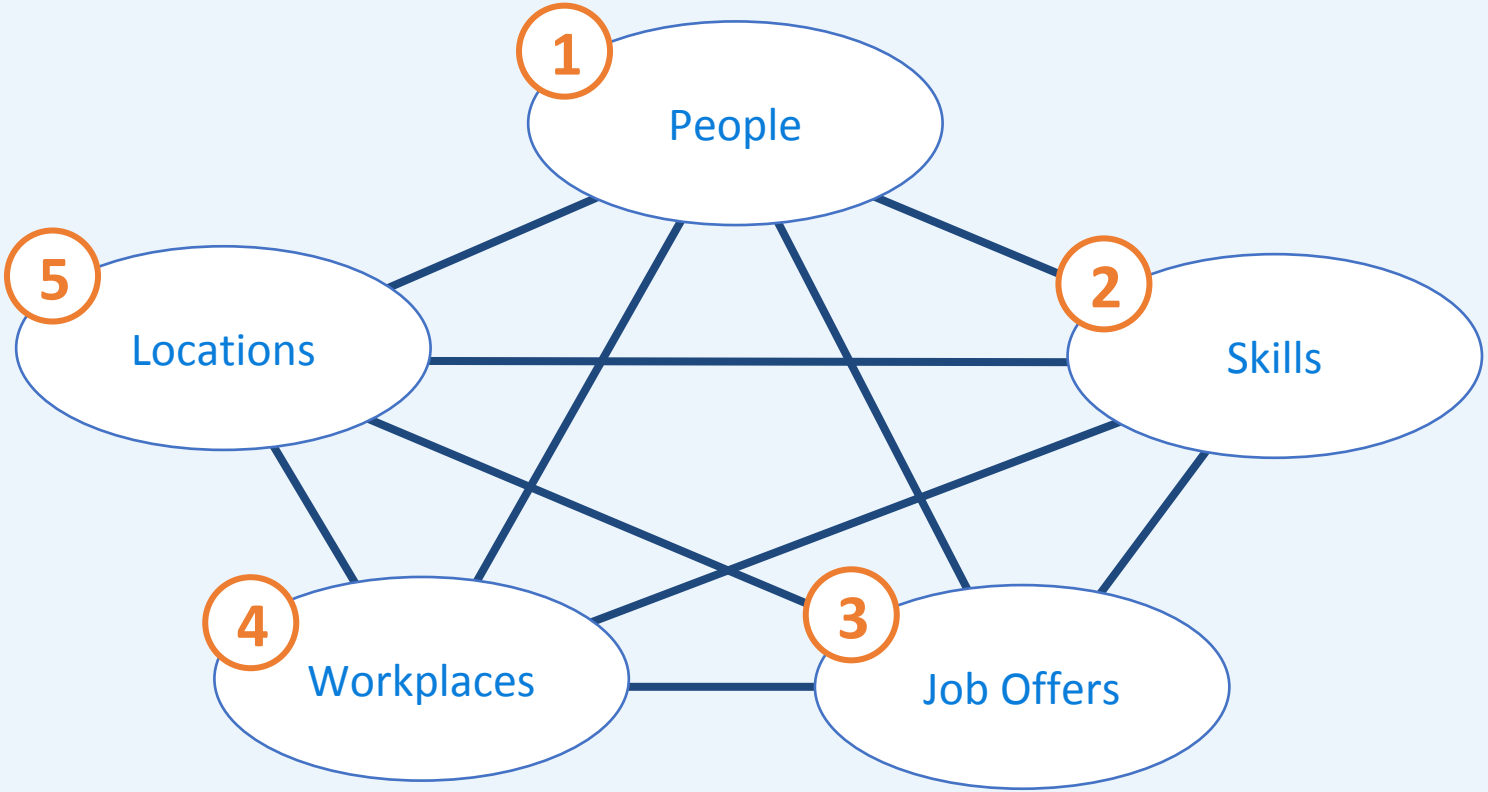
Relationship Extraction

Automatic Summarization

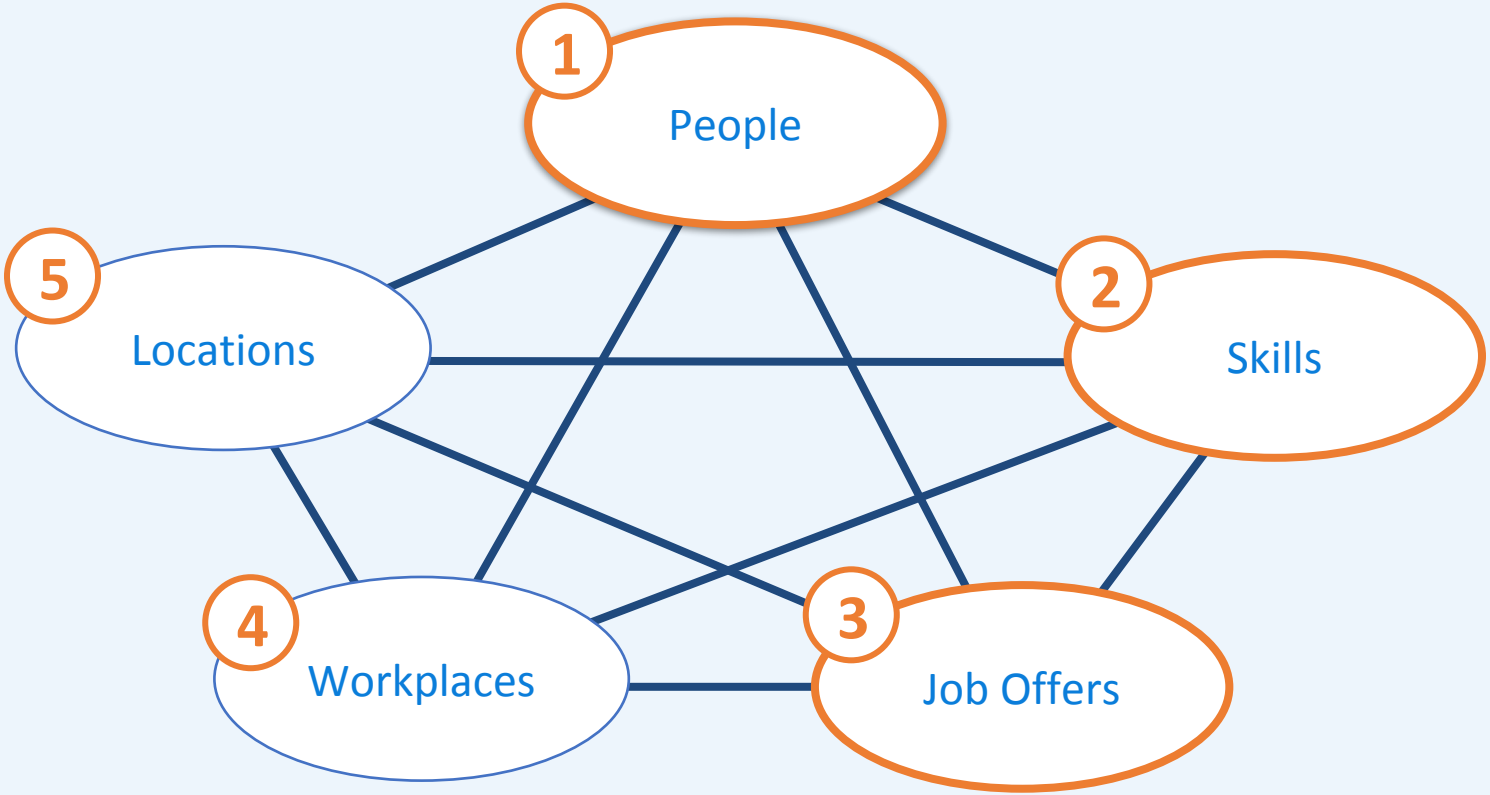
Question Answering

HR Opportunities - Concepts

HR Opportunities - Concepts



HR Opportunities - Concepts



Search Engine

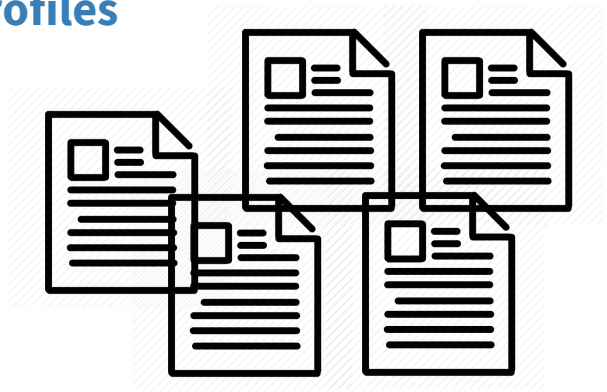
Query

Job Offer

- Required Skills
- Important Skills
- Bonus Skills

Documents

Profiles



Metrics

Normalized Discounted Cumulative Gain (NDCG)

Search Engine

Query

Job Offer

	required	important	bonus
python	1		
nlp			1
java			
scikit-learn		1	
...			

Documents

Profiles

	headline	bio	last xp
python	1	2	
nlp			
java		1	
scikit-learn		2	1
...			

...

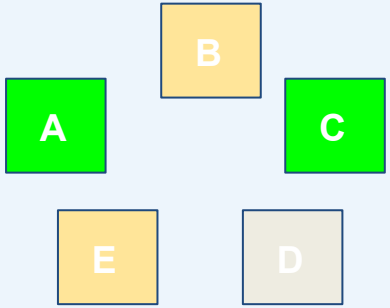
Metrics

Normalized Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain (NDCG)

Extinction factor: **0.8**

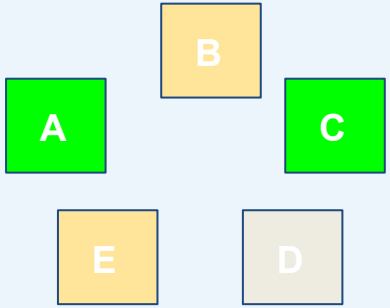


Optimal Ranking

