University of Glasgow | Università di Pisa

CIKM 2021 1-5 NOVEMBER

IR From Bag-of-words to BERT and Beyond through Practical Experiments

A CIKM 2021 tutorial with PyTerrier and OpenNIR

Sean MacAvaney*
Craig Macdonald*
Nicola Tonellotto*

*(*Alphabetical ordering)*


Part 3: Contemporary Retrieval Architectures:
Neural re-rankers such as
BERT, T5, EPIC, ColBERT
Inverted index augmentation approaches such as
Doc2Query and DeepCT

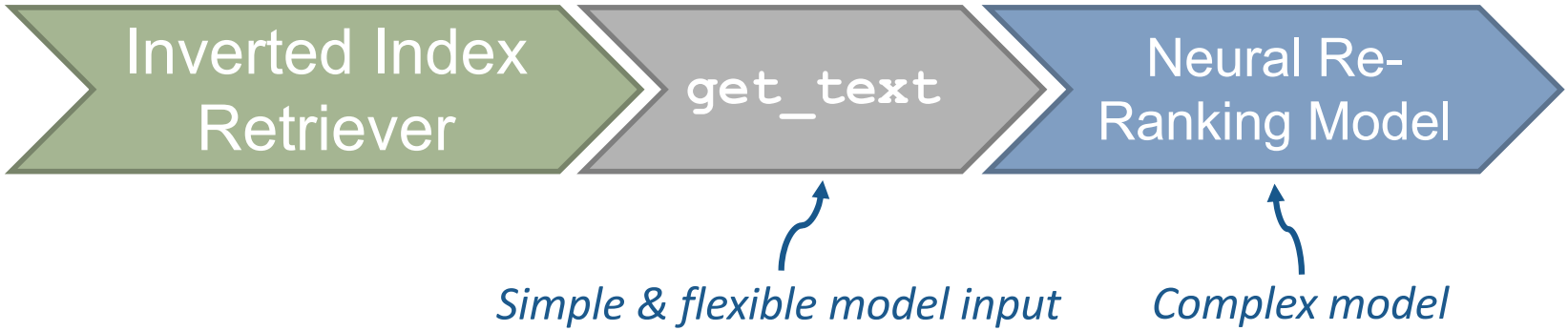
Moving From Features to Text

Università di Pisa | University of Glasgow

Learning to rank approaches require you to design effective features:

Performance heavily depends on these features

Neural methods adapted from NLP can learn patterns from the text itself, reducing the need for manually designed features:



Moving from Features to Text

Università di Pisa | University of Glasgow

In this session, we will cover:

- the basics of neural re-ranking methods.
- some of the current SOTA methods using BERT and T5 for ranking.
- approaches for reducing query-time cost, including pre-computation and inverted index augmentation

You will experience:

- re-ranking with models like BERT and T5 using PyTerrier and OpenNIR.
- performing re-ranking thresholding tuning.
- re-ranking with pre-computed representations to reduce query latency.
- performing index augmentation approaches like DeepCT and doc2query.

Intended Learning Outcomes

Università di Pisa | University of Glasgow

Part 3 – *Contemporary* Retrieval Architectures: Neural re-rankers such as BERT, EPIC, ColBERT and query augmentation approaches such as doc2query and DeepCT

ILO 3A. Understand contemporary retrieval architectures, such as using BERT, EPIC, ColBERT, and T5 as neural re-rankers.

ILO 3B. Understand how query time latency can be reduced by pre-computing representations or using neural models to augment an inverted index.

ILO 3C. Perform experiments with BERT, EPIC, T5, DeepCT, and doc2query in a Python notebook.

Outline of Part 3

Università di Pisa | University of Glasgow

Part 3A: Background

Part 3B: Contextualized Language Models (e.g., BERT)

Part 3C: Managing Efficiency (e.g., EPIC)

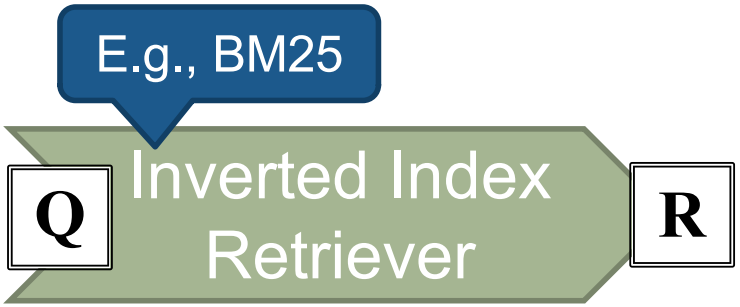
Part 3D: Neural Index Augmentation (e.g., doc2query)

Part 3E: Wrap up and move to practical session

Part 3A

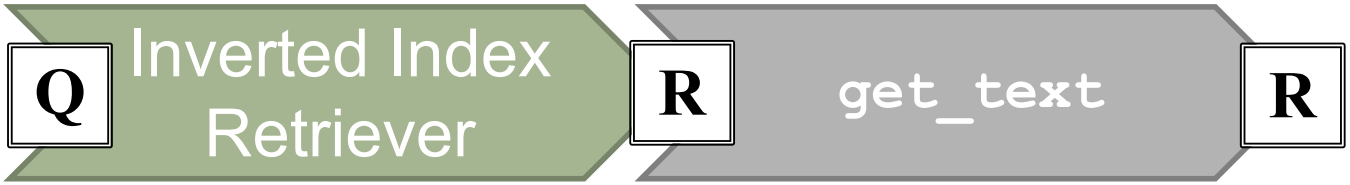
NEURAL RE-RANKING BACKGROUND

Moving from Features to Text



qid	query	docno	score
0	gold coast…	43242	0.1252
0	gold coast…	13746	0.1196
0	gold coast…	86454	0.0934
…	…	…	…

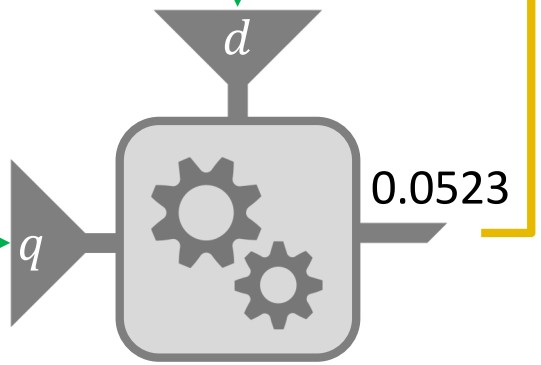
Moving from Features to Text



qid	query	docno	score	text
0	gold coast…	43242	0.1252	The Gold Coast is generally sunny and…
0	gold coast…	13746	0.1196	Temperatures in the Gold Coast of Aust…
0	gold coast…	86454	0.0934	Weather forecast: Australia, Gold Coast…
…	…	…	…	…

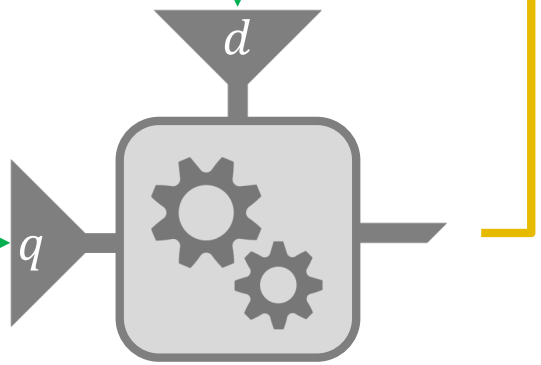
Moving from Features to Text

qid	query	docno	score	text
0	gold coast…	43242	0.0523	The Gold Coast is generally sunny and…
0	gold coast…	13746	0.1196	Temperatures in the Gold Coast of Aust…
0	gold coast…	86454	0.0934	Weather forecast: Australia, Gold Coast…
…	…	…	…	…



