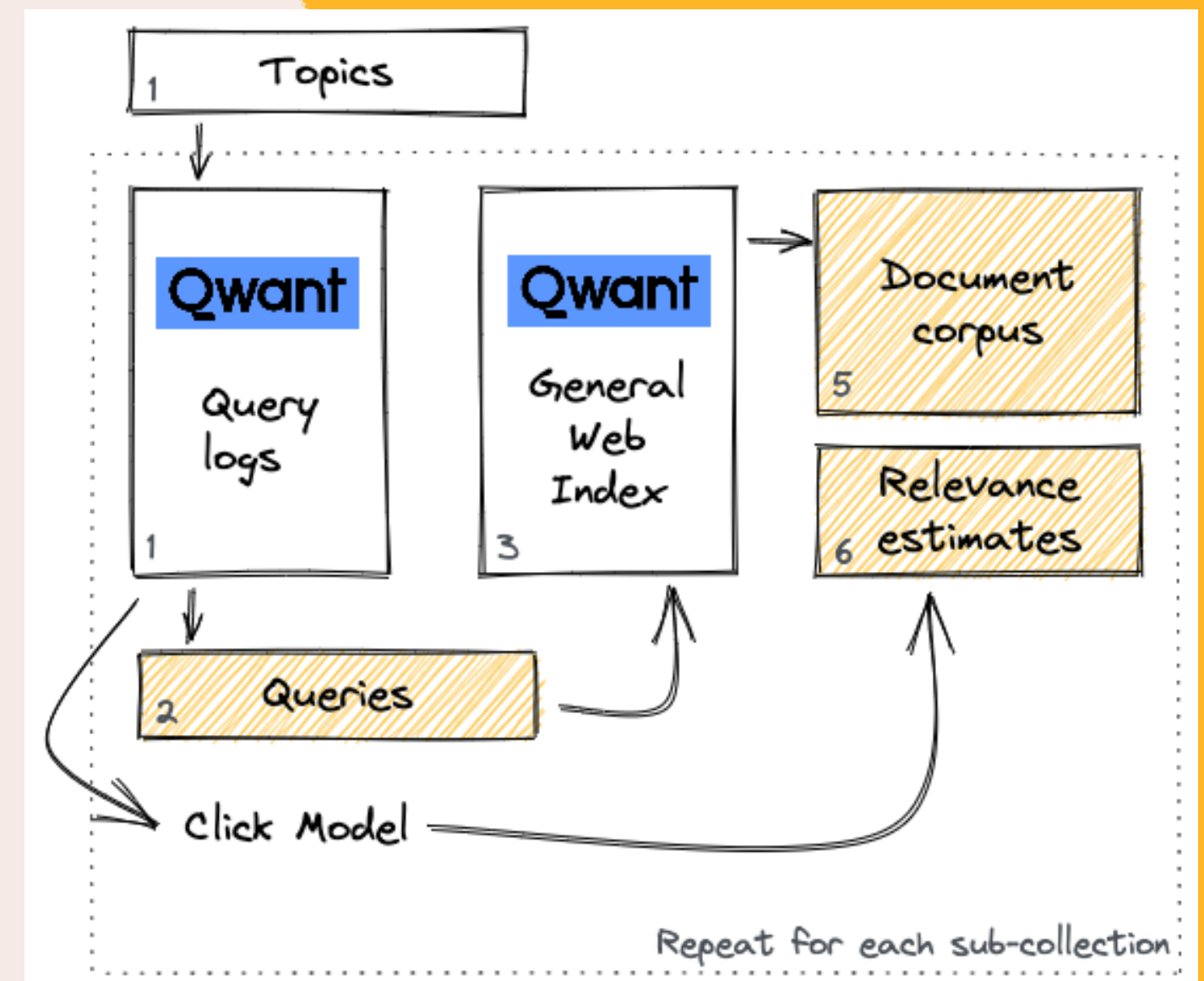


SEUPD@CLEF: DARDS

LongEval



CONTENTS

● ● Tools and components involved

● ● Strategies we applied and resulting systems

● ● Results: performance and highlights of our systems

● ● What we've learned and how could this evolve

OUR PATH TO LONGEVAL'S GOALS

1

A starting point

Build a baseline, simple
IR system with good
performance

2

Continuous improvement

Analysis and ideas turned
into experiments,
be they good or bad!

