Technology
Arts Sciences
TH Köln

Mini-TREC Challenge

Elastic Search Engine for locating relevant documents on a COVID-19 Dataset

Search Engine Technology
Wintersemester 20/21 DIS-17.1
Project Group Placeholder

Lecturer
Dr. Johann (Wanja) Schaible

Attendees
Constantin Krah
Andreas Kruff
Joshua Thos
Anh Huy Matthias Tran

Agenda

Project Planning

1. Used Tools
2. Timeline

Procedure

1. MVP Development
2. Workflow of the final Build

Ranking Optimization

1. Index Pipeline
2. Language Processing
3. Segmentation
4. Similarity Moduls
5. Reranking
6. Query Pipeline

Results

1. Measuring Performance
2. Conclusion
3. How to further improve

01 —

Project Planning

1.1 Used Tools

https://github.com/AH-Tran/DIS17.1-Suchmaschinentechnologie

https://www.docker.com/

https://www.elastic.co/de/elasticsearch/

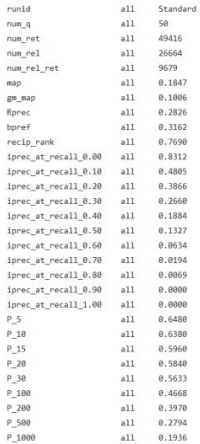
https://www.elastic.co/de/kibana

https://www.python.org/

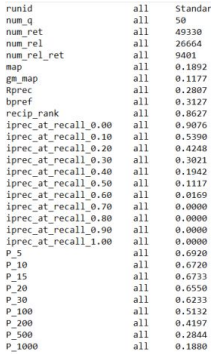
1.2 Timeline

Experiment Evaluation: Recall, Precision @ 5, 10, Recip_rank as FOCUS

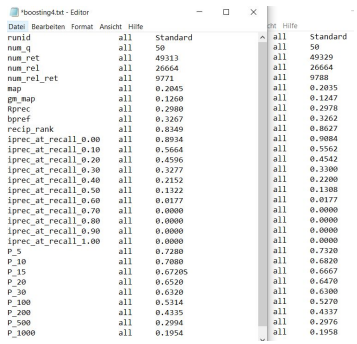
MVP	Index + Query	Boosting	Reranking 1 & 2	Reranking 3
01.11.2020	10.01.2020	20.01.2021	22.01.2021	23.01.2021
Only Abstract + Title	Untuned Boosting + Synonyms	Adjusting Weights	Improved Synonyms for best Recall Results	Reranking + Query Specific Cluster
	LMJ DFR		LMJ DFR BM25	



runid	all	Standard
num_q	all	50
num_ret	all	49416
num_rel	all	26664
num_rel_ret	all	9679
map	all	0.1847
gm_map	all	0.1006
Rprec	all	0.2826
bpref	all	0.3162
recip_rank	all	0.7690
iprec_at_recall_0.00	all	0.8312
iprec_at_recall_0.10	all	0.4805
iprec_at_recall_0.20	all	0.3866
iprec_at_recall_0.30	all	0.2660
iprec_at_recall_0.40	all	0.1884
iprec_at_recall_0.50	all	0.1327
iprec_at_recall_0.60	all	0.0634
iprec_at_recall_0.70	all	0.0194
iprec_at_recall_0.80	all	0.0069
iprec_at_recall_0.90	all	0.0000
iprec_at_recall_1.00	all	0.0000
P_5	all	0.6480
P_10	all	0.6380
P_15	all	0.5960
P_20	all	0.5840
P_30	all	0.5633
P_100	all	0.4668
P_200	all	0.3970
P_500	all	0.2794
P_1000	all	0.1936



runid	all	Standard
num_q	all	50
num_ret	all	49330
num_rel	all	26664
num_rel_ret	all	9401
map	all	0.1892
gm_map	all	0.1177
Rprec	all	0.2807
bpref	all	0.3127
recip_rank	all	0.8627
iprec_at_recall_0.00	all	0.9076
iprec_at_recall_0.10	all	0.5390
iprec_at_recall_0.20	all	0.4248
iprec_at_recall_0.30	all	0.3021
iprec_at_recall_0.40	all	0.1942
iprec_at_recall_0.50	all	0.1117
iprec_at_recall_0.60	all	0.0169
iprec_at_recall_0.70	all	0.0000
iprec_at_recall_0.80	all	0.0000
iprec_at_recall_0.90	all	0.0000
iprec_at_recall_1.00	all	0.0000
P_5	all	0.6920
P_10	all	0.6720
P_15	all	0.6733
P_20	all	0.6550
P_30	all	0.6233
P_100	all	0.5132
P_200	all	0.4197
P_500	all	0.2844
P_1000	all	0.1880