Tested Ranking

Normalized Discounted Cumulative Gain (NDCG)

Extinction factor: **0.8**



Optimal Ranking

1 + 1 x 0.8 + 0.4 x 0.64 + 0.4 x 0.51 + 0 = 2.26 **1**

Tested Ranking

1 + 0 + 0.4 x 0.64 + 1 x 0.51 + 0.4 x 0.41 = 1.93 **0.85**

Now, let’s have fun !

Naive Scorers

Simplification

Taking the sum on each row

Job Offer

	required	important	bonus	
python	1			1
nlp			1	1
java				0
scikit-learn		1		1

Profile

	headline	bio	last xp	
python	1	2		3
nlp			3	3
java		1		1
scikit-learn				0

Naive Scorers

Simple Similarities

w =

1
1
0
1

x =

3
3
1
0

S(w,x) = <w, F(x)>

- F(x) = x
- F(x) = sqrt(x)
- F(x) = log(1 + x)
- F(x) = x / || x ||
- F(x) = x / | x |

Hall of Fame

Score	Post Processes
33.9	$x \rightarrow \log(1+x)$ *then* L2-normalization
31.6	L2-nomalization
31.3	L1-nomalization
30.7	$x \rightarrow \log(1+x)$
30.4	$x \rightarrow \sqrt{x}$
28.4	none
0	RANDOM

TF - IDF

TF - IDF

A must know

TF-IDF : Term Frequency– Inverse Document Frequency

Term Frequency

How often a given term appears in a given document

Document Frequency

How often a given term is present (at least once) in the documents of the corpus

TF - IDF

A must know

TF-IDF : Term Frequency– Inverse Document Frequency

Term Frequency (D, t)

1
count(D,t)
log(1 + count(D,t))
count(D,t) / size(D)
count(D,t) / “max count(D, t’)”
log(1 + count(D,t) / size(D))

Inverse Document Frequency (t)

1
N / N(t)
log(N / N(t))
log(1 + N / N(t))

N(t) : number of documents in which t appears (at least once)

TF - IDF

A must know

TF-IDF : Term Frequency– Inverse Document Frequency

Term Frequency (D, t)

1
count(D,t)
log(1 + count(D,t))
count(D,t) / size(D)
count(D,t) / “max count(D, t’)”
log(1 + count(D,t) / size(D))

Inverse Document Frequency (t)