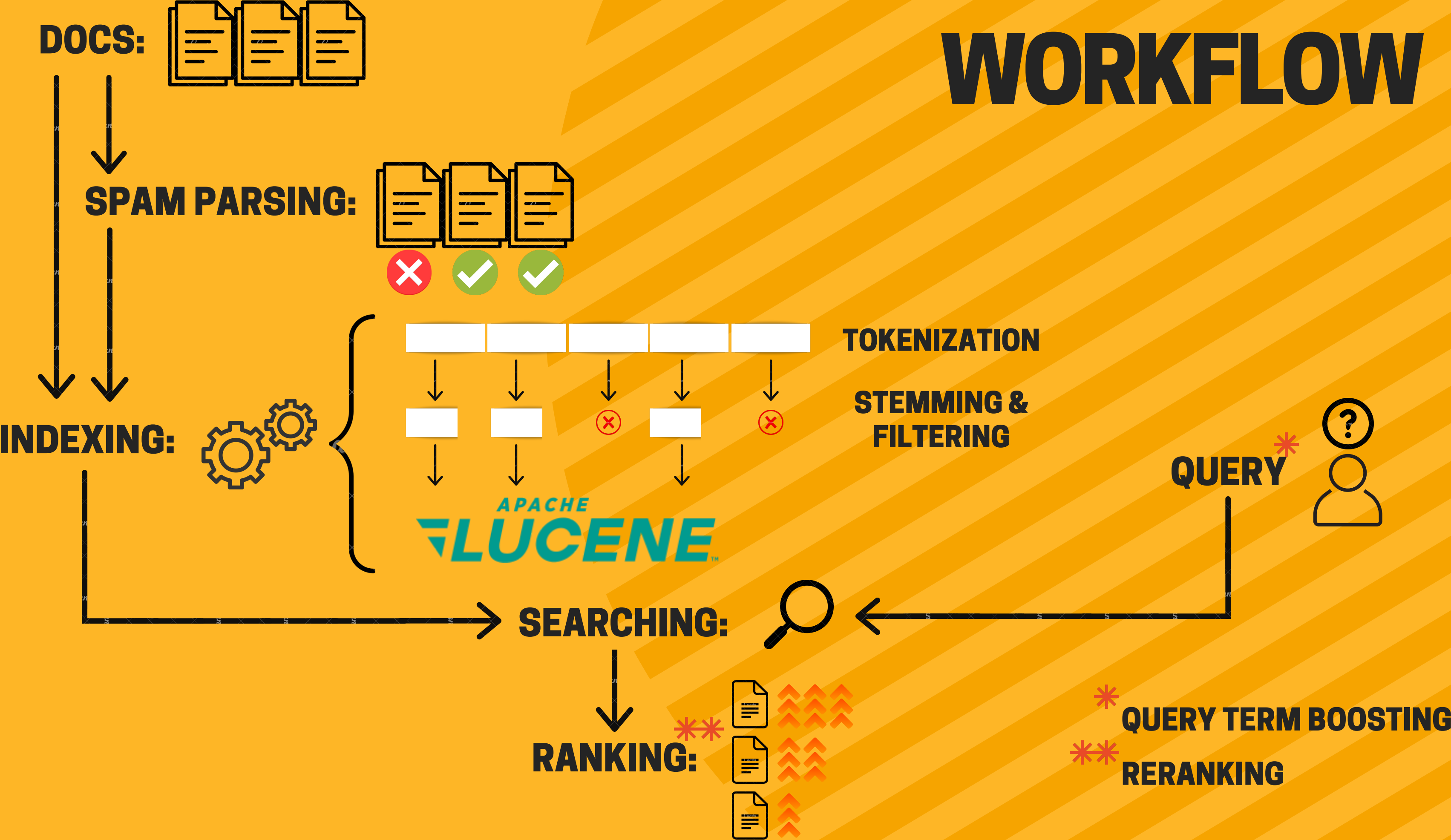
3

End of the journey?

Where we've managed
to come, results and
opportunities

Development of long-term stable IR systems!

