Various improvements to a Deep neural language model based method for academic expert search

HOUATI Chakib Mouloud
MEZIANI Serine

University Of Science And Technology Houari Boumediene

BELLAZZOUGUI Djamal
CHAA Messaoud

Research Center On Scientific and Technical Information

PLAN

- Introduction
- Proposed approaches
- Evaluation corpus
- Experiments and evaluations
- Analysis and discussion
- Conclusion

INTRODUCTION

Expert search

Expert search aims to find and rank experts based on a user's query. Iit's a recurring task in the academic world.

These systems are used to look for:

- Supervisors
- Evaluators for research project proposals
- Members of conference program committees, etc.

Expert search models

Several models have been proposed to tackle the problem of expert search.

- Generative probabilistic models
- Discriminative models
- Voting models
- Graph-based models

The use Deep Learning

Recently, new approaches based on Deep Learning have shown very good results.

The use of DL in information retrieval is mainly in the data representation stage.

Embeddings

Vectorial representation using neural networks, to capture the meaning of the word in a vector according to the context

Several models have been proposed:

- Word2vec
- glove
- BERT :
  - RoBERTa
  - SciBERT

Baseline model

Berger et al. : Effective Distributed Representations for Academic Expert Search [2020]

Uses a variant of the neural text representation model BERT to represent documents and queries, and apply various improvements.

OVERVIEW OF OUR WORK

Three main contributions

1

Query expansion

2

Indexing
by sentence

3

Document and author
matching
(scoring with the
author's domain)

PROPOSED APPROACHES

Query expansion

QE is a technique that aims to reduce the gap between the query and the document, in order to improve the quality of the results

We have retrieved the query definitions from Wikipedia

Query expansion

Mean method :

Query

Retrieve the definition from Wikipedia

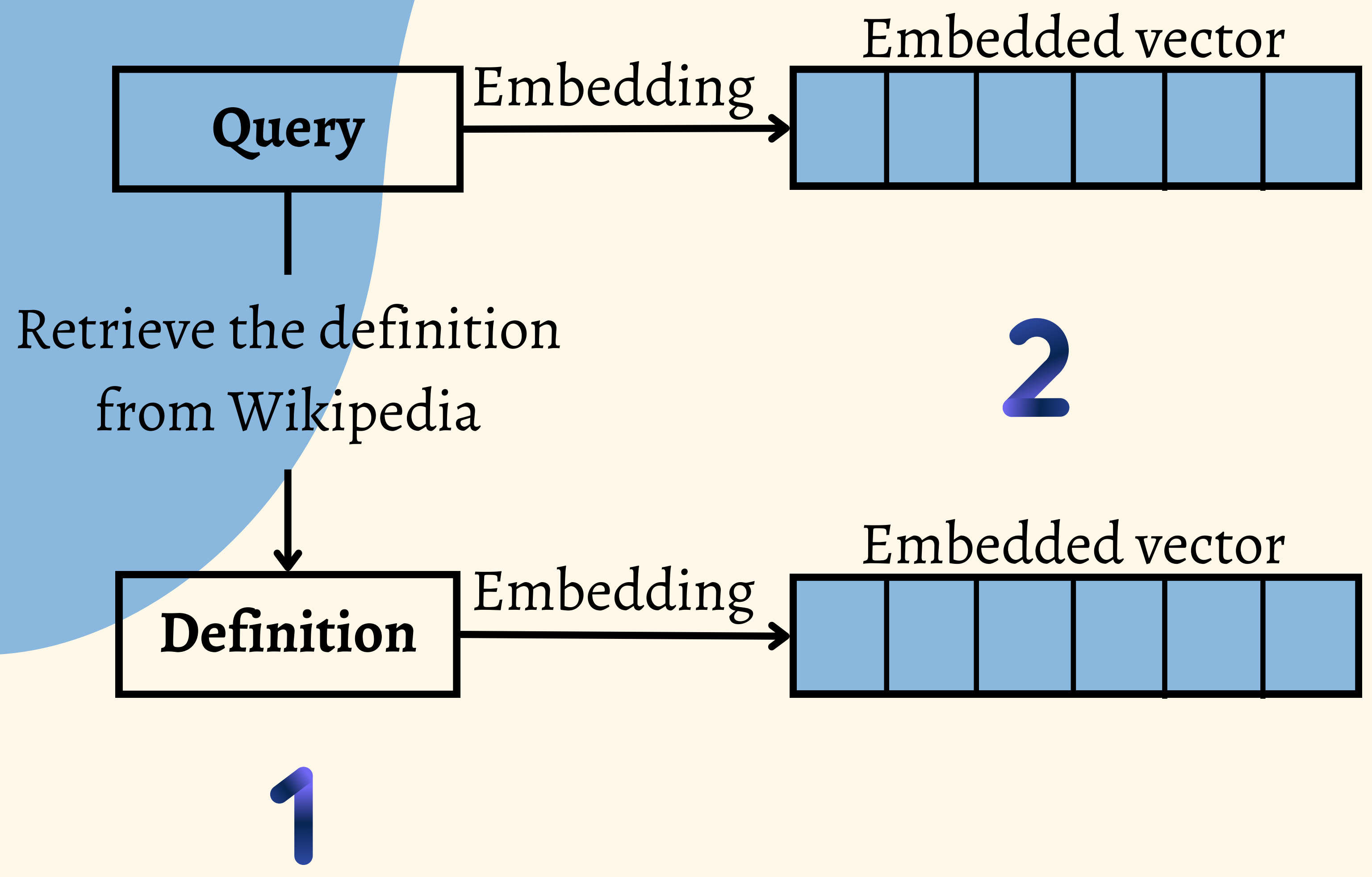
↓

Definition

1

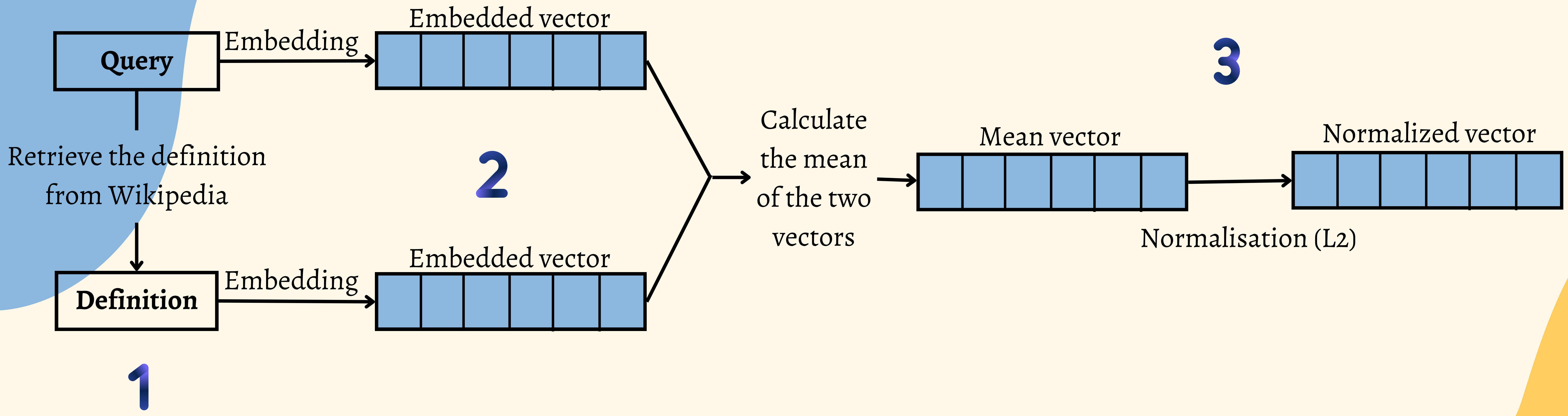
Query expansion

Mean method :