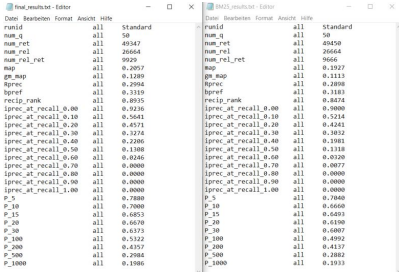


*boosting4.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

runid	all	Standard
num_q	all	50
num_ret	all	49313
num_rel	all	26664
num_rel_ret	all	9771
map	all	0.2045
gm_map	all	0.1260
Rprec	all	0.2980
bpref	all	0.3267
recip_rank	all	0.8349
iprec_at_recall_0.00	all	0.8934
iprec_at_recall_0.10	all	0.5664
iprec_at_recall_0.20	all	0.4596
iprec_at_recall_0.30	all	0.3277
iprec_at_recall_0.40	all	0.2152
iprec_at_recall_0.50	all	0.1322
iprec_at_recall_0.60	all	0.0177
iprec_at_recall_0.70	all	0.0000
iprec_at_recall_0.80	all	0.0000
iprec_at_recall_0.90	all	0.0000
iprec_at_recall_1.00	all	0.0000
P_5	all	0.7280
P_10	all	0.7080
P_15	all	0.67205
P_20	all	0.6520
P_30	all	0.6320
P_100	all	0.5314
P_200	all	0.4335
P_500	all	0.2994
P_1000	all	0.1954

	all	50
	all	49329
	all	26664
	all	9788
	all	0.2035
	all	0.1247
	all	0.2978
	all	0.3262
	all	0.8627
	all	0.9084
	all	0.5562
	all	0.4542
	all	0.3300
	all	0.2200
	all	0.1308
	all	0.0177
	all	0.0000
	all	0.0000
	all	0.0000
	all	0.0000
	all	0.7320
	all	0.6820
	all	0.6667
	all	0.6470
	all	0.6300
	all	0.5270
	all	0.4337
	all	0.2976
	all	0.1958



final_results.txt - Editor

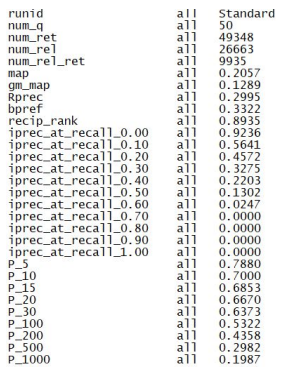
Datei Bearbeiten Format Ansicht Hilfe

runid	all	Standard
num_q	all	50
num_ret	all	49347
num_rel	all	26664
num_rel_ret	all	9929
map	all	0.2057
gm_map	all	0.1289
Rprec	all	0.2994
bpref	all	0.3319
recip_rank	all	0.8935
iprec_at_recall_0.00	all	0.9236
iprec_at_recall_0.10	all	0.5641
iprec_at_recall_0.20	all	0.4571
iprec_at_recall_0.30	all	0.3274
iprec_at_recall_0.40	all	0.2206
iprec_at_recall_0.50	all	0.1308
iprec_at_recall_0.60	all	0.0246
iprec_at_recall_0.70	all	0.0000
iprec_at_recall_0.80	all	0.0000
iprec_at_recall_0.90	all	0.0000
iprec_at_recall_1.00	all	0.0000
P_5	all	0.7080
P_10	all	0.7000
P_15	all	0.6853
P_20	all	0.6670
P_30	all	0.6373
P_100	all	0.5322
P_200	all	0.4357
P_500	all	0.2984
P_1000	all	0.1986

BM25_results.txt - Editor

Datei Bearbeiten Format Ansicht Hilfe

runid	all	Standard
num_q	all	50
num_ret	all	49450
num_rel	all	26664
num_rel_ret	all	9666
map	all	0.1927
gm_map	all	0.1113
Rprec	all	0.2898
bpref	all	0.3103
recip_rank	all	0.8674
iprec_at_recall_0.00	all	0.9000
iprec_at_recall_0.10	all	0.5214
iprec_at_recall_0.20	all	0.4241
iprec_at_recall_0.30	all	0.3032
iprec_at_recall_0.40	all	0.1981
iprec_at_recall_0.50	all	0.1318
iprec_at_recall_0.60	all	0.0320
iprec_at_recall_0.70	all	0.0077
iprec_at_recall_0.80	all	0.0000
iprec_at_recall_0.90	all	0.0000
iprec_at_recall_1.00	all	0.0000
P_5	all	0.7040
P_10	all	0.6660
P_15	all	0.6493
P_20	all	0.6190
P_30	all	0.6007
P_100	all	0.4992
P_200	all	0.4137
P_500	all	0.2882
P_1000	all	0.1933



runid	all	Standard
num_q	all	50
num_ret	all	49348
num_rel	all	26663
num_rel_ret	all	9935
map	all	0.2057
gm_map	all	0.1289
Rprec	all	0.2995
bpref	all	0.3322
recip_rank	all	0.8935
iprec_at_recall_0.00	all	0.9236
iprec_at_recall_0.10	all	0.5641
iprec_at_recall_0.20	all	0.4572
iprec_at_recall_0.30	all	0.3275
iprec_at_recall_0.40	all	0.2203
iprec_at_recall_0.50	all	0.1302
iprec_at_recall_0.60	all	0.0247
iprec_at_recall_0.70	all	0.0000
iprec_at_recall_0.80	all	0.0000
iprec_at_recall_0.90	all	0.0000
iprec_at_recall_1.00	all	0.0000
P_5	all	0.7880
P_10	all	0.7000
P_15	all	0.6853
P_20	all	0.6670
P_30	all	0.6373
P_100	all	0.5322
P_200	all	0.4358
P_500	all	0.2982
P_1000	all	0.1987