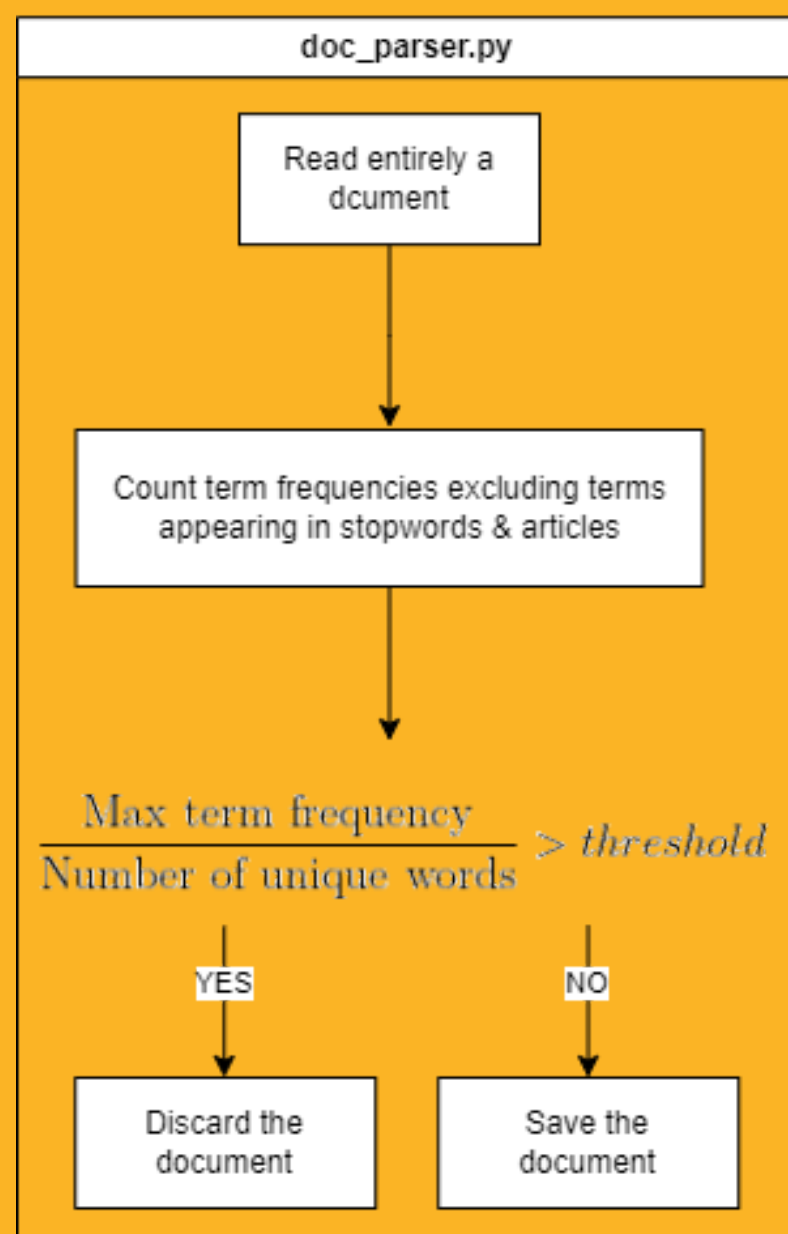
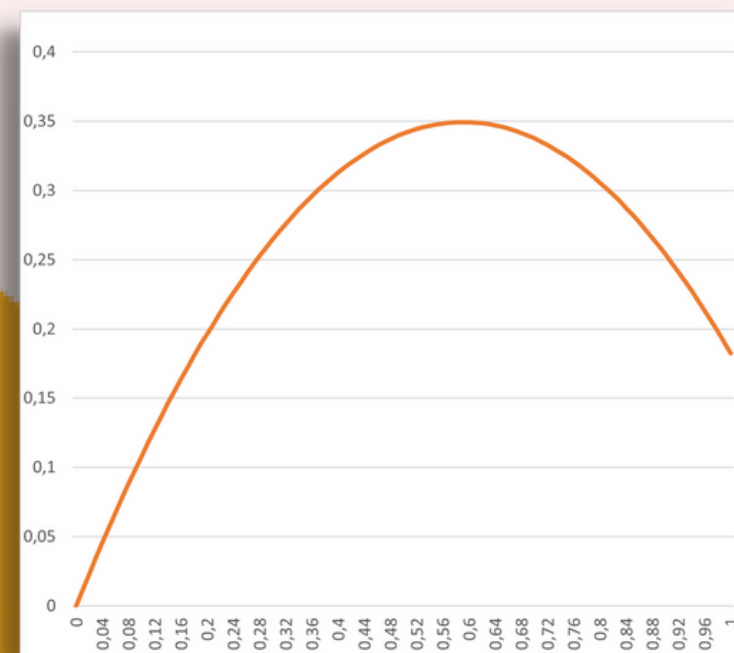
WORKFLOW