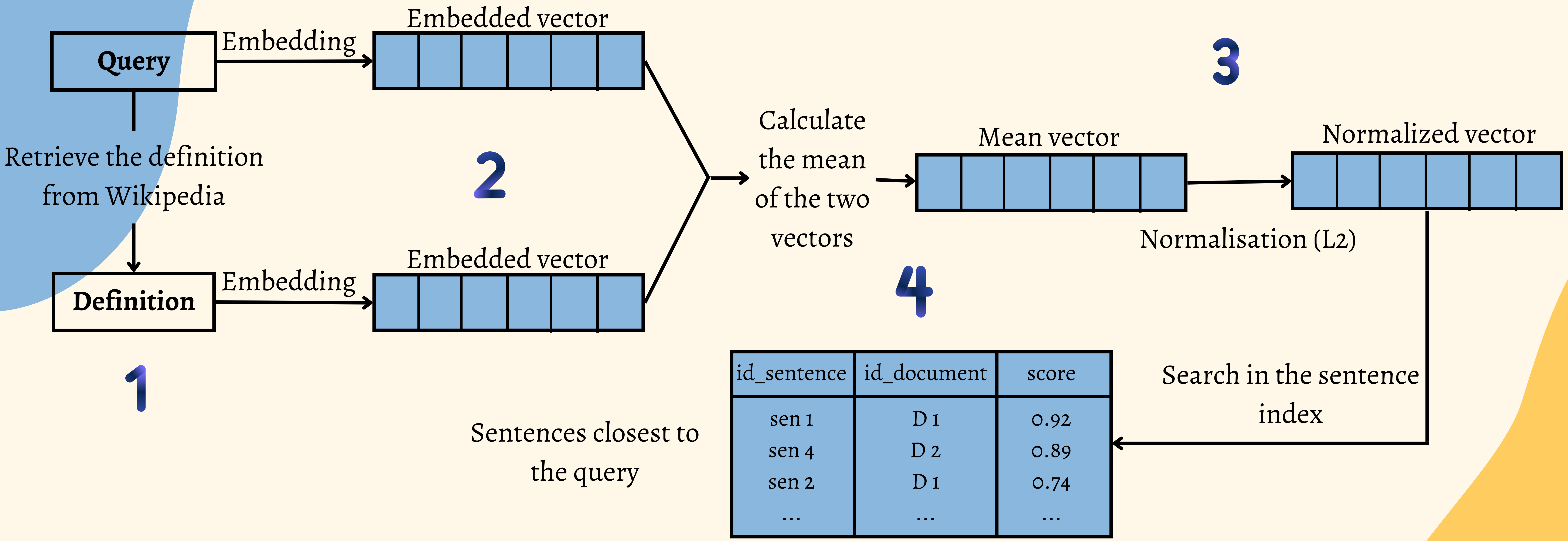
Query expansion

Mean method :



Query expansion

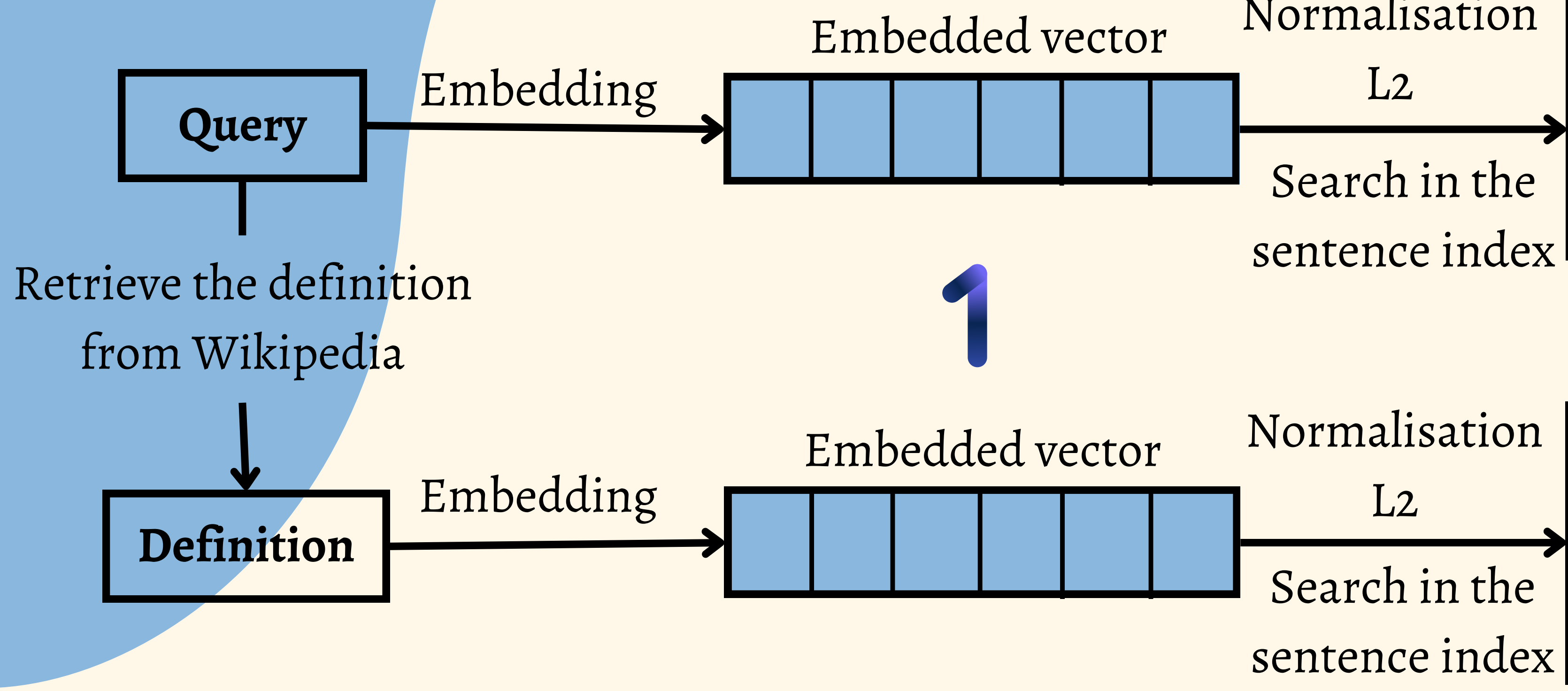
Mean method :



id_sentence	id_document	score
sen 1	D 1	0.92
sen 4	D 2	0.89
sen 2	D 1	0.74
...	...	...