Measuring Performance on Page. 20

02

Procedure

2.1 MVP Development

Preprocessing metadata.csv

- Identify documents with missing date
- Manually search for the dates of these documents
- Add the date to the specific paper via Python
- Add 01.01 to dates with only the year in it

Index metadata.csv

- Index the metadata.csv via Kibana interface

Execute all 50 queries in topic on the Index / Transform search results into trec_eval format

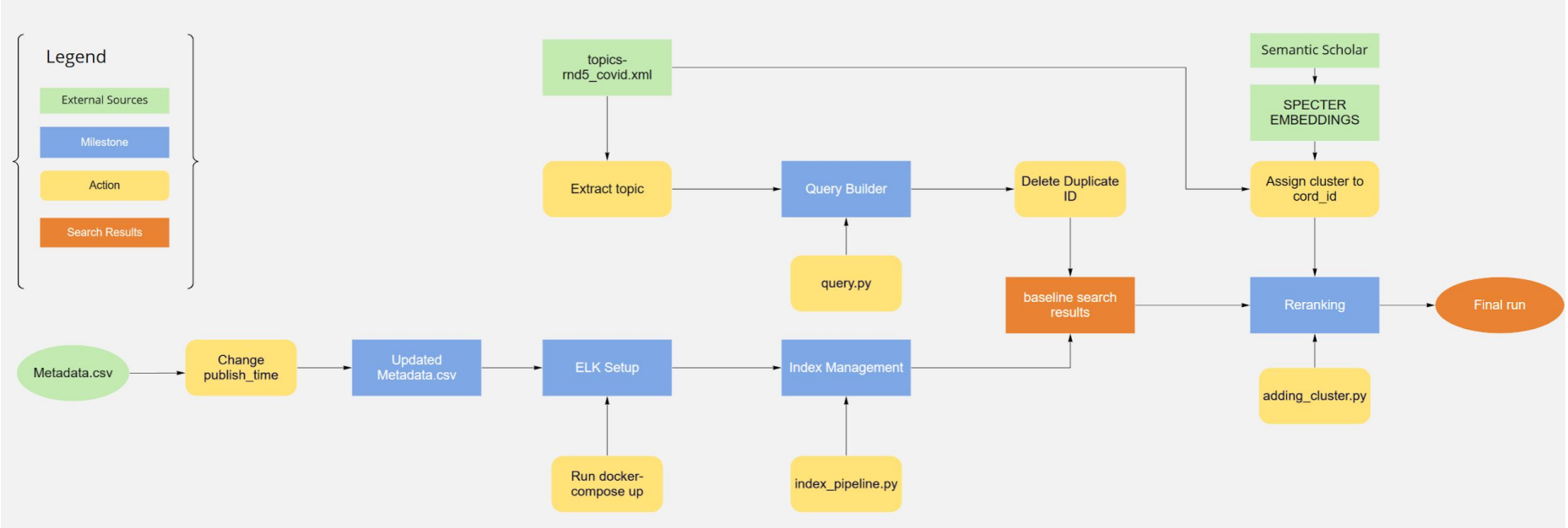
- Search for the given queries in the Index
- Drop duplicates within the same query results
- Transform the results into the trec_eval format so we can compare it with given qrels file

Evaluating the results with trec_eval

- Compare the resulting results.txt

runid	all	Standard
num_q	all	50
num_ret	all	49416
num_rel	all	26664
num_rel_ret	all	9679
map	all	0.1847
gm_map	all	0.1006
Rprec	all	0.2826
bpref	all	0.3162
recip_rank	all	0.7690
iprec_at_recall_0.00	all	0.8312
iprec_at_recall_0.10	all	0.4805
iprec_at_recall_0.20	all	0.3866
iprec_at_recall_0.30	all	0.2660
iprec_at_recall_0.40	all	0.1884
iprec_at_recall_0.50	all	0.1327
iprec_at_recall_0.60	all	0.0634
iprec_at_recall_0.70	all	0.0194
iprec_at_recall_0.80	all	0.0069
iprec_at_recall_0.90	all	0.0000
iprec_at_recall_1.00	all	0.0000
P_5	all	0.6480
P_10	all	0.6380
P_15	all	0.5960
P_20	all	0.5840
P_30	all	0.5633
P_100	all	0.4668
P_200	all	0.3970
P_500	all	0.2794
P_1000	all	0.1936