Moving from Features to Text

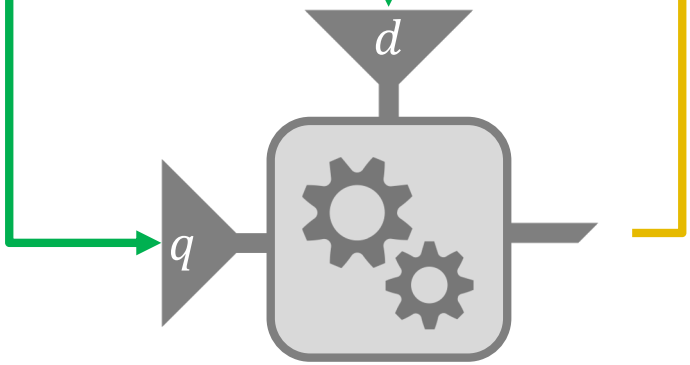
qid	query	docno	score	text
0	gold coast…	43242	0.0523	The Gold Coast is generally sunny and…
0	gold coast…	13746	0.8411	Temperatures in the Gold Coast of Aust…
0	gold coast…	86454	0.0934	Weather forecast: Australia, Gold Coast…
…	…	…	…	…



Moving from Features to Text

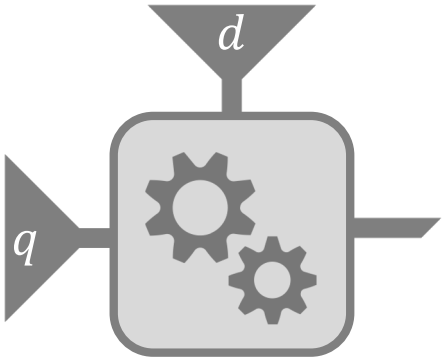
Università di Pisa | University of Glasgow

qid	query	docno	score	text
0	gold coast…	43242	0.0523	The Gold Coast is generally sunny and…
0	gold coast…	13746	0.8411	Temperatures in the Gold Coast of Aust…
0	gold coast…	86454	0.1535	Weather forecast: Australia, Gold Coast…
…	…	…	…	…



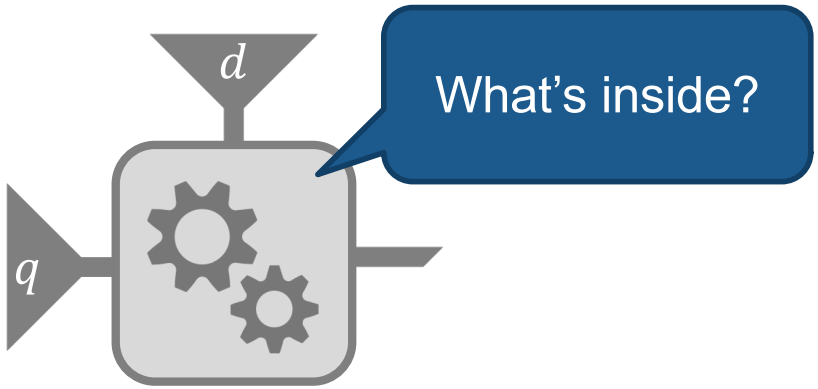
Moving from Features to Text

qid	query	docno	score	text
0	gold coast…	43242	0.0523	The Gold Coast is generally sunny and…
0	gold coast…	13746	0.8411	Temperatures in the Gold Coast of Aust…
0	gold coast…	86454	0.1535	Weather forecast: Australia, Gold Coast…
…	…	…	…	…



Moving from Features to Text

qid	query	docno	score	text
0	gold coast…	13746	0.8411	Temperatures in the Gold Coast of Aust…
0	gold coast…	86454	0.1535	Weather forecast: Australia, Gold Coast…
0	gold coast…	43242	0.0523	The Gold Coast is generally sunny and…
…	…	…	…	…



Representing Text

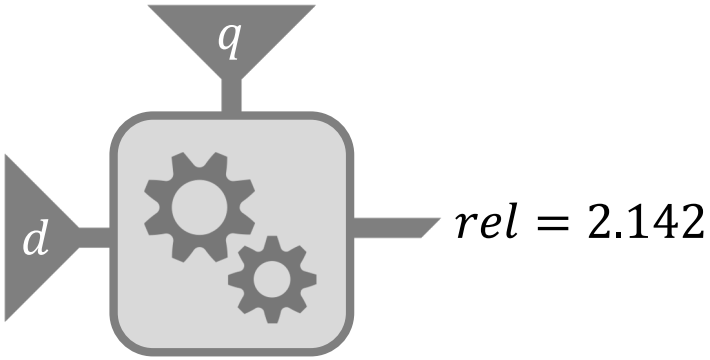
Università di Pisa | University of Glasgow

Coronavirus Early Symptoms

Document:

Title: How can we evaluate an interrelation of symptoms?

Abstract: A pandemic of 2019 novel coronavirus (COVID-19) is an international problem and factors associated with increased risk of mortality have been reported. However, there exists limited statistical method to estimate a comprehensive risk for a case in which a patient has several characteristics…



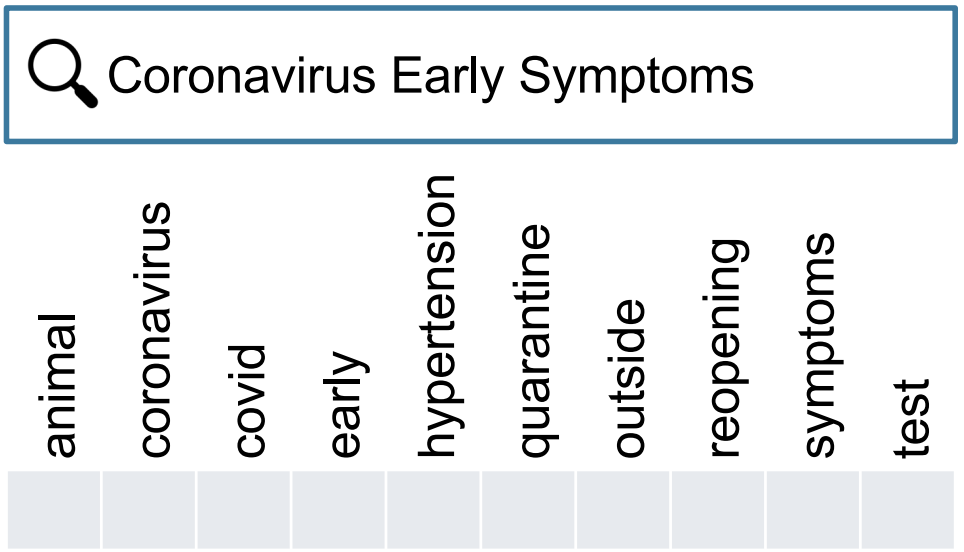
Representing Text

Option 1: One-hot encoding

animal	coronavirus	covid	early	hypertension	quarantine	outside	reopening	symptoms	test	...

Representing Text

Option 1: One-hot encoding



Representing Text

Option 1: One-hot encoding

Coronavirus Early Symptoms

	animal	**coronavirus**	covid	**early**	hypertension	quarantine	outside	reopening	**symptoms**	test	...
Bag of words		1		1					1		