Query expansion

Hybrid method :

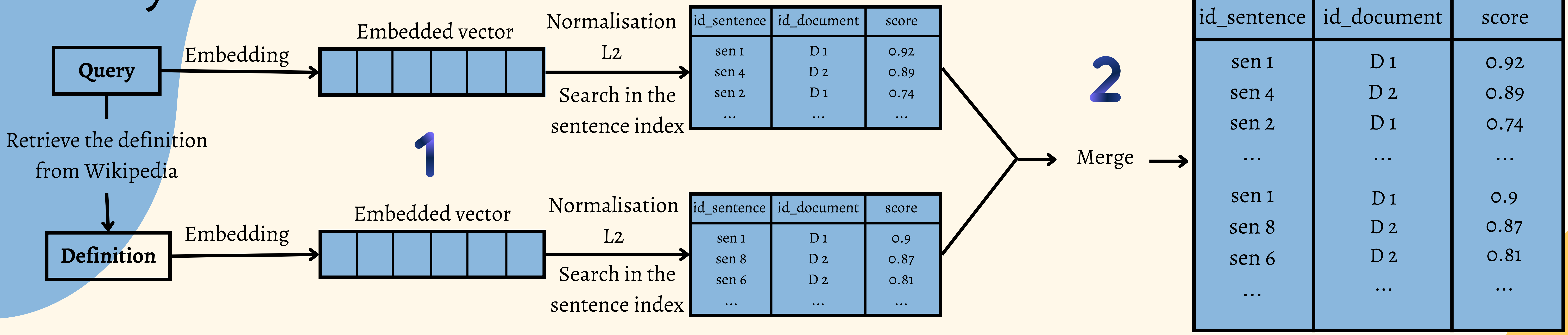


id_sentence	id_document	score
sen 1	D 1	0.92
sen 4	D 2	0.89
sen 2	D 1	0.74
...	...	...

id_sentence	id_document	score
sen 1	D 1	0.9
sen 8	D 2	0.87
sen 6	D 2	0.81
...	...	...

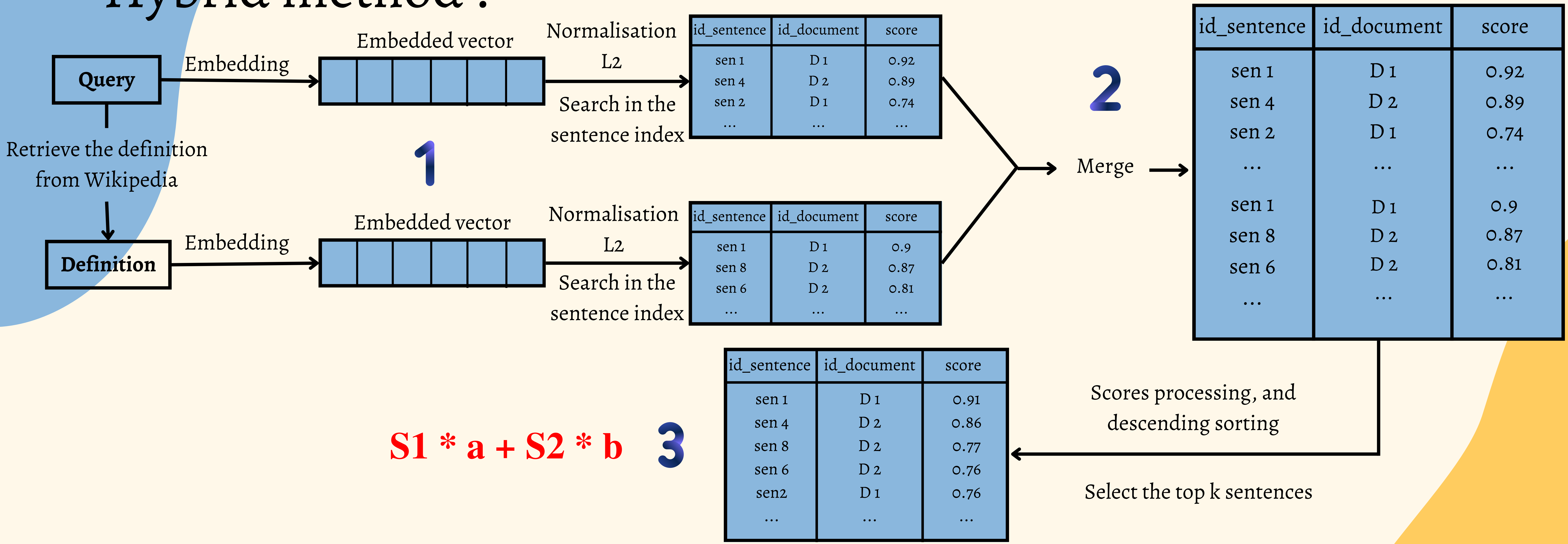
Query expansion

Hybrid method :



Query expansion

Hybrid method :