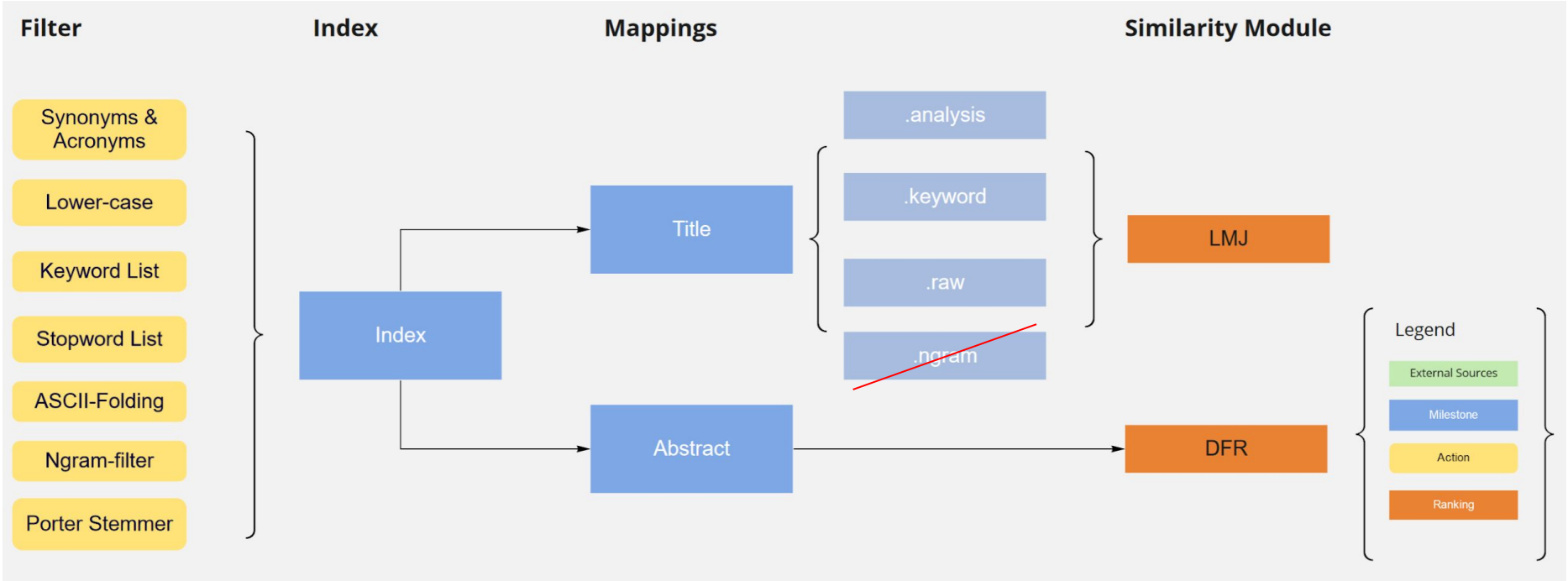
2.2 Workflow of the final Build



03 —

Ranking Optimization

3.1 Index Pipeline



3.2 Language Processing

Stopwords

- Delete stopwords with built-in functions from Elasticsearch
- Only remove english stop words

Synonyms

- Extract all topic, narrative and question for every query and remove stopwords, punctuation etc. with python
- Add synonyms
  - For common words:
    - to the resulting list with the wordnet python library
  - For medical terms:
    - from the American Chemical Society (CAS) COVID-19 Thesaurus based on certain string similarities
- Due to a lack of context we had to filter and combine synonyms manually

3.3 Segmentation | Tokenizing & Stemming

General

- Ascii II folding in the Analyzers to transform letters like á to a
- Transform all words to lowercase
- Define keywords that should not be separated like “Covid-19”
- Define characters that should be written as a word, like $ => dollar

```
GET /_analyze
{
  "tokenizer" : "standard",
  "filter" : ["asciifolding"],
  "text" : "açaí à la carte"
}

[ acai, a, la, carte ]
```

Tokenizing

- Tokenized with the standard built-in tokenizer from Elasticsearch
- The separator for each token is a whitespace

```
POST _analyze
{
  "tokenizer": "standard",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}

[ The, 2, QUICK, Brown, Foxes, jumped, over, the, lazy, dog's, bone ]
```

Stemming

- Compared Snowball Stemmer and the Porter Stemmer
- Snowball Stemmer is a much more aggressive than the Porter Stemmer
- Better results with the Porter Stemmer, which is the standard in Elasticsearch
- Only stemmed english words because of the queries

```
GET /_analyze
{
  "tokenizer": "standard",
  "filter": [ "stemmer" ],
  "text": "the foxes jumping quickly"
}

[ the, fox, jump, quickli ]
```

3.4 Similarity Modules

LMJ(Language Model based on the Jelinek Mercer Smoothing)