Representing Text

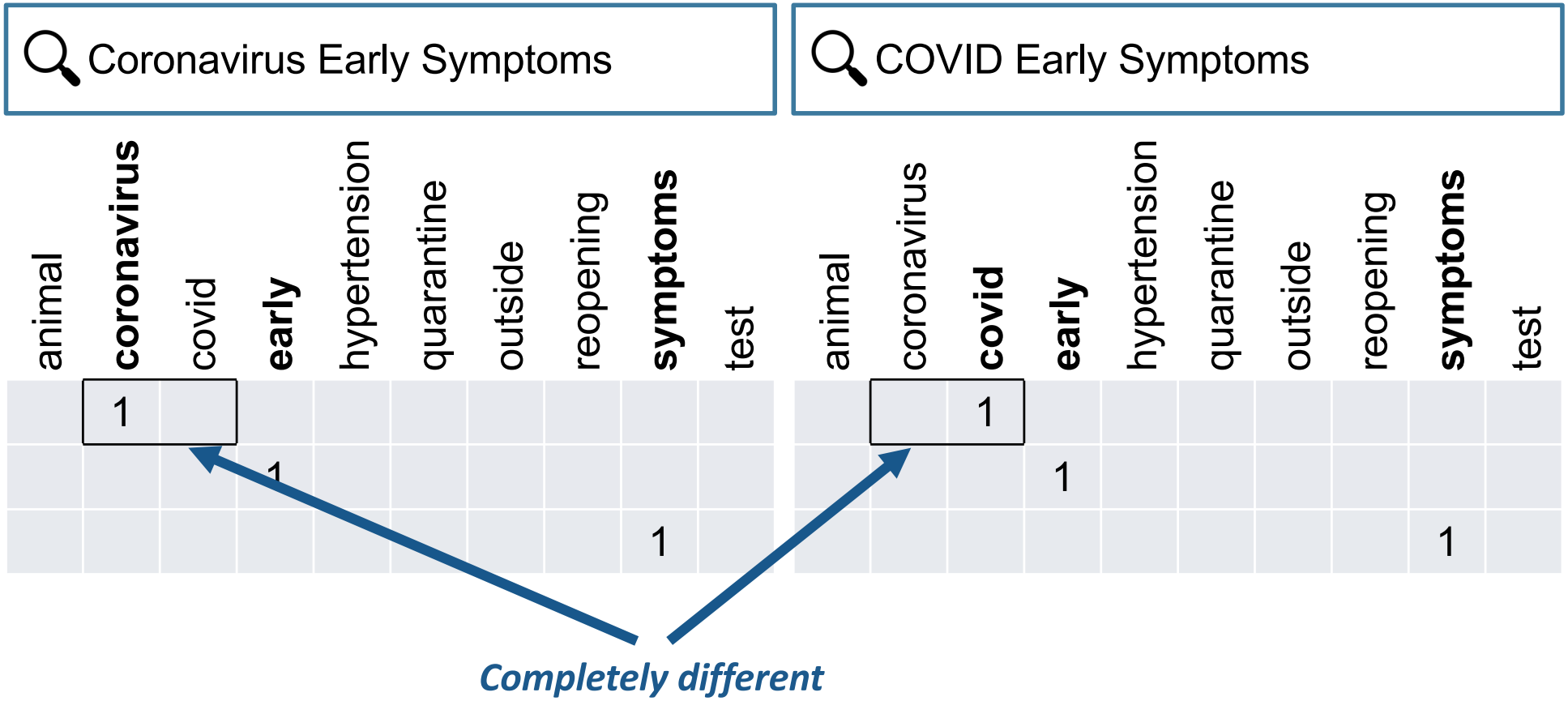
Option 1: One-hot encoding

Coronavirus Early Symptoms

	animal	**coronavirus**	covid	**early**	hypertension	quarantine	outside	reopening	**symptoms**	test	…
		1									
Sequence				1							
									1		

Representing Text

Problem: no relationship between words



Representing Text

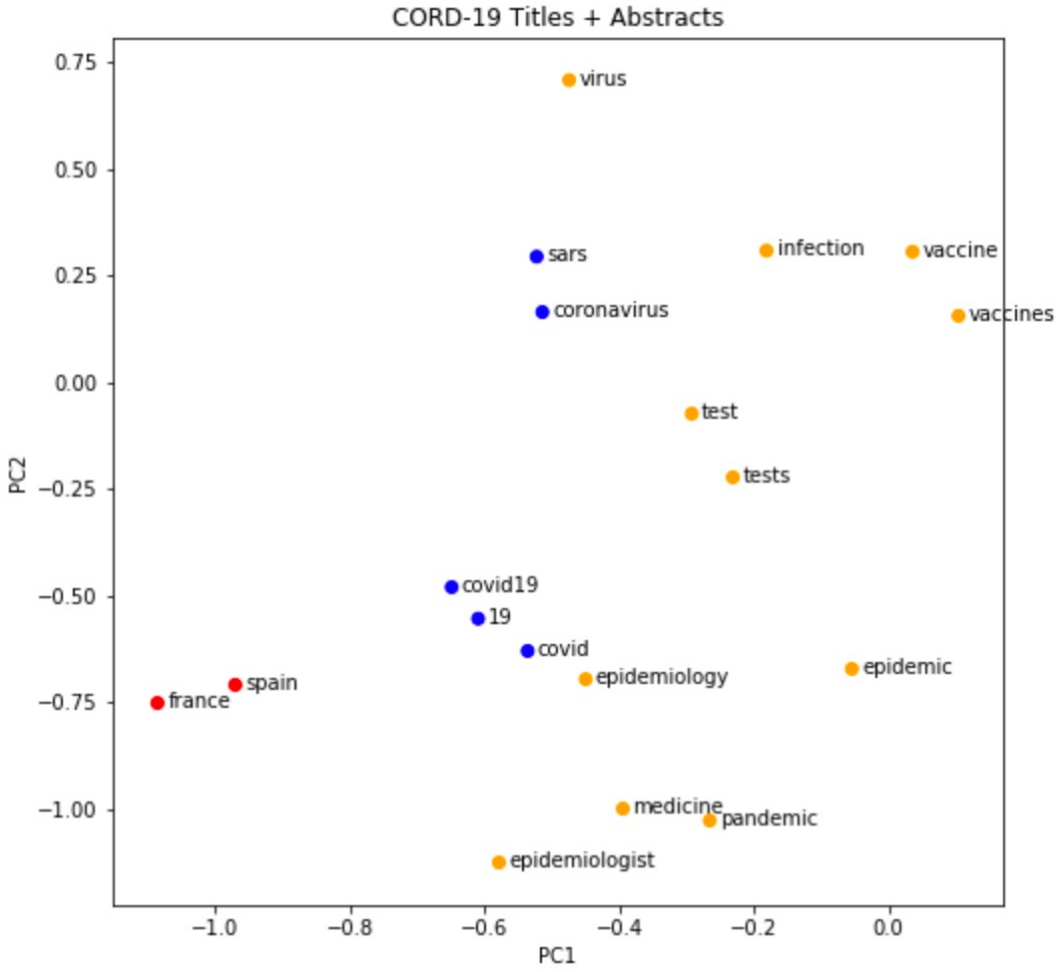
Word Vectors: Map each word to a dense vector

covid =

```
[0.60586,
 0.04596,
 0.12191,
-0.18414,
-0.04422,
 0.13495,
 0.31471,
 0.33992,
 0.01285,
-0.18592,
-0.43352,
-0.62741,
 0.24341,
 0.07149,
...]
```

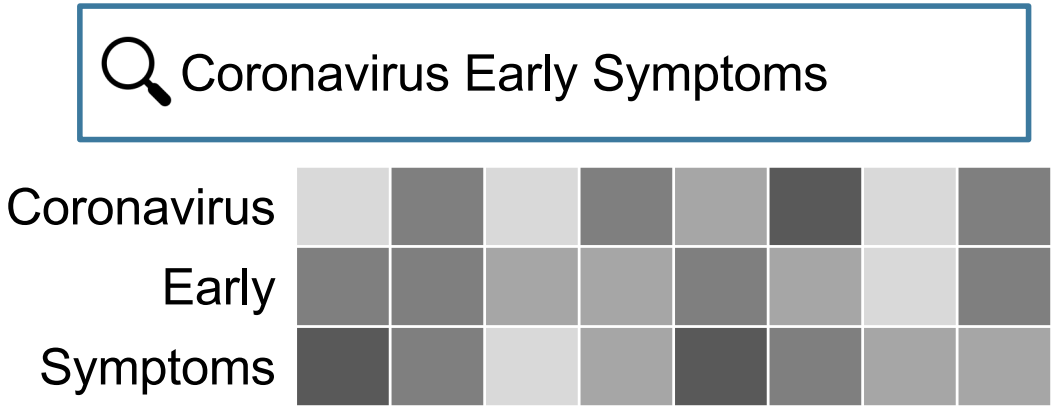
coronavirus =

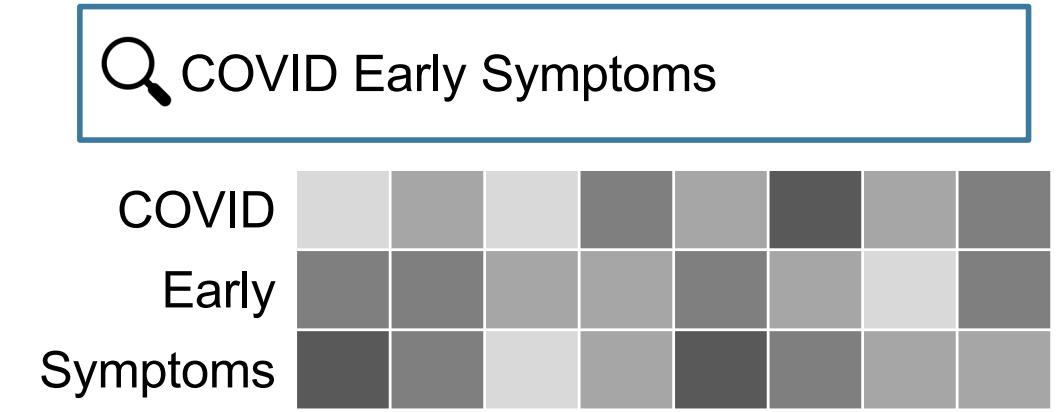
```
[0.33853,
-0.27798,
 0.08317,
-0.19729,
-0.49235,
 0.26514,
 0.03004,
 0.25704,
-0.38031,
-0.32722,
-0.47273,
-0.01596,
 0.32322,
-0.04947,
...]
```