FIGHT AGAINST SPAM

EXPECTED BEHAVIOUR



stopwords & articles



doc_parser.py

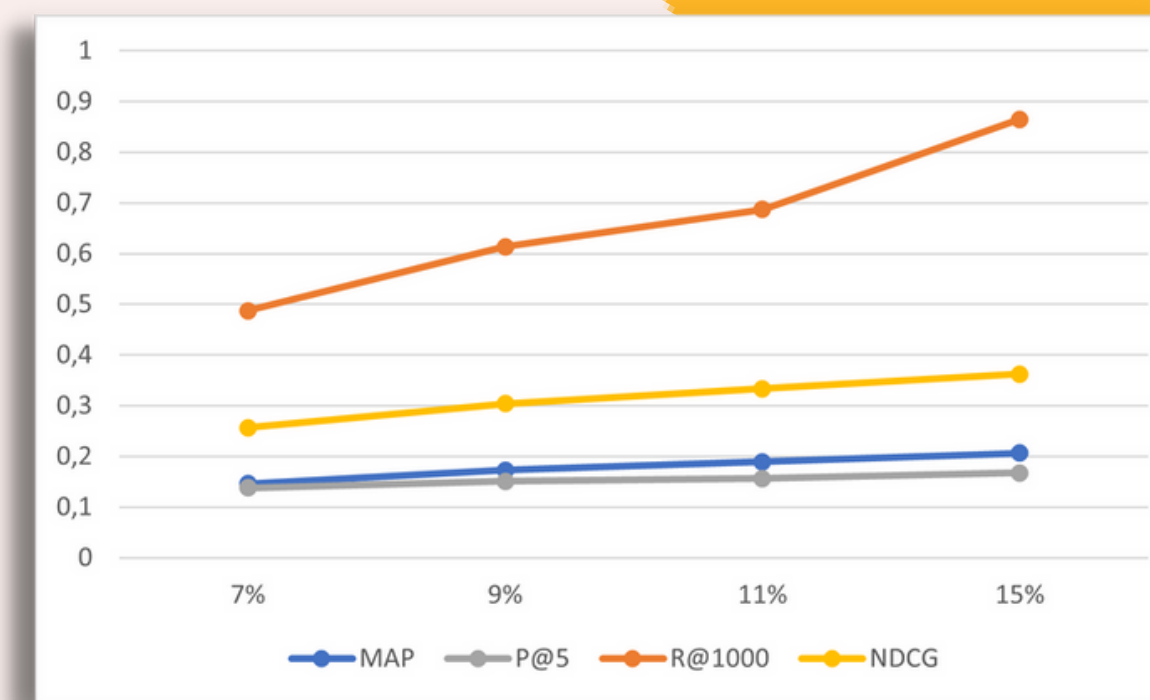


documents



parsed documents

ACTUAL BEHAVIOUR



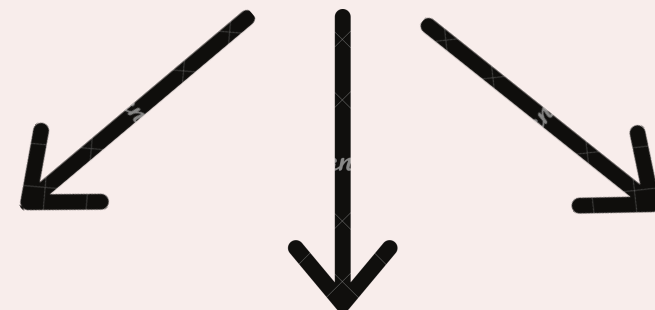
DOCS EXAMPLE

"Panneaux d'indication du code de la route
Panneaux de signalisation du code de la route
Vous pourrez trouver sur cette page, les images et définitions des différents signaux et panneaux de signalisation du code de la route Code de la route 100% gratuit ! Créez votre compte en 2 min et testez vos connaissances du code de la route Je m'inscris! Code de la route 2022."

© Copyright 2010-2022."

INPUT'S STRUCTURE

**DOCS
STRUCTURE**



ID

BODY

URL

(optional)

**QUERY
STRUCTURE**



**TITLE
(.tsv)**

Trec format

ANALYZER & FILTERS

Tokenizer: StandardTokenizer (in all systems)

SHARED

LowerCaseFilter
StopFilter
SynonymGraphFilter
NumberFilter

FRENCH SYSTEMS

ElisionFilter
ASCIIFoldingFilter
FrenchLightStemFilter

ENGLISH SYSTEMS

NGramFilter
ShingleFilter
PorterStemFilter
KStemFilter

NumberFilter: deletes all the number tokens (based on the TypeAttribute token attribute)