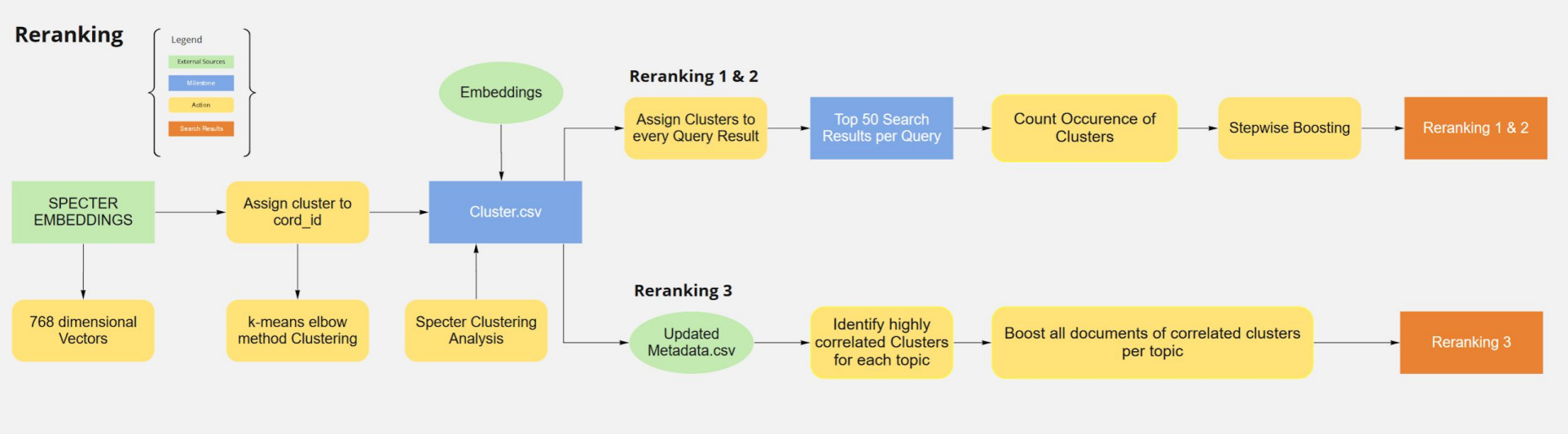
- Calculates numbers between zero and one for the similarity between text and query
- A number near zero is pretty good
- The closer the number is to one the more irrelevant is the document
- Performed best for short texts found in titles

DFR

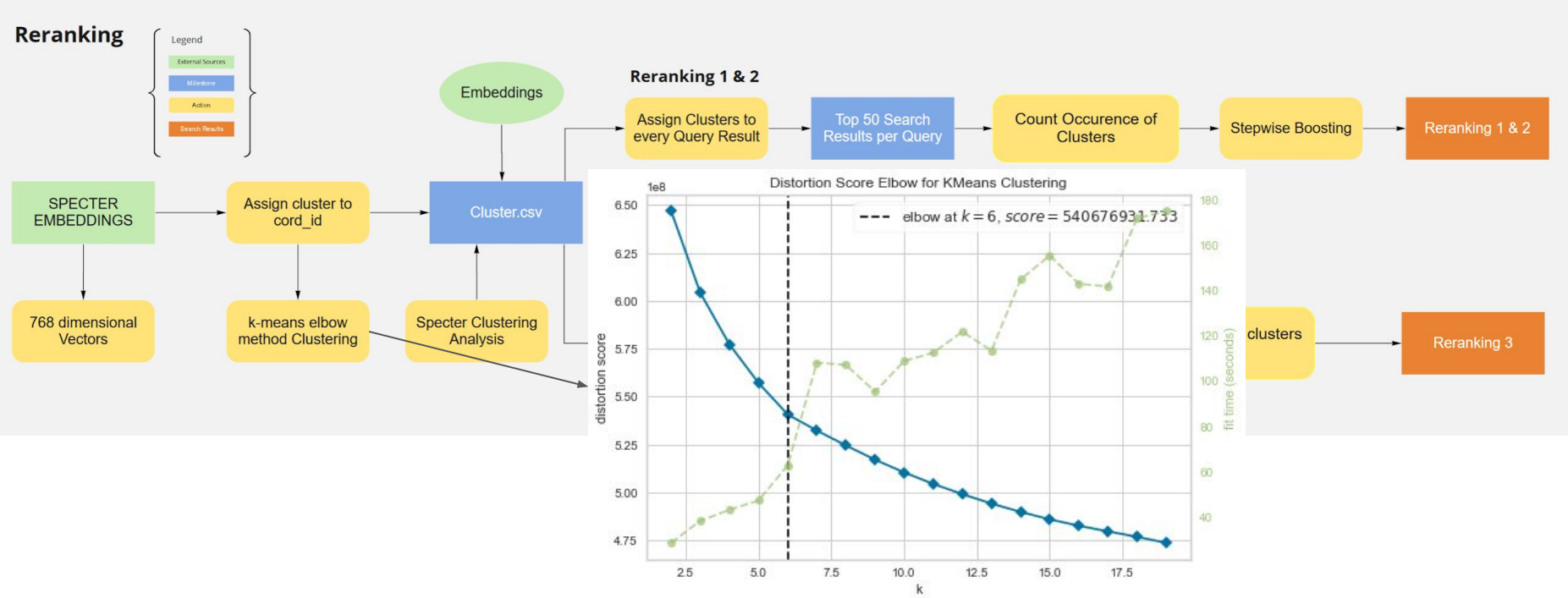
- Calculates with the Bose-Einstein approximation
- Performed best for longer, natural segments of texts found in abstracts

These Similarity Modules were also tested against BM25(Elasticsearch standard) and TF-IDF and performed overall the best.