Representing Text

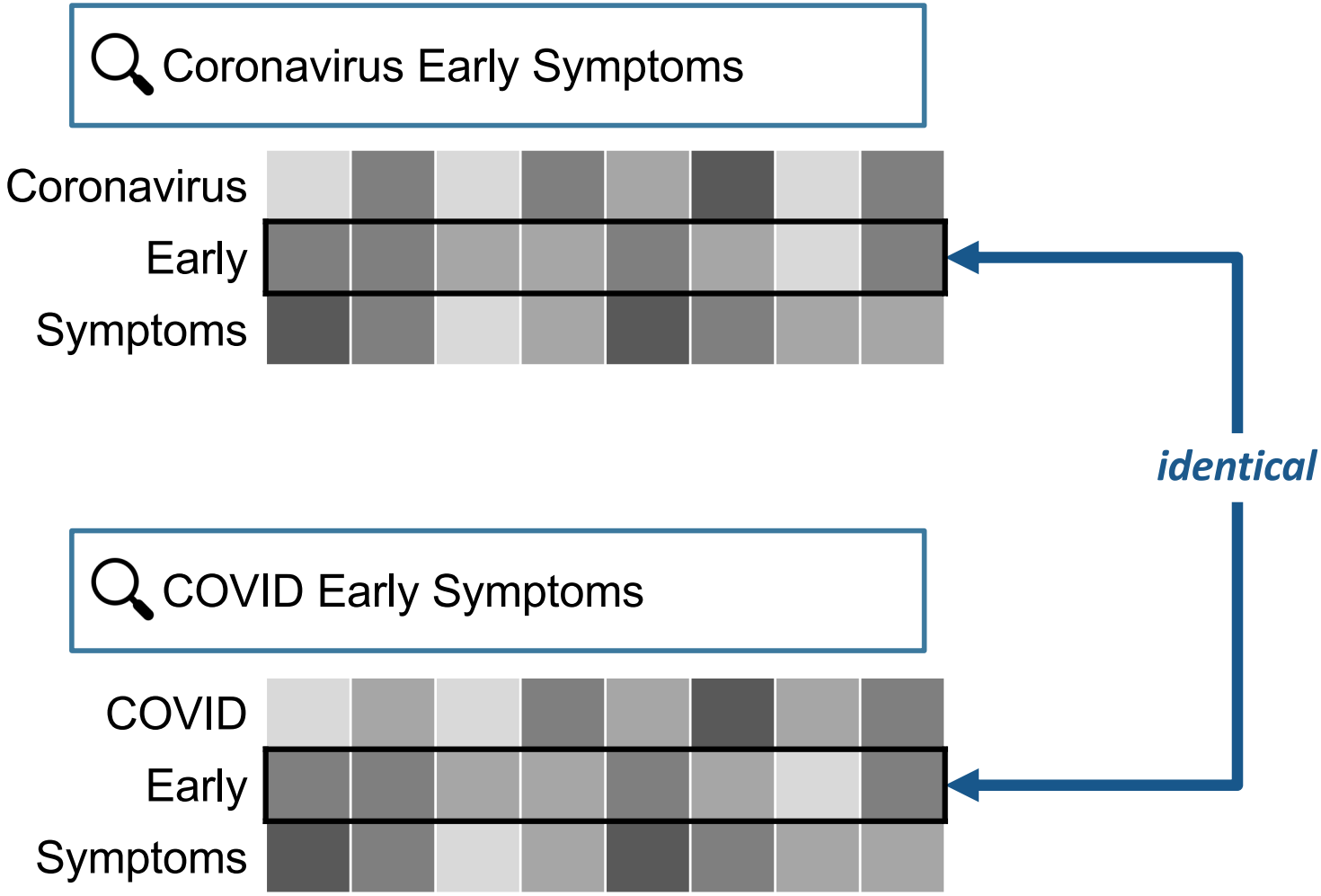
Word Vectors: Map each word to a dense vector