1
N / N(t)
log(N / N(t))
log(1 + N / N(t))

N(t) : number of documents in which t appears (at least once)

TF-IDF simply means “TF multiplied by IDF”

Hall of Fame

Score	Post Processes
34.2	TF: 1+log(count(D,t) / max) ; IDF: log(N / N(t))
33.9	x → log(1+x) *then* L2-normalization
32.2	TF: log(1 + count(D,t) / size(D)) ; IDF : log(1+N / N(t))
29.5	TF: count(D,t) ; IDF: log(N / N(t))
28.4	none
0	RANDOM

Hall of Fame

Score	Post Processes
34.2	TF: 1+log(count(D,t) / max) ; IDF: log(N / N(t))
33.9	x → log(1+x) *then* L2-normalization
31.6	L2-nomalization
31.3	L1-nomalization
30.7	x → log(1+x)
30.4	x →√x
28.4	none
0	RANDOM

How to go further ?

How to go further ?

Better vectorizations

Learning parameters across queries

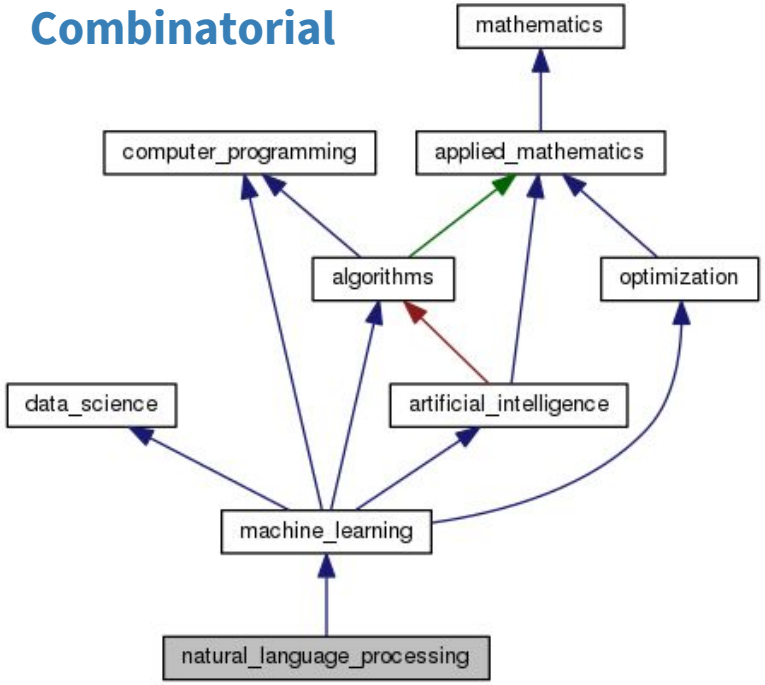
Merging Subscorers

Better Vectorizations (Embeddings)

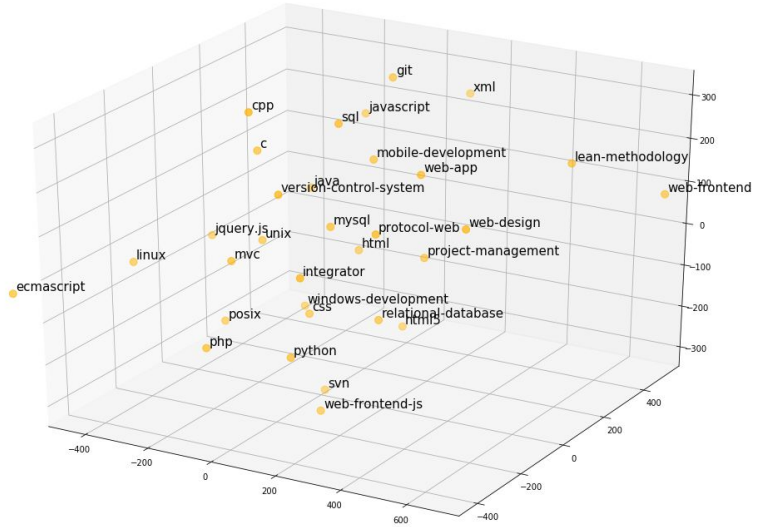
Better Vectorizations (Embeddings)