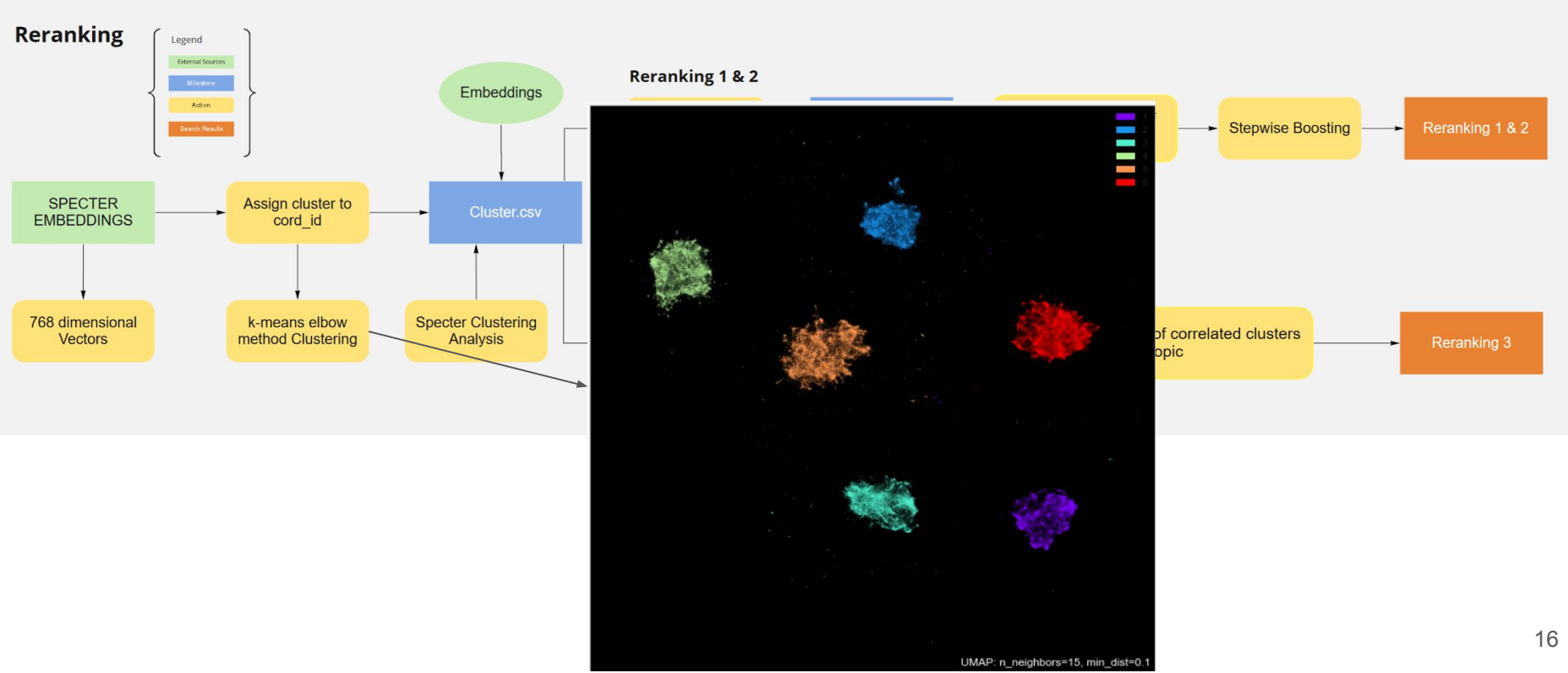
3.5 Reranking



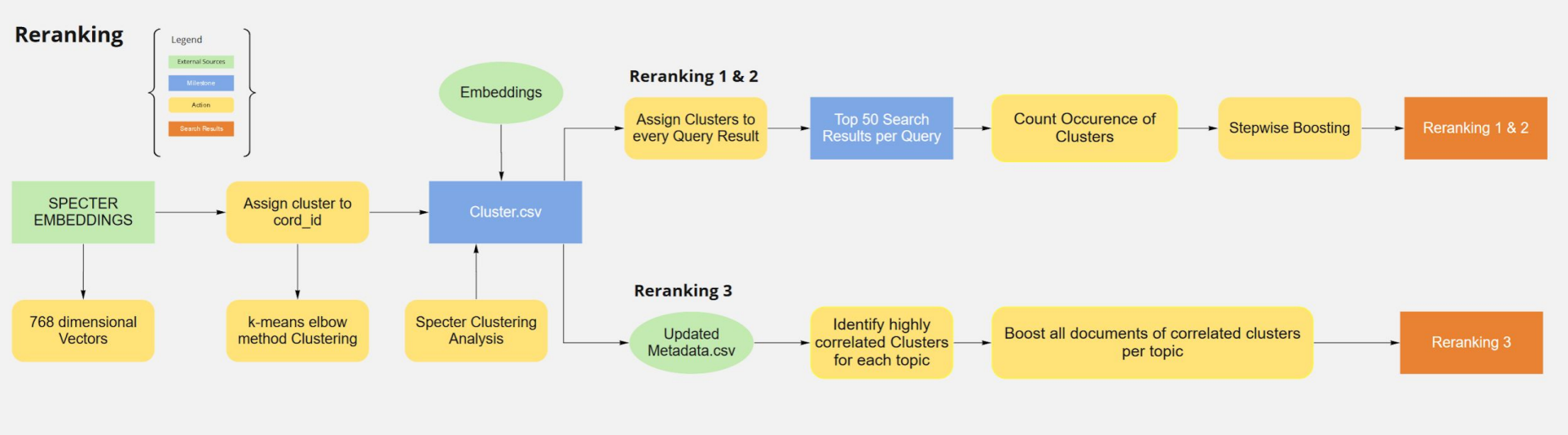
3.5 Reranking



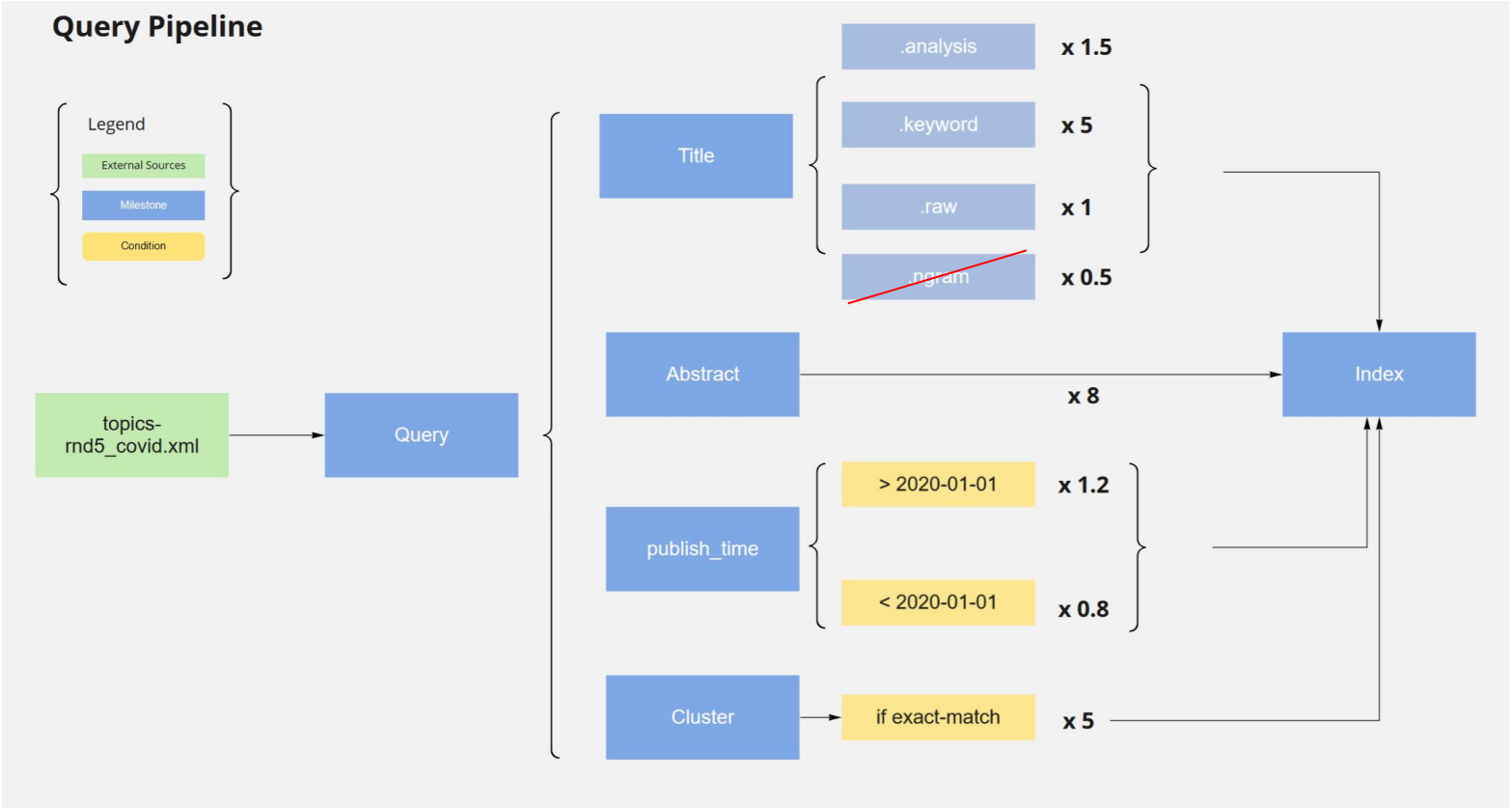
3.5 Reranking



3.5 Reranking



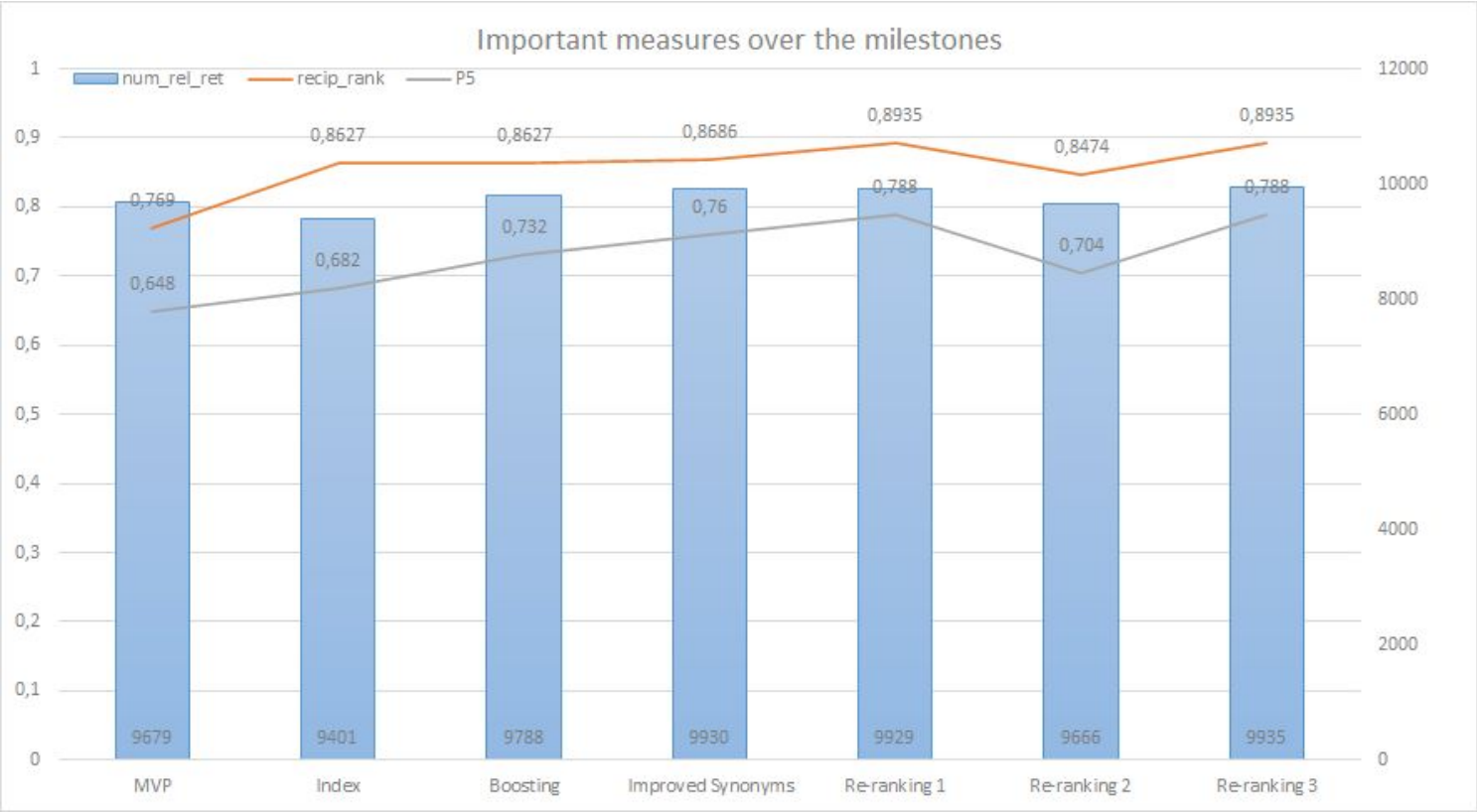
3.6 Query Pipeline



04 —

Results

4.1 Measuring Performance



4.1 Measuring Performance

	MVP	Index	Boosting	Improv. Synonyms	Re-ranking 1	Re-ranking 2	Re-ranking 3
num_q	50	50	50	50	50	50	50
num_ret	49416	49330	49329	49347	49347	49450	49348
num_rel	26664	26664	26664	26664	26664	26664	26664
num_rel_ret	**9679**	**9401**	**9788**	**9930**	**9929**	**9666**	**9935**