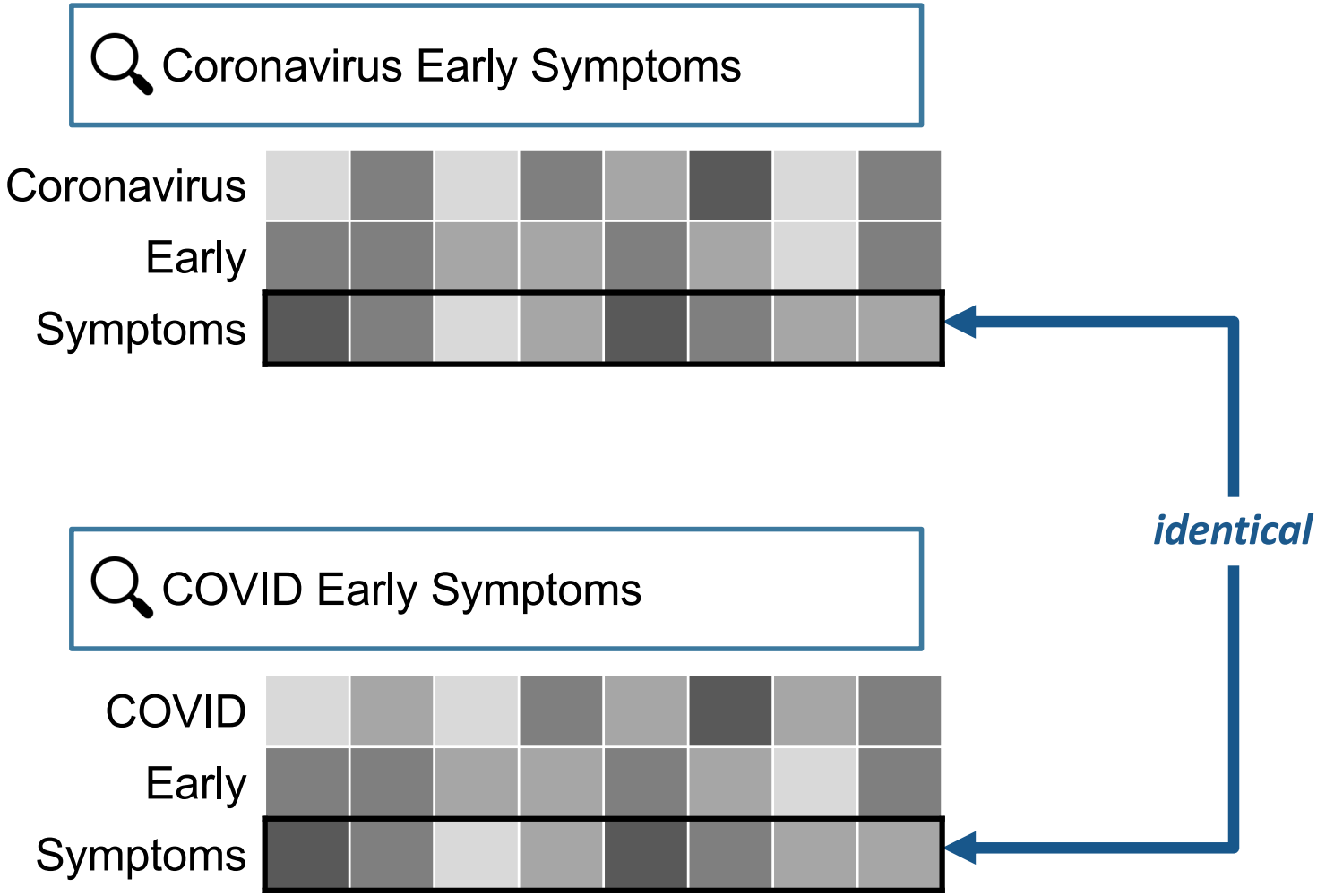
Representing Text

Word Vectors: Map each word to a dense vector



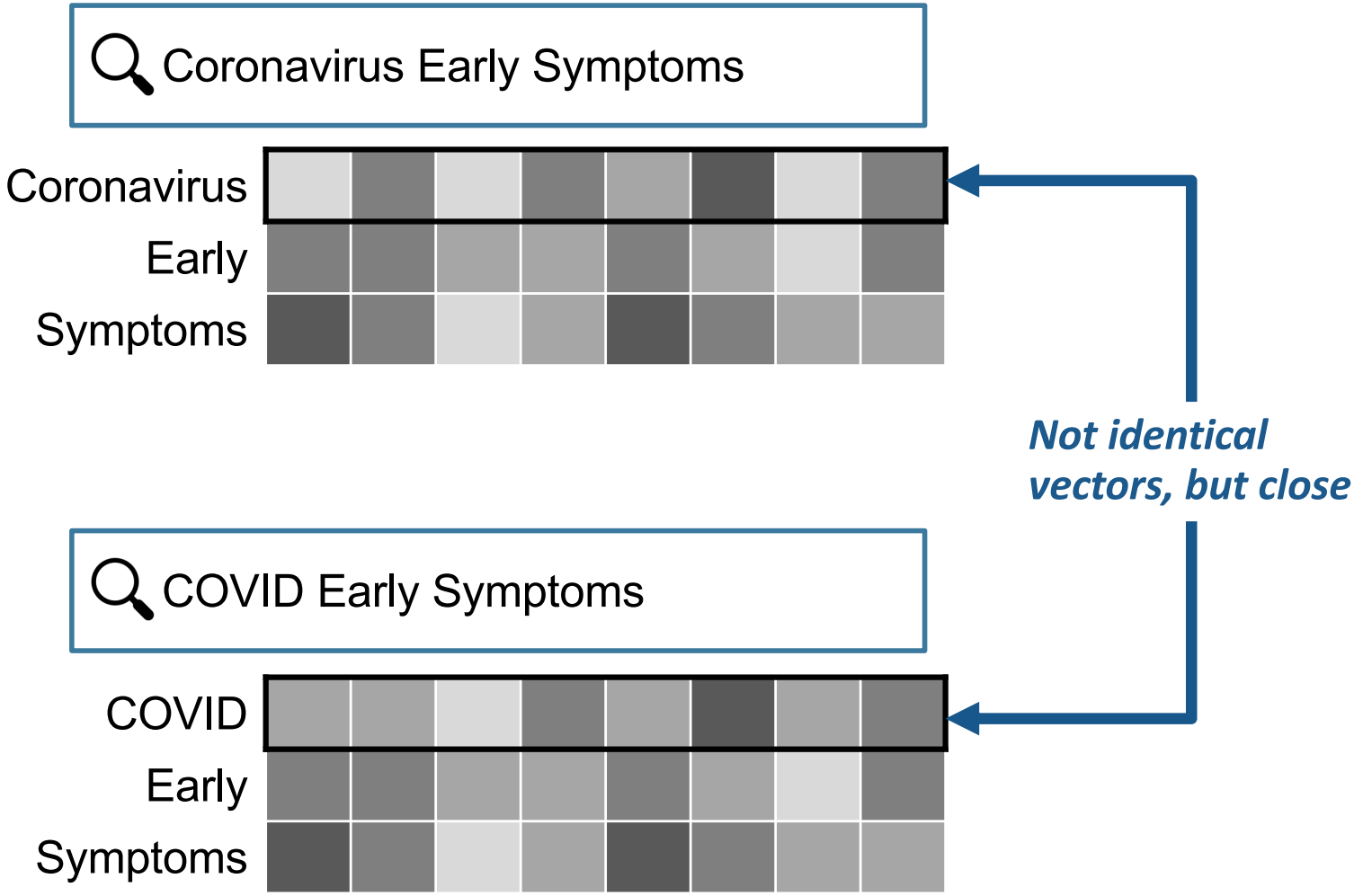
Representing Text

Word Vectors: Map each word to a dense vector



Representing Text

Word Vectors: Map each word to a dense vector



Representing Text

UNIVERSITÀ DI PISA | University of Glasgow

Building word vectors

- Based on word co-occurrences
- Trained using neural network
- e.g. word2vec, gloVe, etc.

Recent: “contextualized” word vectors – different based on the context that they appear in the text

- More on this later…

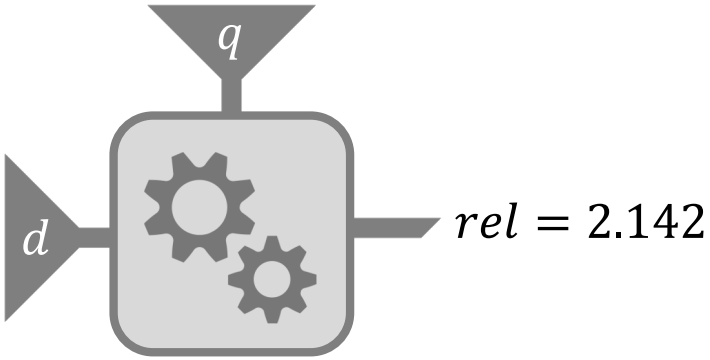
From Text to Features

Coronavirus Early Symptoms

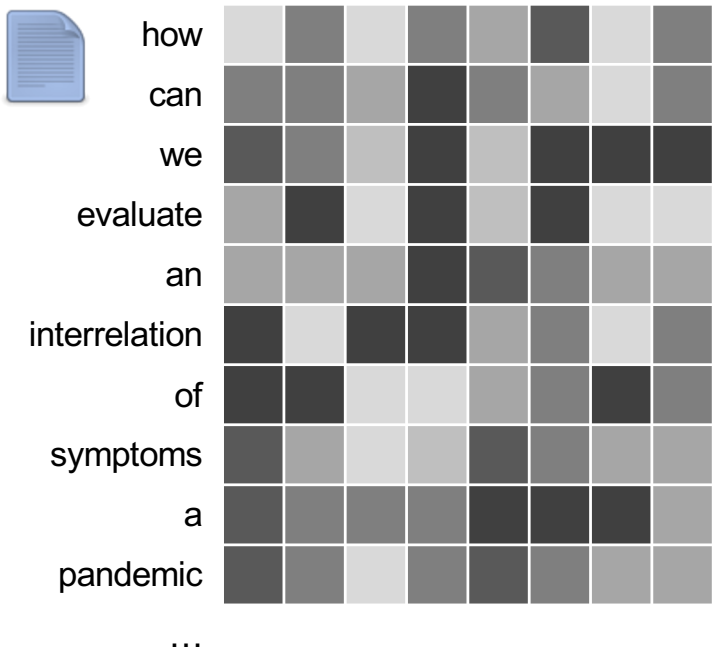
Document:

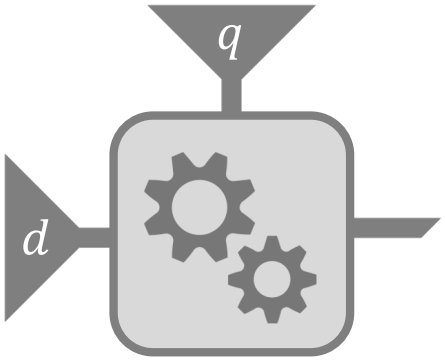
Title: How can we evaluate an interrelation of symptoms?