We have concepts which live in a “structured world”

Combinatorial



Continuous



Better Vectorizations (Embeddings)

We can generalize the notion of "occurrences" => "hints"

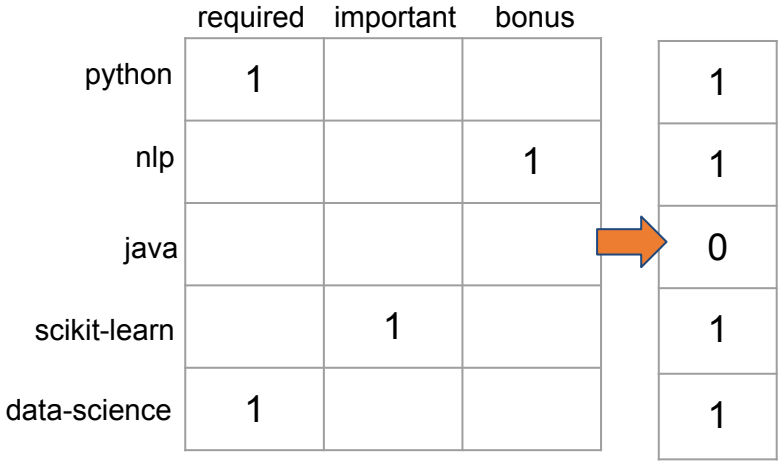
Examples of hints for concept "Python"

- "Python " in headline (1)
- 3 "Django" projects (0.6)
- "Data Science" in summary (0.1)
- "Numpy" in summary (0.8)
- "Python" in 3 forked projects (0.2)
- "Scikit-Learn" in Last XP (0.7)

Aggregating : [1 , 0.8 , 0.7 , 0.6 , 0.2 , 0.1] ⇨ **0.73**

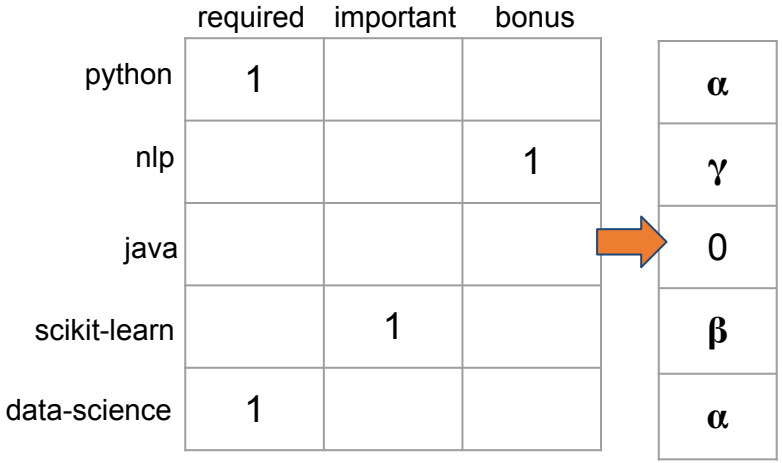
Better Vectorizations (Embeddings)

We can also have a better representations for queries !



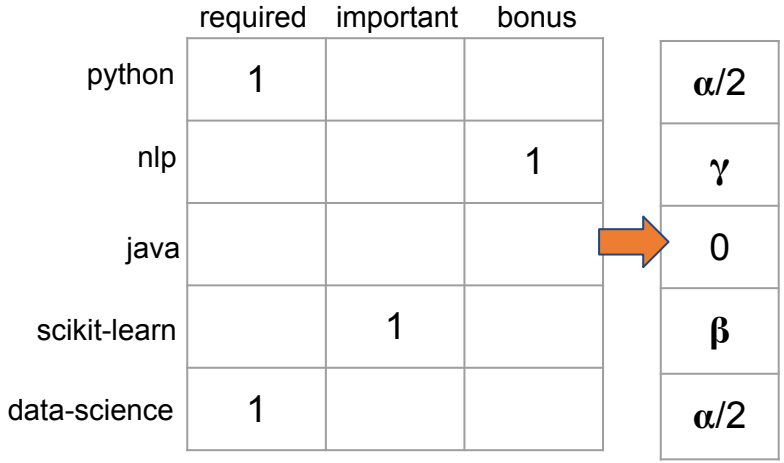
Better Vectorizations (Embeddings)

We can also have a better representations for queries !