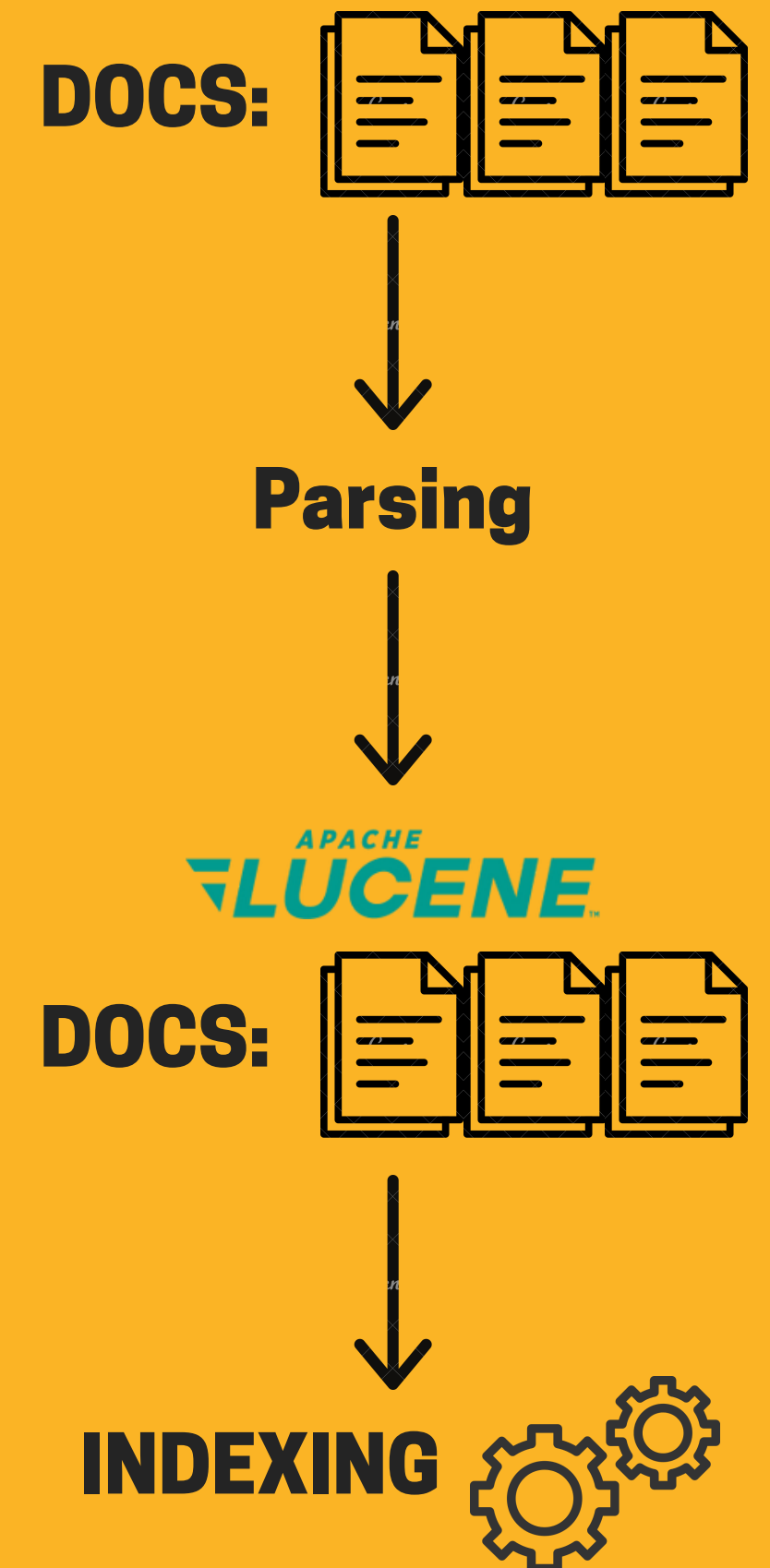
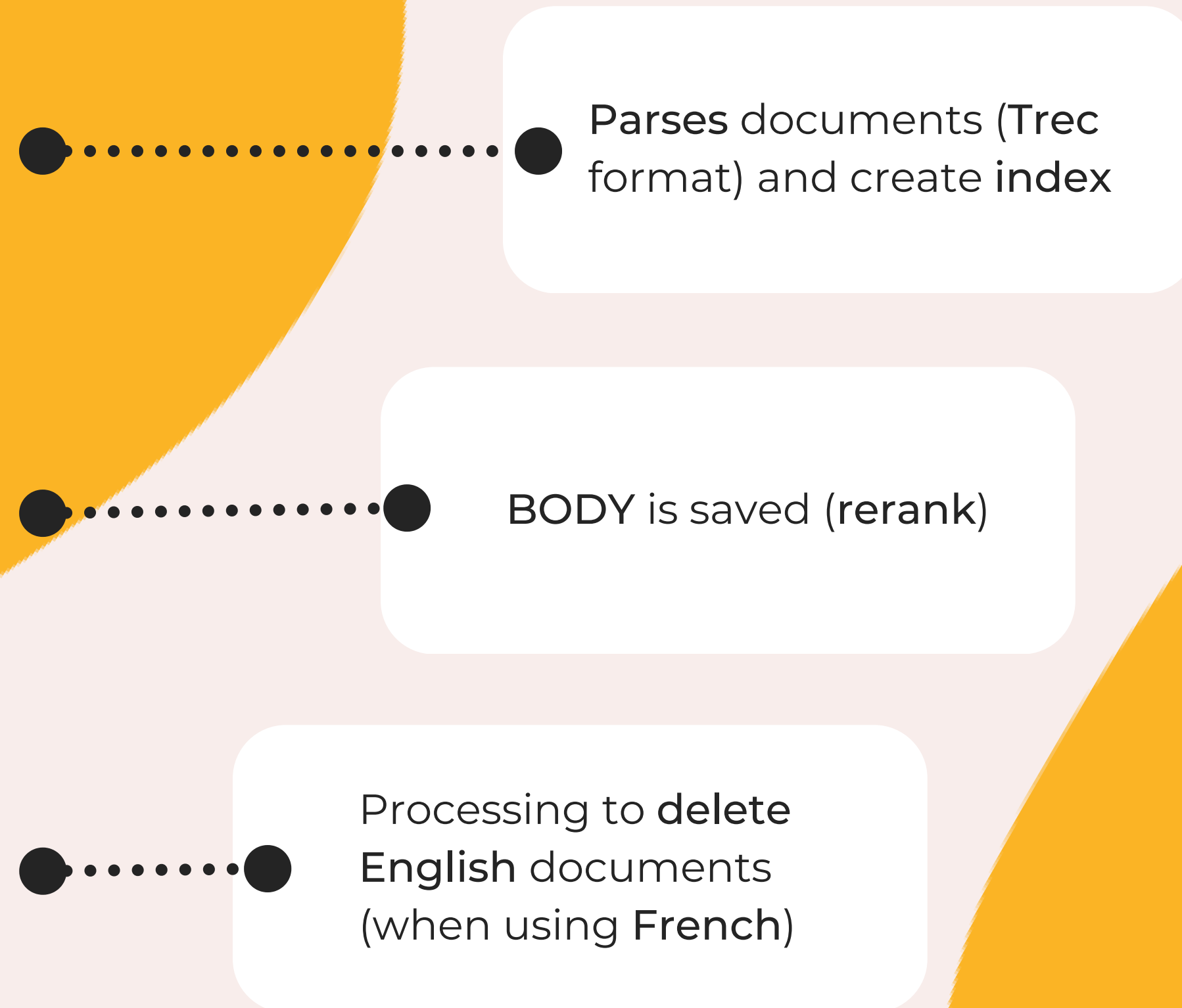
Stoplists: custom based on stoplists found online and on index words