Abstract: A pandemic of 2019 novel coronavirus (COVID-19) is an international problem and factors associated with increased risk of mortality have been reported. However, there exists limited statistical method to estimate a comprehensive risk for a case in which a patient has several characteristics…



From Text to Features



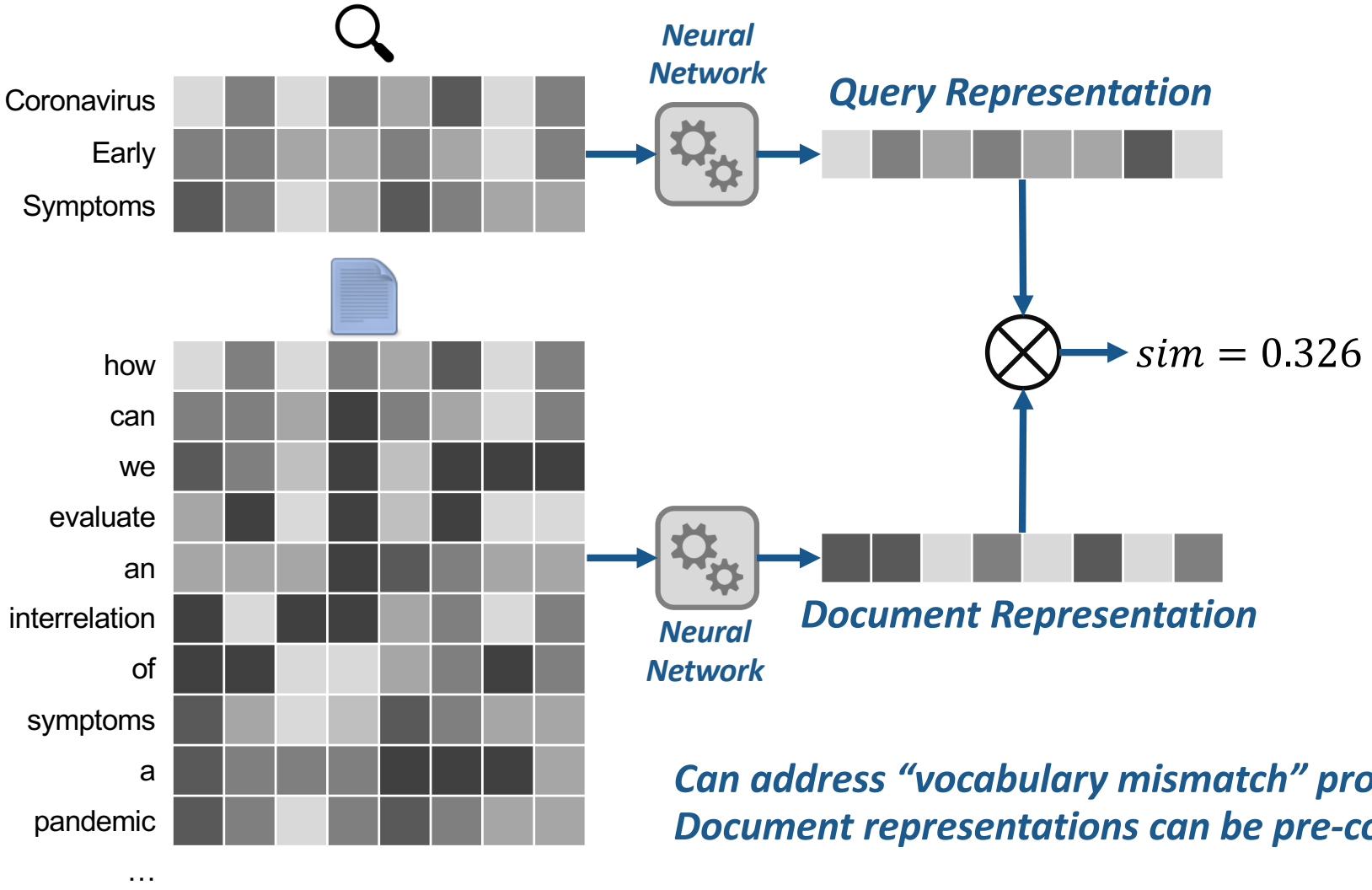


$rel = 2.142$

From Text Features to Scores

Also called “bi-encoder”

Two main approaches: **Representation** and Interaction

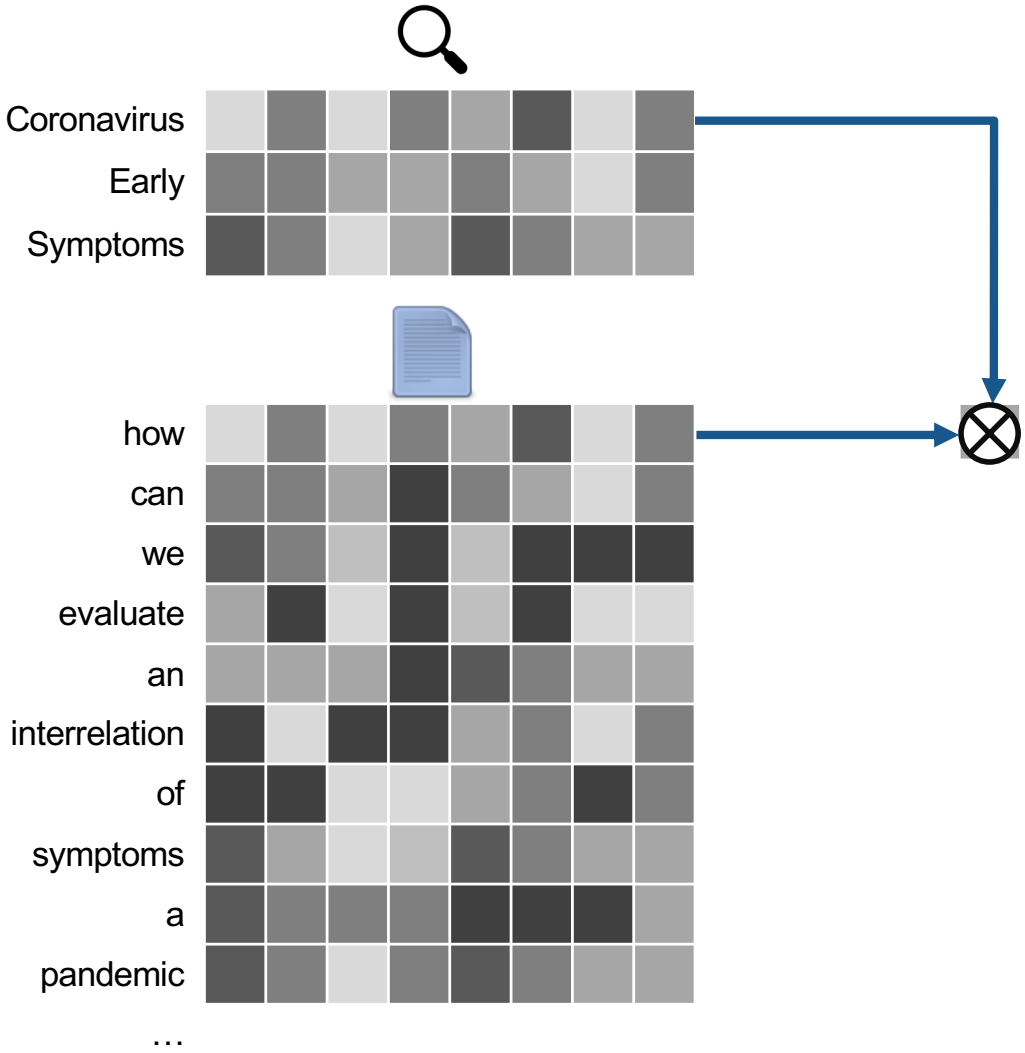


Can address “vocabulary mismatch” problem.
Document representations can be pre-computed.