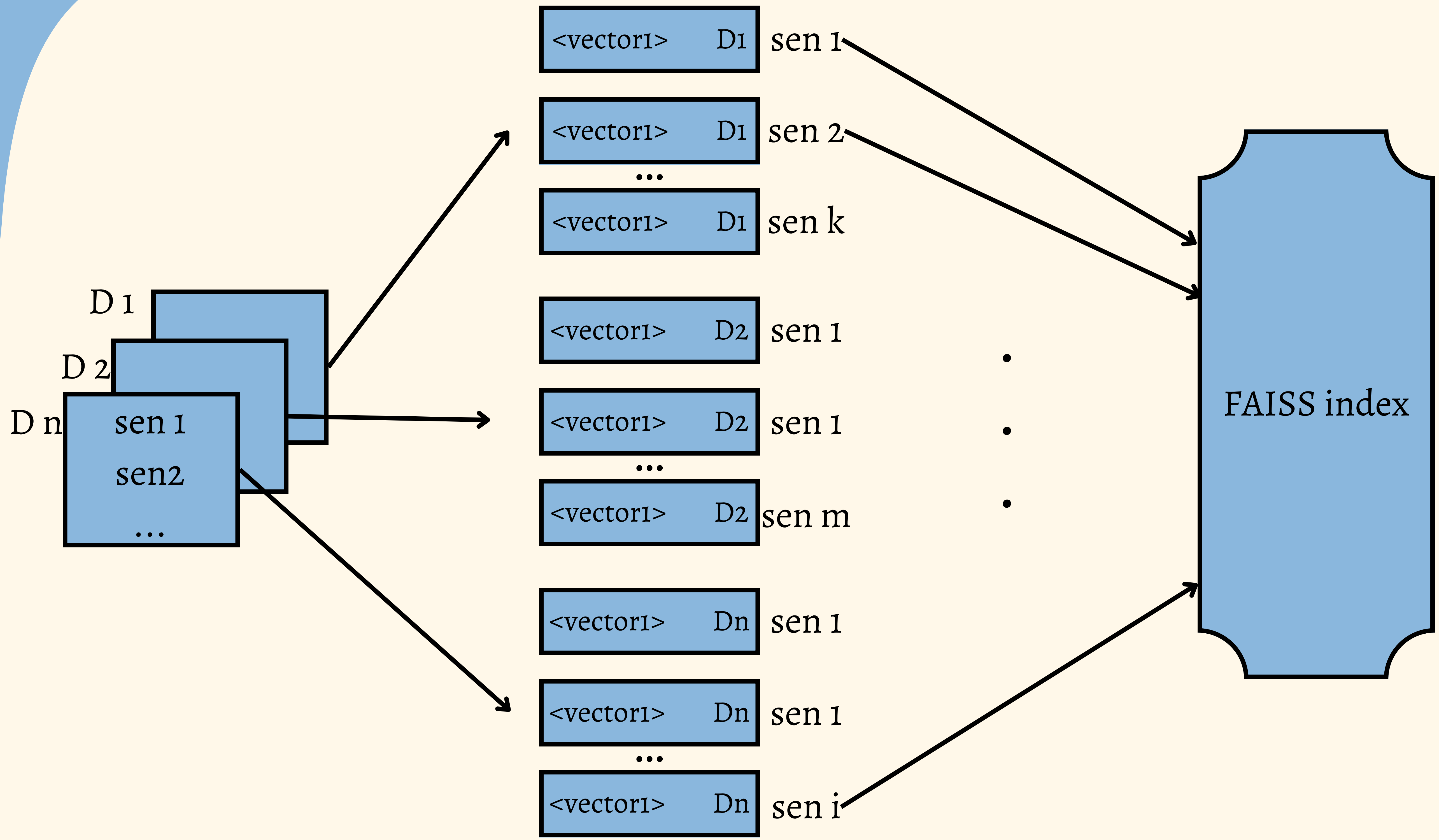
Indexing by sentence

A document is composed of several sentences, and each of them has a different contribution to the overall idea of the paragraph.

The idea behind our method is to limit the impact of sentences that are not significant in relation to the query