ElisionFilter: removes articles, prepositions and conjunctions usually connected by an apostrophe or hyphen (J'aime)

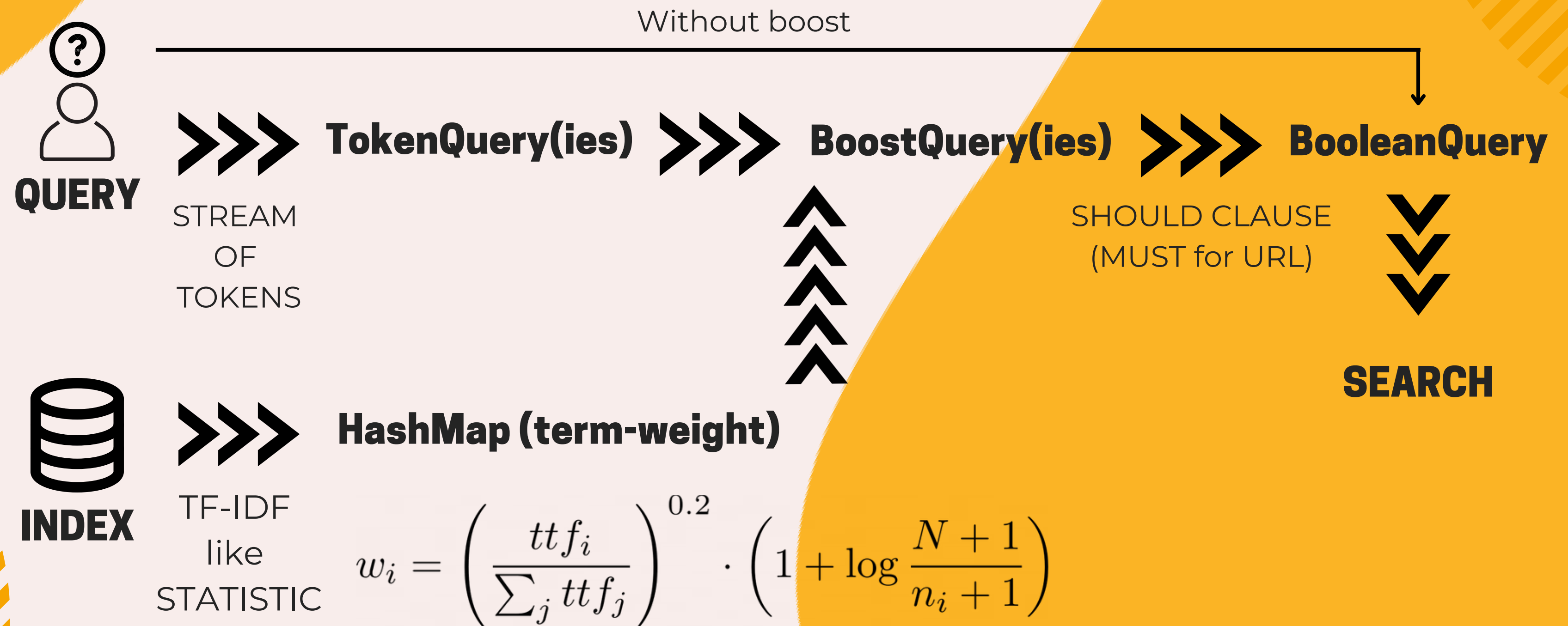
ASCIIFoldingFilter: converts unicode characters in the first 127 ASCII characters (due to accents and diacritical marks in French)