Better Vectorizations (Embeddings)

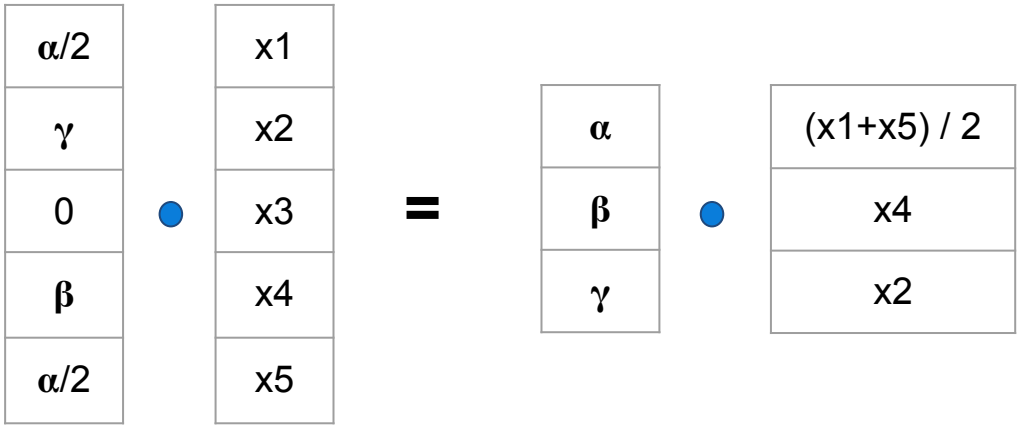
We can also have a better representations for queries !



Learning parameters across queries

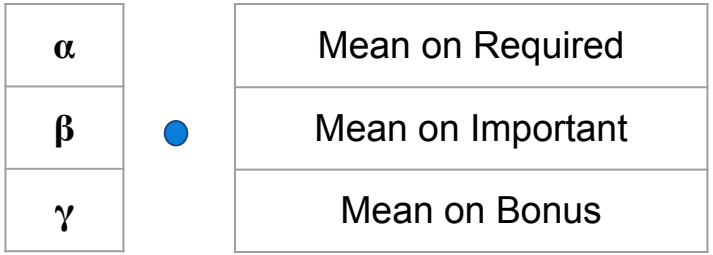
Learning parameters across queries

Let's rewrite this dot product !



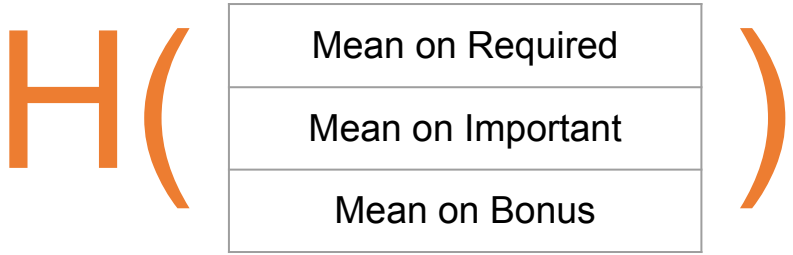
Learning parameters across queries

Let’s rewrite this dot product !



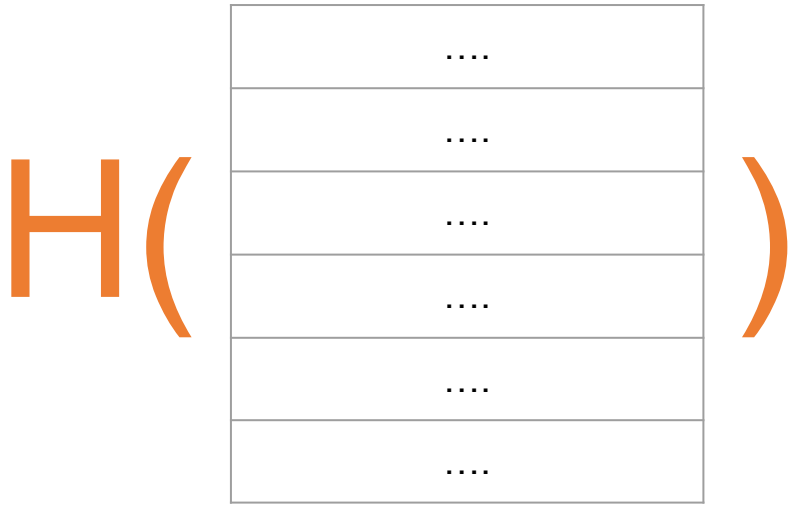
Learning parameters across queries

We can be more general !