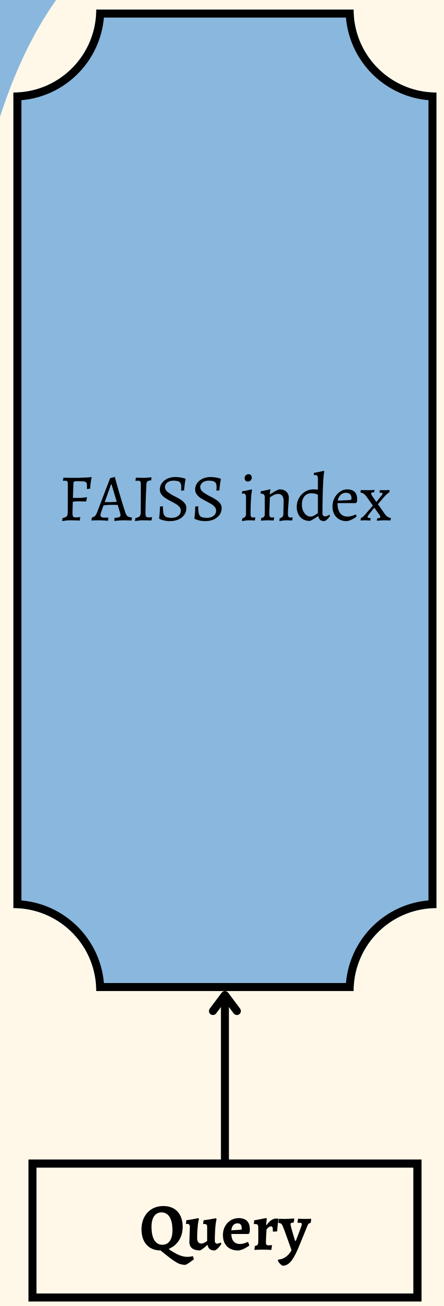
Indexing by sentence

General scheme of the indexing process



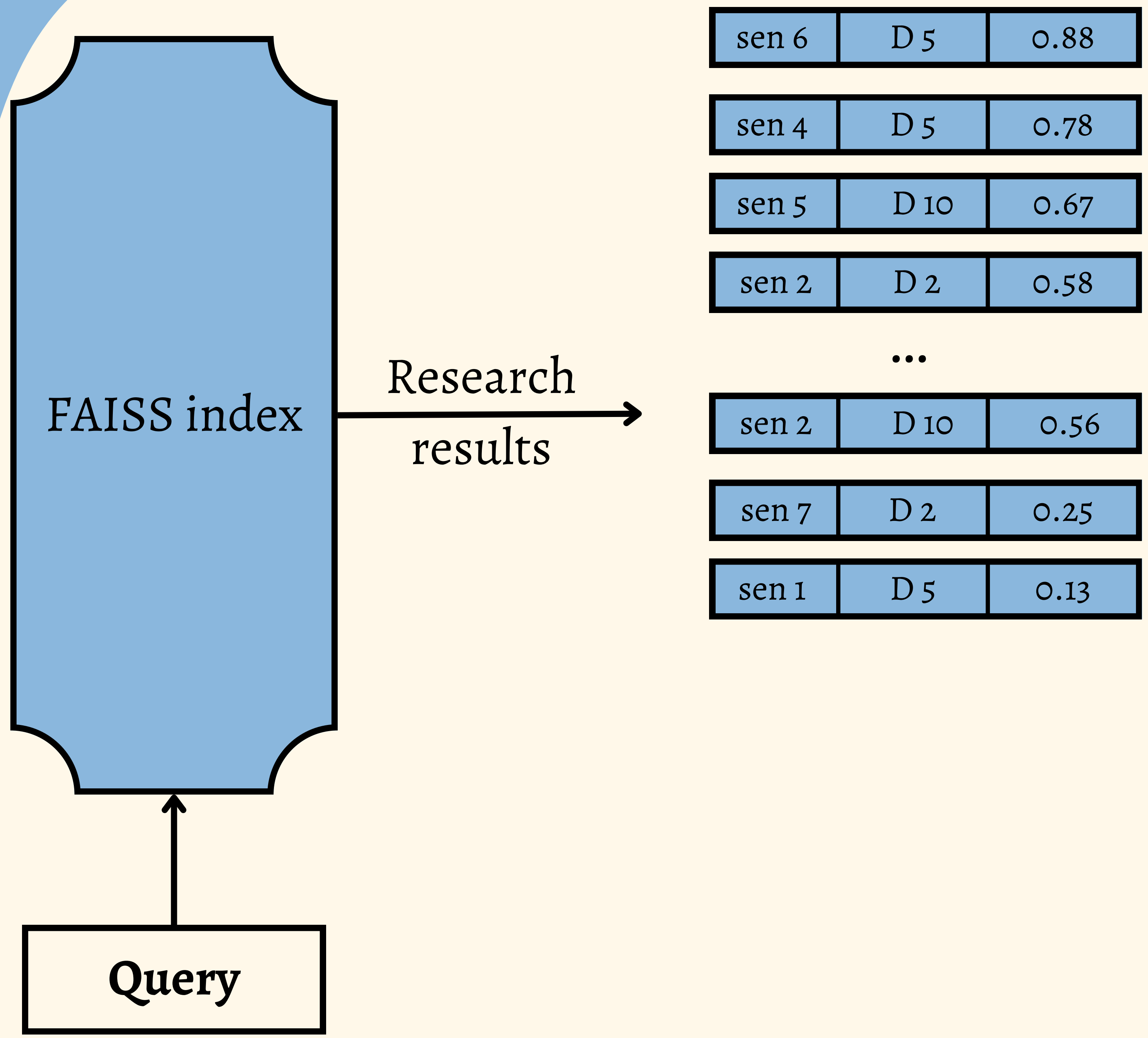
Indexing by sentence

General scheme of the research process



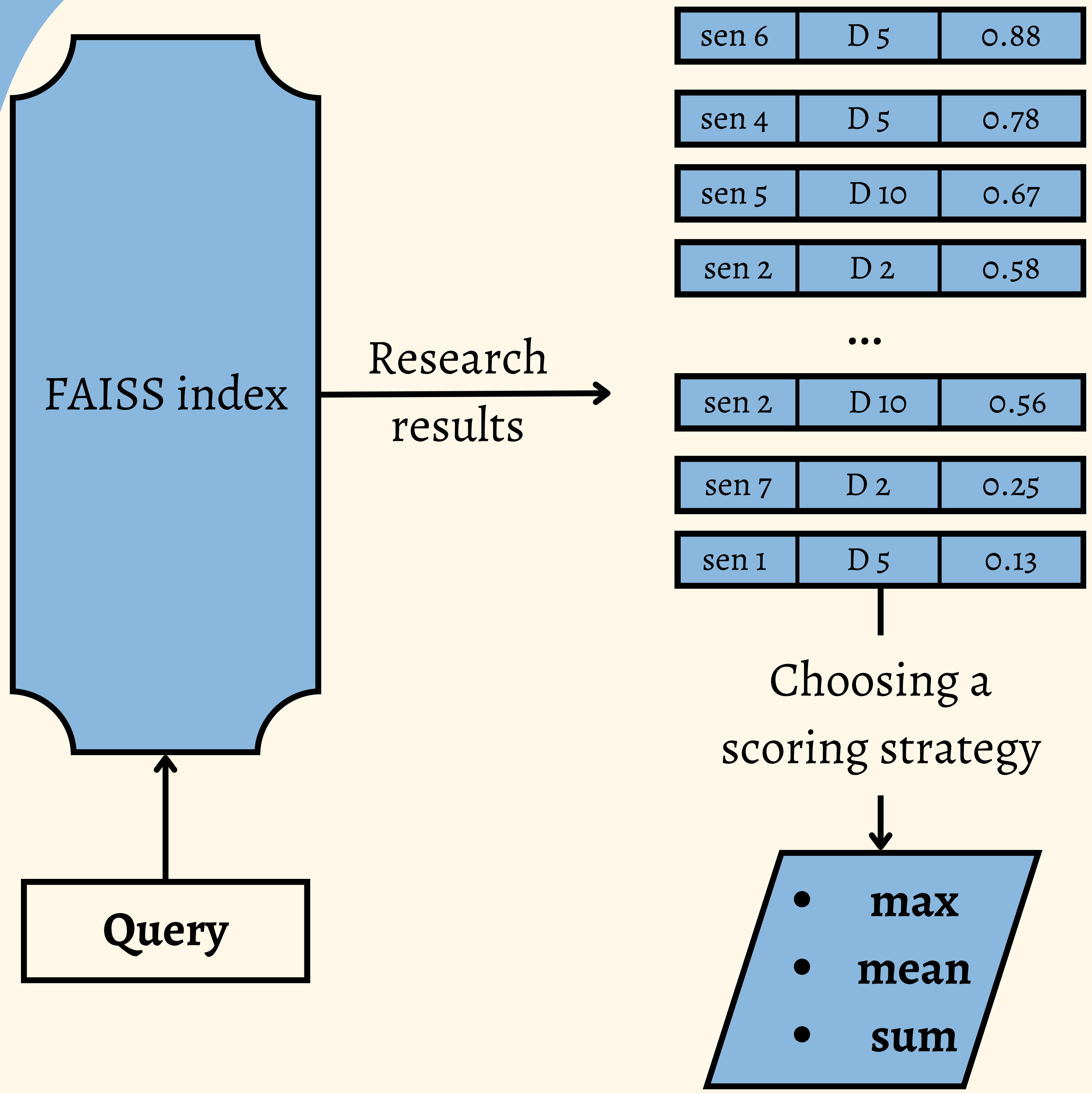
Indexing by sentence

General scheme of the research process



Indexing by sentence

General scheme of the research process



Indexing by sentence

Document scoring :

- Maximum :

$$sim_{QD} = max(sim(v_q, v_{phi}))$$

- Mean :