Indexer

Almost standard indexer:



SEARCHER & BOOSTING



RERANK

RERANK by REINDEXING: we tried to change the index statistics to push relevant documents



(usually with same query, but not always)

We reranked the TOP-100 documents , why?

RECALL@100	66.67%
RECALL@200	72.29%
RECALL@500	79.75%
RECALL@1000	84.37%

Since the majority of relevant retrieved documents was in the first 100 documents we kept the number of documents reranked low to try to avoid to index a lot of NOT relevant documents .

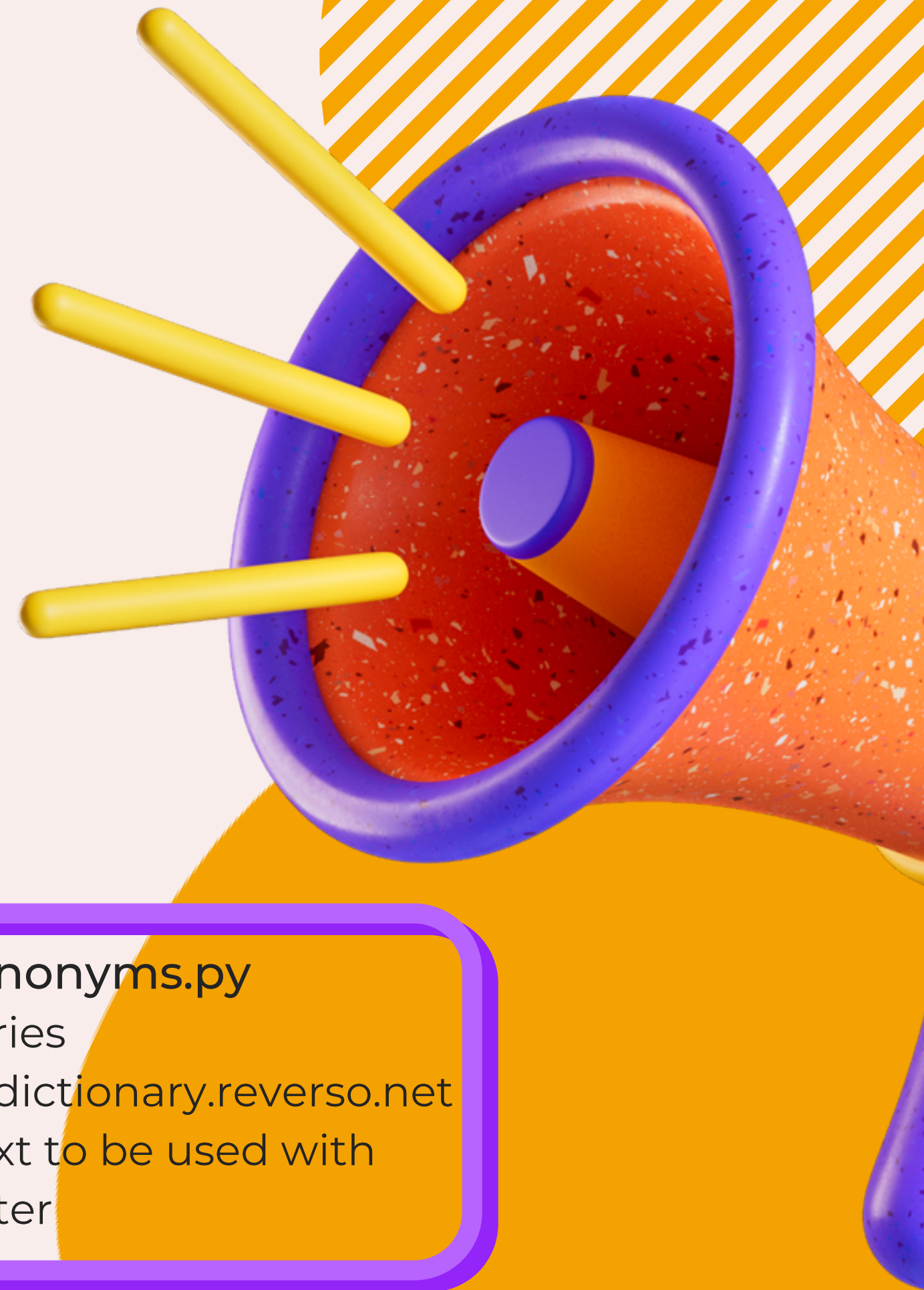
SIMILARITY

- Similarity functions compute documents' scores based on different statistics.
- MultiSimilarity combines evidence from multiple similarity functions.