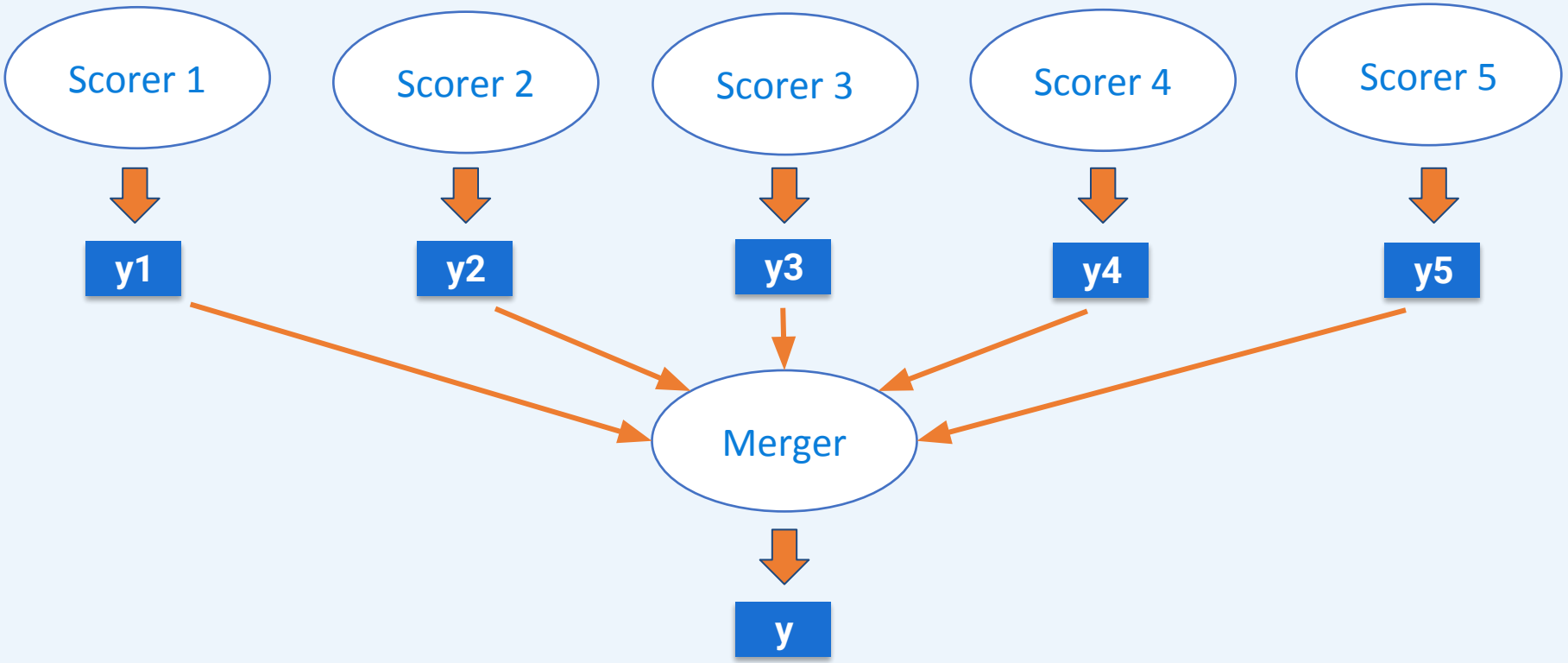
Learning parameters across queries

We can be (even) more general !