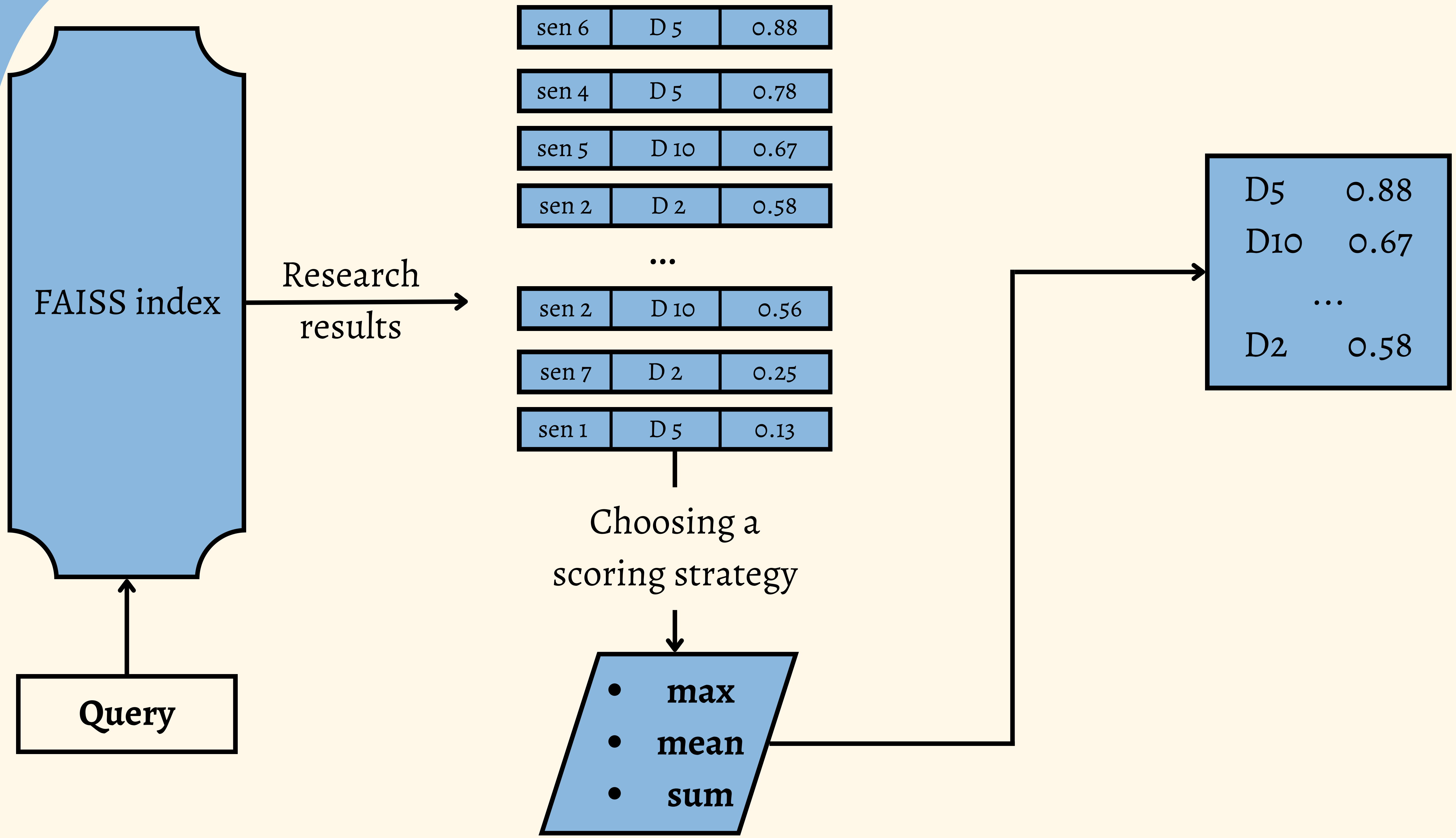
$$sim_{QD} = mean(sim(v_q, v_{phi}))$$

- Sum :