From Text Features to Scores

Also called “cross-encoder”

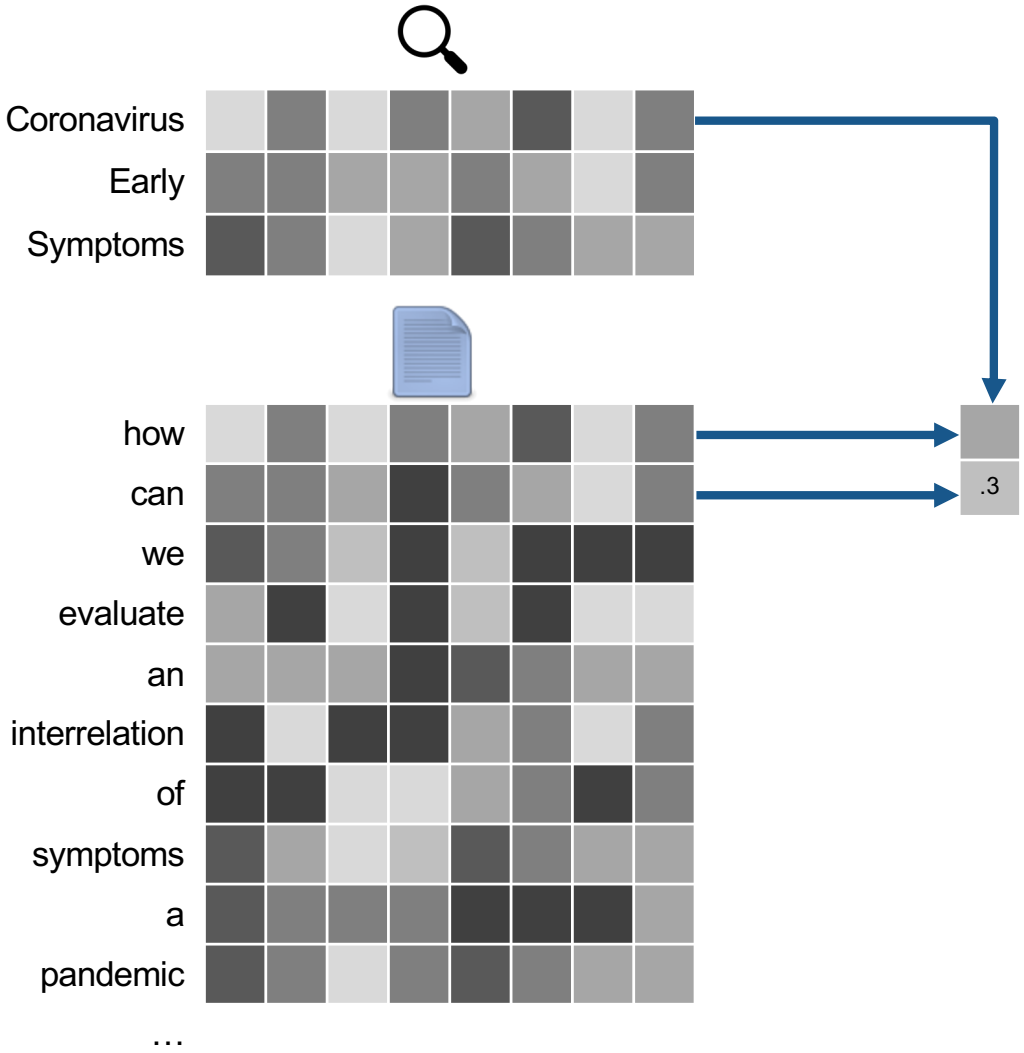
Two main approaches: Representation and **Interaction**



From Text Features to Scores

Also called “cross-encoder”

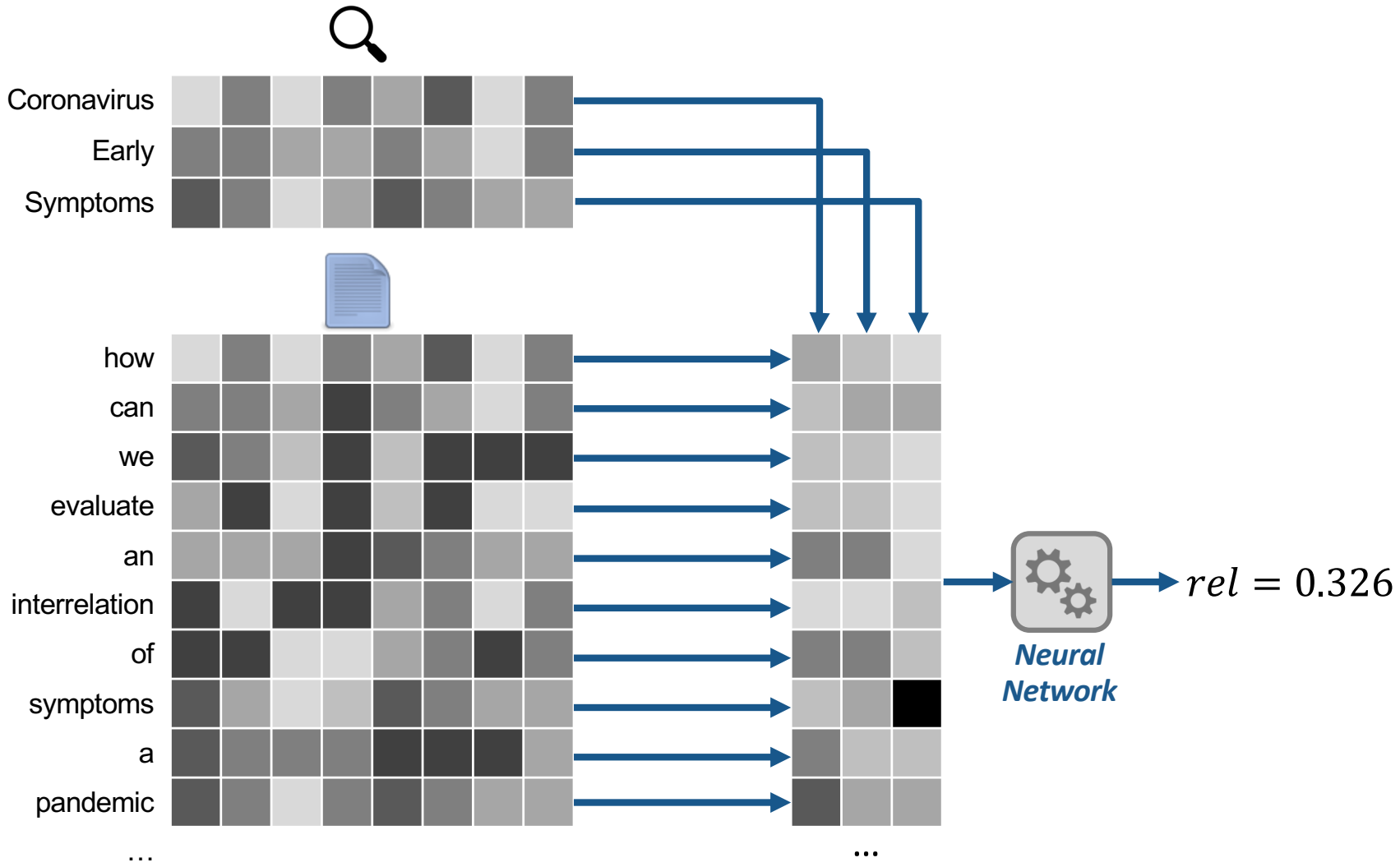
Two main approaches: Representation and **Interaction**