Merging Subscorers

Merging Subscorers



Merging Subscorers

Score	Algo
37.3	S5
36.9	S4
36.3	S3
34.2	S2
33.9	S1
28.4	none
0	RANDOM

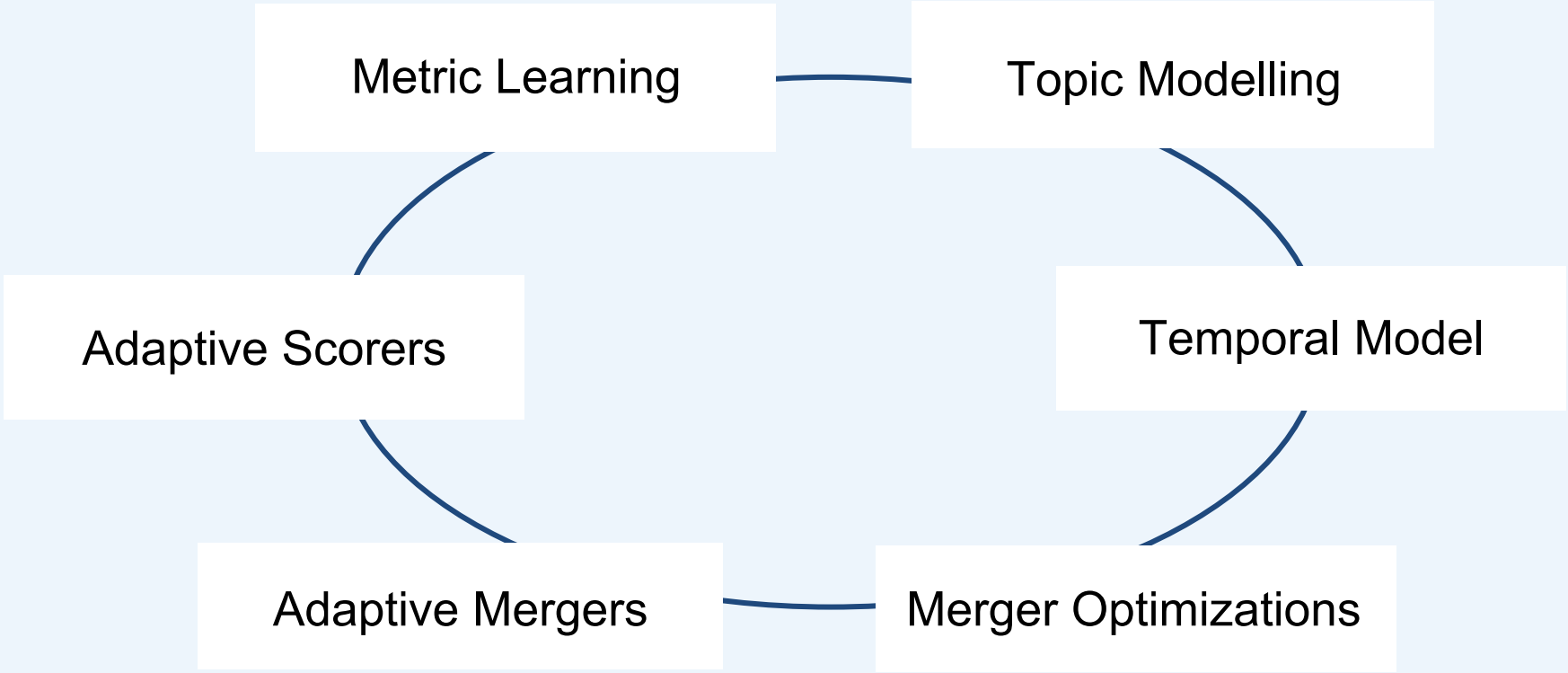
Merging Subscorers

Score	Algo
39.2	Mean(S1, S2, S3, S4, S5)
37.3	S5
36.9	S4
36.3	S3
34.2	S2
33.9	S1
28.4	none
0	RANDOM

Merging Subscorers

Score	Algo
42.0	OptimizedMerge(S1, S2, S3, S4, S5)
39.2	Mean(S1, S2, S3, S4, S5)
37.3	S5
36.9	S4
36.3	S3
34.2	S2
33.9	S1
28.4	none
0	RANDOM

More Advanced Stuff



Thank you !