$$sim_{QD} = sum(sim(v_q, v_{phi}))$$

Indexing by sentence

General scheme of the research process

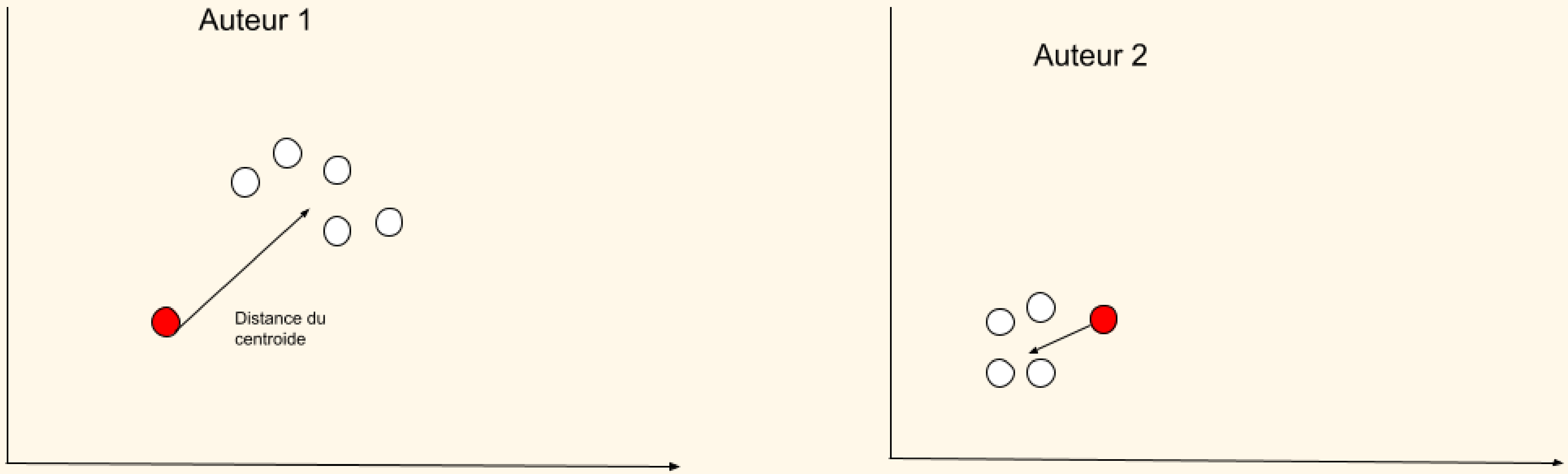


Document and author matching : scoring with the author's domain

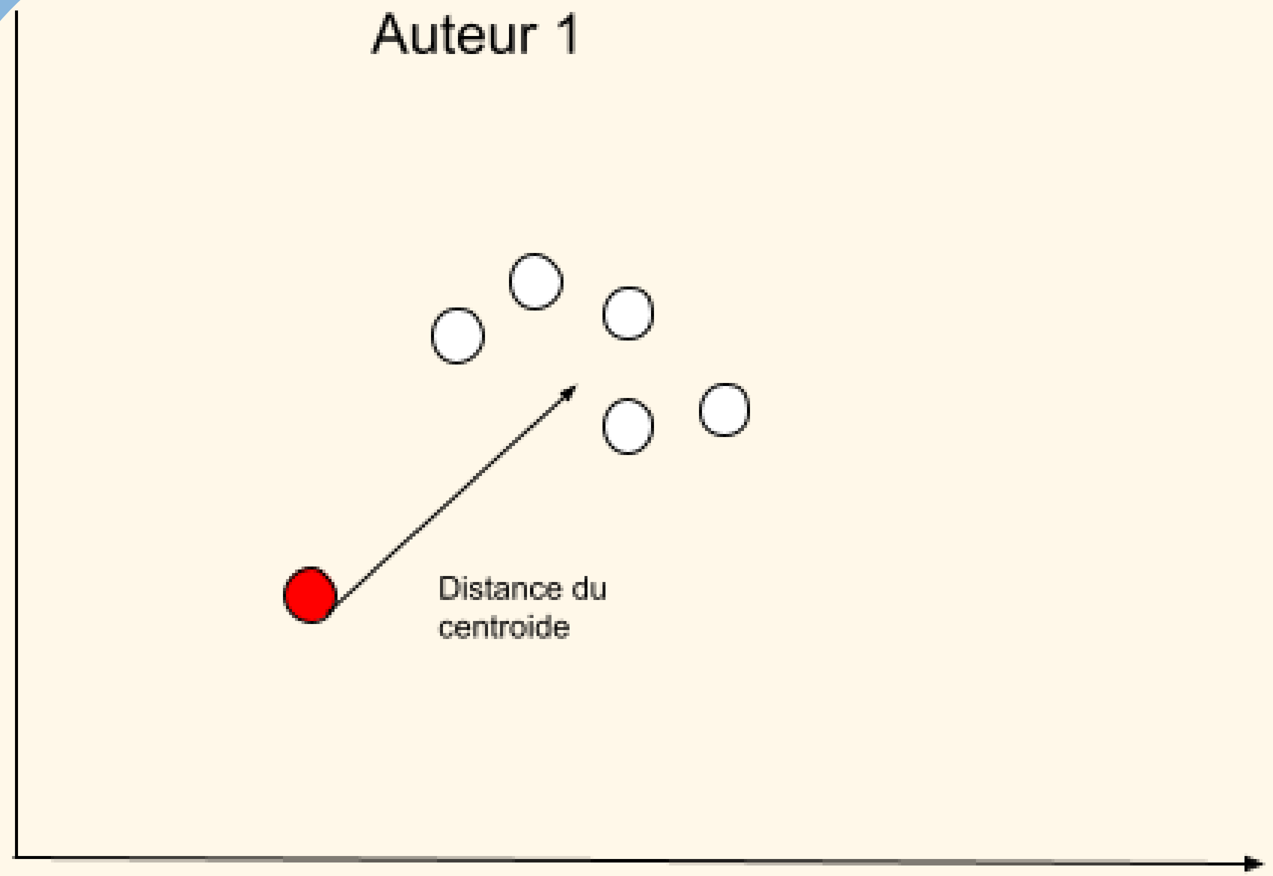
Penalising the score of very prolific authors in order to improve the quality of the results

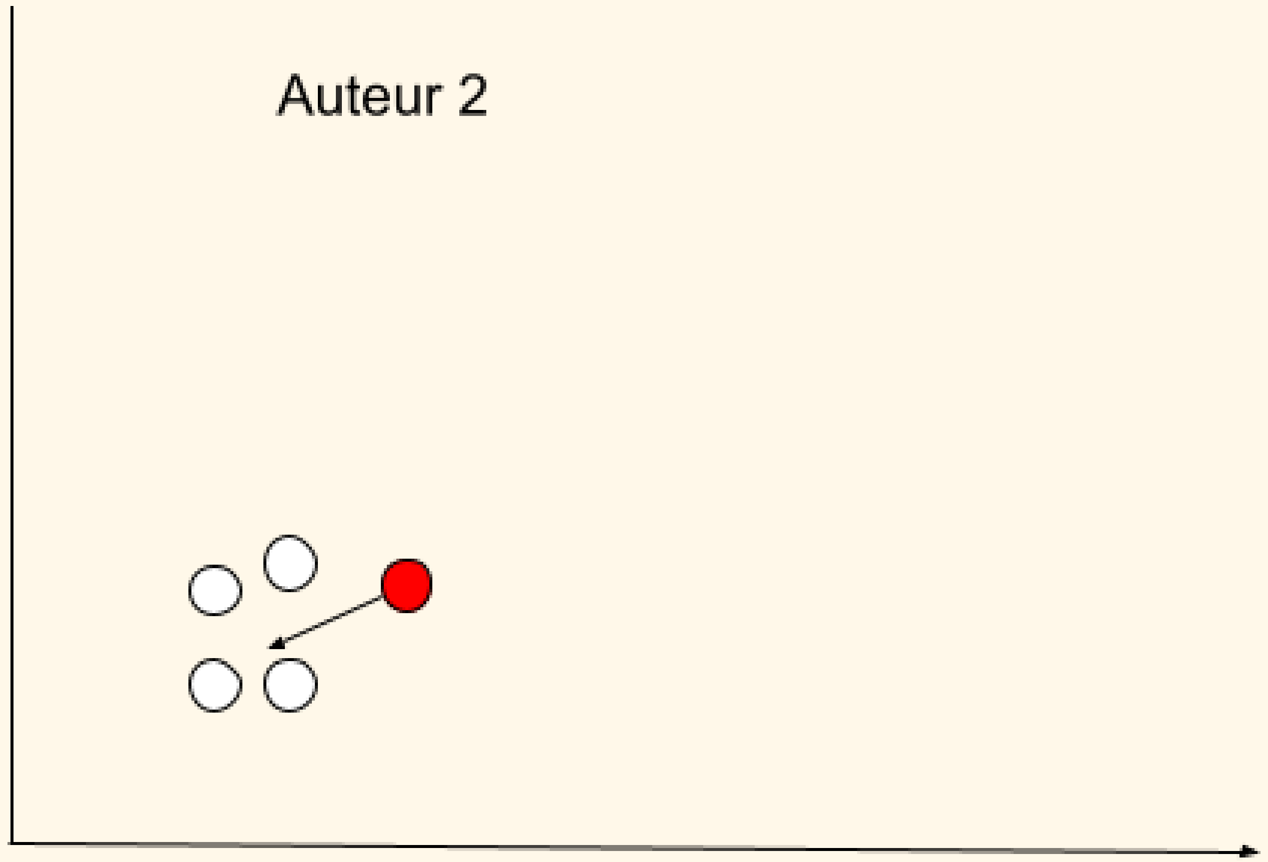
Document and author matching : scoring with the author's domain