System	MAP
BM25Similarity	0.1424
DFRSimilarity	0.1423
AxiomaticF2LOG	0.1358
AxiomaticF2EXP	0.1346
LMJelinekMercerSimilarity	0.1255
LMDirichletSimilarity	0.1204
ClassicSimilarity	0.0740
IndriDirichletSimilarity	0.0137
BooleanSimilarity	0.0121

Values refer to an initial system that performed the search on the English corpus.

QUERY EXPANSION



- Expands queries using synonyms
- Sometimes combined with reranking
- Query Drift
- Contextualized synonyms (overfitting)

`search_synonyms.py`

- parses french queries
- makes request to [dictionary.reverso.net](https://www.dictionnaire.reverso.net/)
- writes `synonyms.txt` to be used with `SynonymGraphFilter`

System	Filters	NDCG	TIME
BM25FRENCHBOOSTURL	ElisionFilter, LowerCaseFilter, StopFilter, ASCIIFoldingFilter, FrenchLightStemFilter	0.3815	MEDIUM
BM25FRENCHBASE	same as BM25FRENCHBOOSTURL	0.3812	FAST
BM25FRENCHRERANK100	same as BM25FRENCHBOOSTURL	0.3657	MEDIUM
BM25FRENCHDOCEXPANSION	ElisionFilter, LowerCaseFilter, StopFilter, SynonymGraphFilter, FlattenGraphFilter, ASCIIFoldingFilter, FrenchLightStemFilter, RemoveDuplicatesTokenFilter	0.3650	MEDIUM
BM25FRENCHSPAM	same as BM25FRENCHBOOSTURL	0.3623	SLOW
BM25FRENCHQUERYEXPANSION	ElisionFilter, LowerCaseFilter, StopFilter, SynonymGraphFilter, ASCIIFoldingFilter, FrenchLightStemFilter	0.3567	FAST
BM25TRANSLATEDQUERIES	LowerCaseFilter, StopFilter, KStemFilter	0.3037	MEDIUM



WHAT
WE'VE
ACHIEVED

System	NDCG WT	NDCG ST	NDCG LT	MAP	Recall @1000
BM25FRENCHBOOSTURL	0.3859	0.3866	0.3945	0.2152	0.8421
BM25FRENCHBASE	0.3843	0.3924	0.3916	0.2146	0.8451
BM25FRENCHRERANK100	0.3755	0.3756	0.3758	0.1960	0.8437
BM25FRENCHSPAM	0.3605	0.3680	0.3643	0.2067	0.7648
BM25TRANSLATEDQUERIES	0.3072	0.3051	0.3189	0.1523	0.7437

MAIN PROBLEMS WE HAVE FACED

Time sustainability

Advanced algorithms could not be implemented (e.g. RAKE, NLP)

Translation difficulties

Translation induced noise and its interference with IR

Ground truth quality

Is the Oracle really always right?

Non-uniform translation example

Item ID	French	English
q0622311 doc062200210641	bourse de l emploi public "Sélectionnée par Emploi Public"	Public Employment Exchange "Selected by Public servant"

ENGLISH-BASED SYSTEMS

Query translation error examples

Query ID	French	English
q062213307	cuisson gigot agneau	leg leg
q062228	aeroport bordeaux	airport

The translation of the queries is performed in the searcher, after parsing and before tokenization.

Developed tool: Google App Script platform based, deployed as a web application. It is used as a **REST resource**.

Non uniform translation example

Impact of different translators

Query translation error example



CONCLUSIONS

Understanding of possible future evolutions of the system

Main achievements:

0.382

nDCG and *recall*
good results
(BM25FRENCHBOOSTURL)

9

Multiple techniques
tested - no
overfitting on the
training data

1

Development of a
translation tool to
improve
effectiveness

1

Development of a
Python algorithm to
target SPAM

Further development:
doc. length, word freq.
distribution, variance,
sentence, language

4

Ideas for future
development and
understanding of
possible evolutions
of the system

FUTURE WORK



Improve translation

Execute the documents BODY field translation in a computationally acceptable runtime



Systems combination

Perform multilingual search



Learn To Rank (LTR)

Implementation of LTR techniques



ML techniques implementation

e.g.
threshold/weights optimisation or SPAM detection improvement

**THANK
YOU!**