map	0.1847	0.1892	0.2035	0.2059	0.2057	0.1927	0.2057
Rprec	0.2826	0.2807	0.2978	0.3027	0.2994	0.2898	0.2995
recip_rank	**0.7690**	**0.8627**	**0.8627**	**0.8686**	**0.8935**	**0.8474**	**0.8935**
P_5	**0.6480**	**0.6820**	**0.7320**	**0.7600**	**0.7880**	**0.7040**	**0.7880**
P_10	0.6380	0.6720	0.6820	0.6980	0.7000	0.6660	0.7000
P_30	0.5633	0.6233	0.6300	0.6327	0.6373	0.6007	0.6373
P_1000	0.1936	0.1880	0.1958	0.1986	0.1986	0.1933	0.1987
Recall	0.3630	0.3526	0.3671	0.3724	0.3724	0.3625	0.3726

4.2 Conclusion

✚ Good approaches	➖ Bad approaches (our specific use case)
Reranking 1 & 2	Reranking 3 (Just 6 more rel_ret results)
Boosting (Publish_time, multi-fields)	Index-Management with untuned Boosting
Porter Stemmer	Snowball Stemmer
Field specific ranking models (DFR, LMJ)	Standard Ranking Models BM25, TF-IDF
Index expansion with synonyms	Query expansion with synonyms
	Ngrams

4.3 How to further improve

- Query-specific boosting & synonyms/query expansion
- Machine Learning Approach to create synonyms without contextual problems
- Cleaning the metadata (publish_time) to facilitate easier optimization of boosting
- Test & compare Stemming vs Lemmatization
- Automatization of testing procedures to determine optimal weights, boost and values

Vielen Dank !