The document in red is common to authors 1 and 2



Document and author matching : scoring with the author's domain



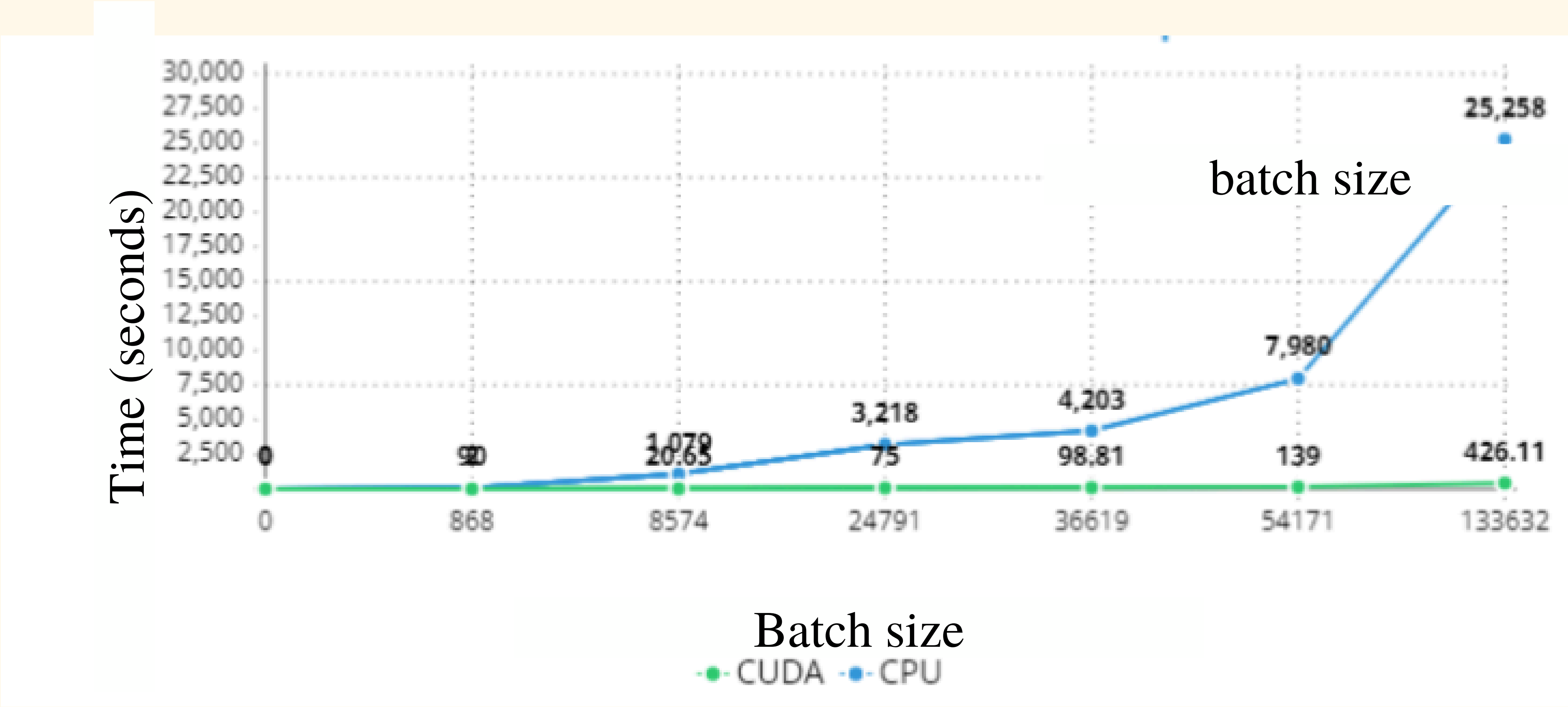


$$\mathrm{score}_{\mathrm{AQ}} = \sum_{i=1}^{N} e^{\mathrm{sim}_{\mathrm{QD}_i}}$$

$$score_{AQ} = \sum_{i=1}^{N} e^{sim_{QDi}/\underline{dist_{ADi}}}$$

OPTIMISATION

The NVIDIA® CUDA® Toolkit allows high performance GPU-accelerated applications



CPU : Ryzen 7 4800H,
8 Gb of RAM

GPU (CUDA):
Nvidia GTX 1660 Ti
10 Gb of VRAM

EVALUATION CORPUS

Arxiv + MAG

- 127 716 articles
- 67 808 authors

- Restricted to the IT field

Corpus ACM

- 291 811 articles
- 13 931 authors

- Many fields

- A database collected, organised and cleaned by us

EXPERIMENTS AND EVALUATIONS

Evaluation methods

- Exact method
- Approximate method

Evaluation metrics

- MRR@10
- MAP@10
- MP@5
- MP@10

Evaluation with Arxiv+MAG

Type de plongement	Type d'indexation	Requête	Exact MRR@10	Approximate MRR@10	Exact MAP@10	Approximate MAP@10	Exact MP@5	Approximate MP@5	Exact MP@10	Approximate MP@10
RoBerta	Indexation par document	Requête initiale	0.835	0.887	0.52	0.687	0.622	0.776	0.594	0.75
		Avec QE	0.865	0.916	0.547	0.708	0.654	0.786	0.614	0.761
	Indexation par phrases	Requête initiale	0.853	0.912	0.471	0.656	0.586	0.738	0.555	0.719
		Avec QE	0.886	0.957	0.549	0.719	0.662	0.81	0.62	0.765
SciBert	Indexation par document	Requête initiale	0.702	1.0	0.227	0.984	0.352	0.99	0.308	0.989
		Avec QE	0.856	1.0	0.477	0.999	0.62	1.0	0.555	0.999
	Indexation par phrases	Requête initiale	0.809	1.0	0.466	0.999	0.568	1.0	0.553	0.999
		Avec QE	0.867	1.0	0.617	0.998	0.72	0.998	0.681	0.999

Evaluation with ACM

Type d'indexation	Requête		Exact MRR@10	Approximate MRR@10	Exact MAP@10	Approximate MAP@10	Exact MP@5	Approximate MP@5	Exact MP@10	Approximate MP@10
Indexation par document	Requête initiale	Sans DA	0.713	1.0	0.346	1.0	0.496	1.0	0.443	1.0
		Avec DA	0.71	1.0	0.314	0.995	0.446	0.996	0.424	0.997
	Avec QE	Sans DA	0.757	1.0	0.428	0.998	0.578	0.998	0.533	0.999
		Avec DA	0.803	1.0	0.422	1.0	0.566	1.0	0.526	1.0
Indexation par phrases	Requête initiale	Sans DA	0.699	1.0	0.328	0.999	0.472	1.0	0.426	0.999
		Avec DA	0.686	1.0	0.332	0.999	0.476	1.0	0.433	0.999
	Avec QE	Sans DA	0.69	1.0	0.229	0.992	0.362	0.996	0.334	0.995
		Avec DA	0.689	1.0	0.237	0.988	0.362	0.994	0.341	0.992

ANALYSIS AND DISCUSSION

Arxiv+MAG database

Query expansion :

- improves the results of all our approaches.
- the "Mean" expansion performs better than the " Hybrid".

Indexing by sentence vs. by document :

- overall, Indexing by sentence was more efficient.
- The "Maximum" strategy, without standardisation, was the best, meaning that working with the most significant sentences of a document was the most effective way.

ACM database

Query expansion :

- Indexing by document : improvement.
- Indexing by sentence : regression.

Indexing by sentence vs. by document :

- In general, indexing by document was better than indexing by sentence.

Document and author matching (scoring with the author's domain):

- brought some improvements in accuracy only in some cases

Scibert vs Roberta

- RoBERTa performed better in the "Exact" evaluations, and SciBERT was better with "Approximate".

- Getting a better score with the Exact method is more difficult, hence RoBERTa is considered to be more efficient than SciBERT.

CONCLUSION

Conclusion

- Indexing by sentence can gives good results with a small corpus.
- The voting formula did not produce the expected results.
- RoBERTa is preferable to SciBERT for indexing
- SciBERT is more appropriate for modelling short sentences in the scientific domain

THANK YOU FOR YOUR ATTENTION!

Any questions?