From Text Features to Scores

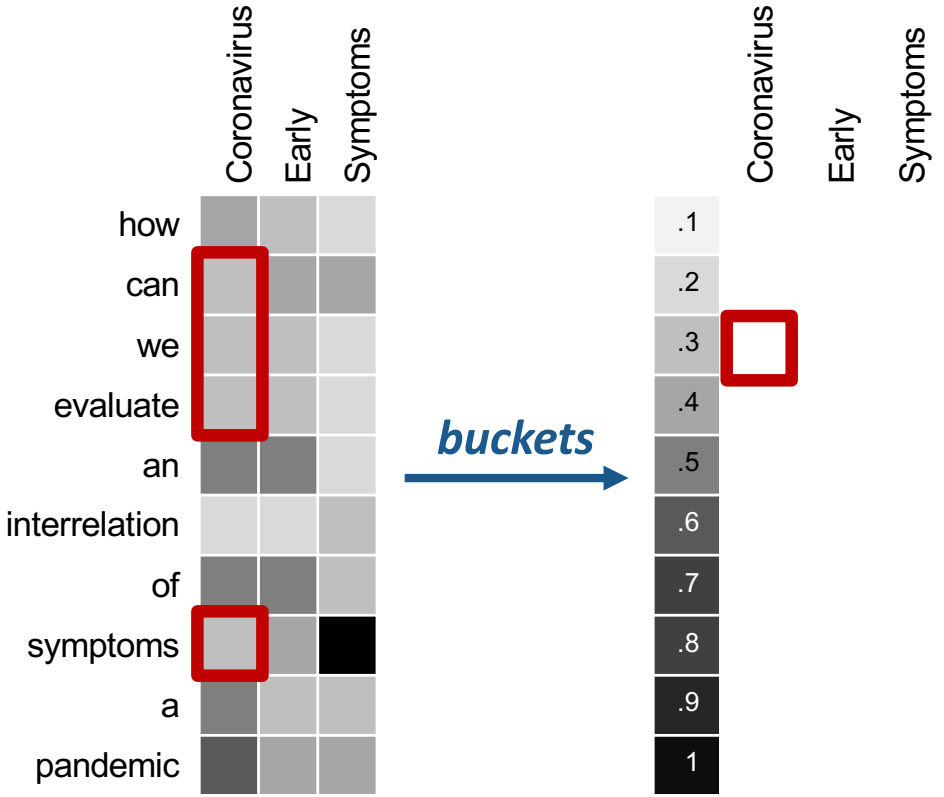
Also called “cross-encoder”

Two main approaches: Representation and **Interaction**



Interaction Aggregation

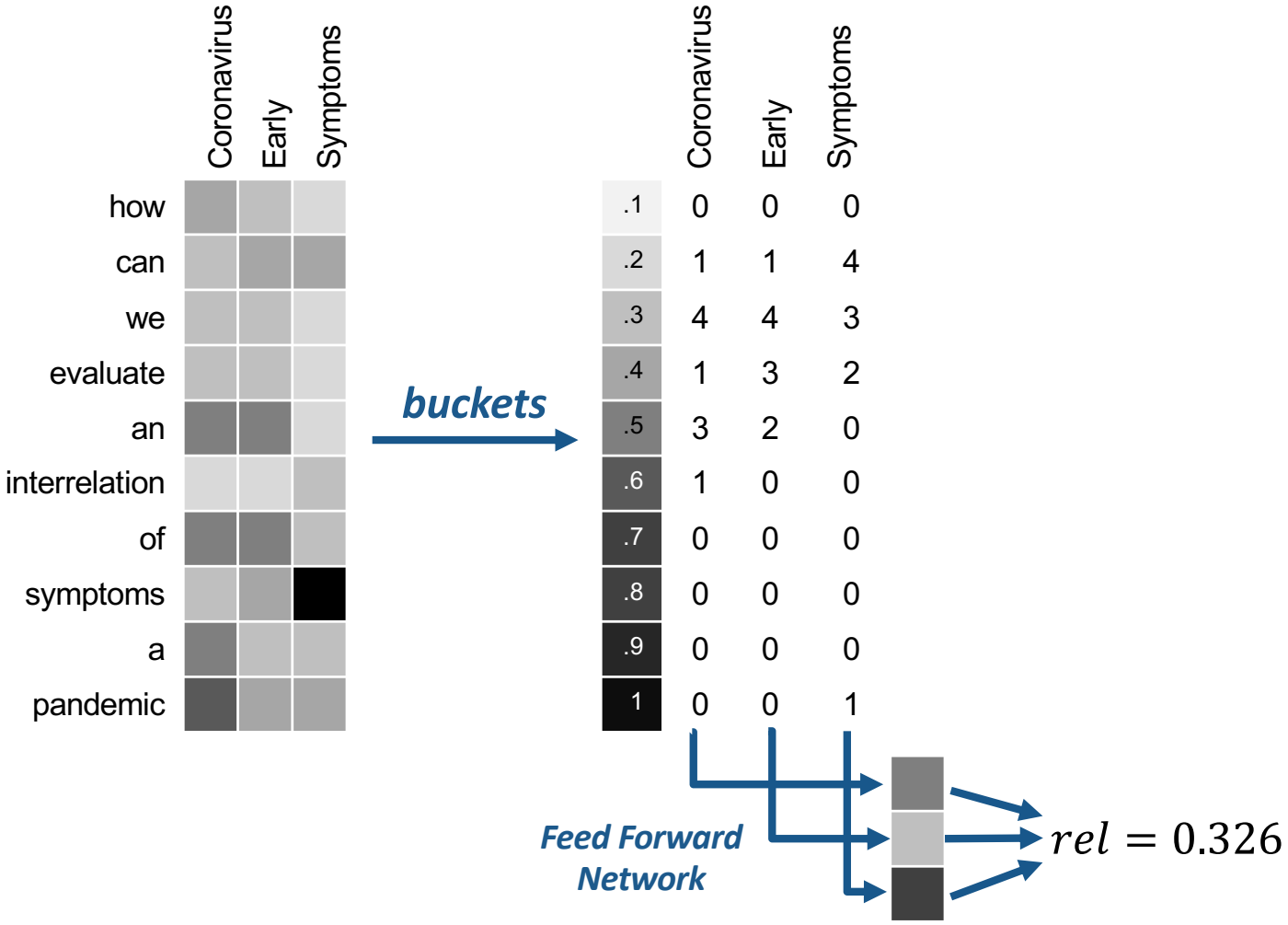
Bucketing - DRMM (Deep Relevance Matching Model)



Guo et al. A Deep Relevance Matching Model for Ad-hoc Retrieval. CIKM 2017.

Interaction Aggregation

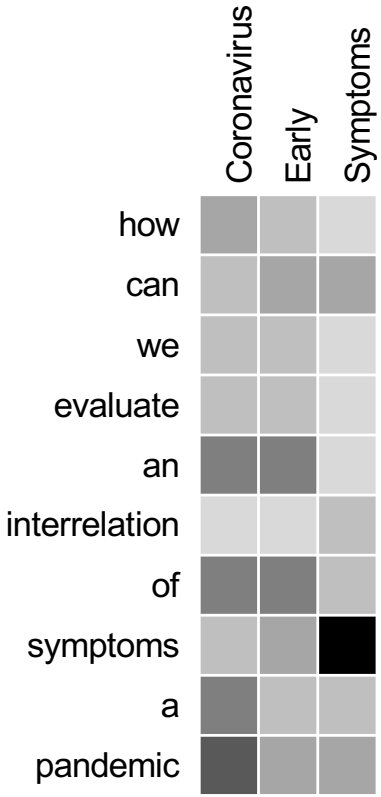
Bucketing - DRMM (Deep Relevance Matching Model)

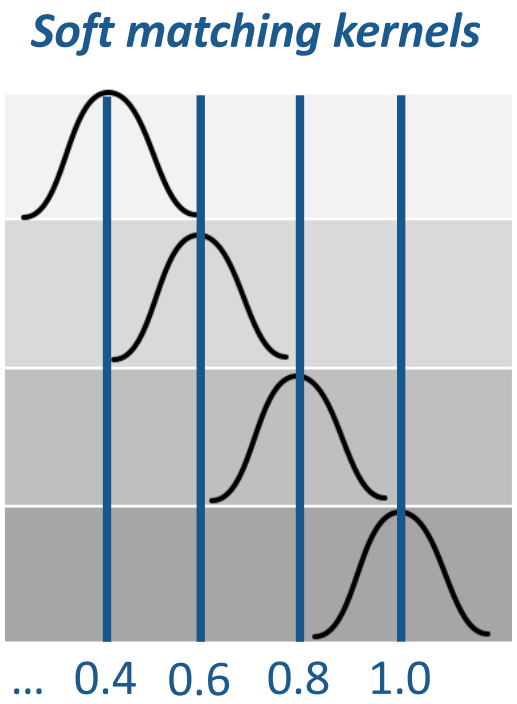


Guo et al. A Deep Relevance Matching Model for Ad-hoc Retrieval. CIKM 2017.

Interaction Aggregation

Soft Bucketing - KNRM (Kernel-based Neural Ranking Model)

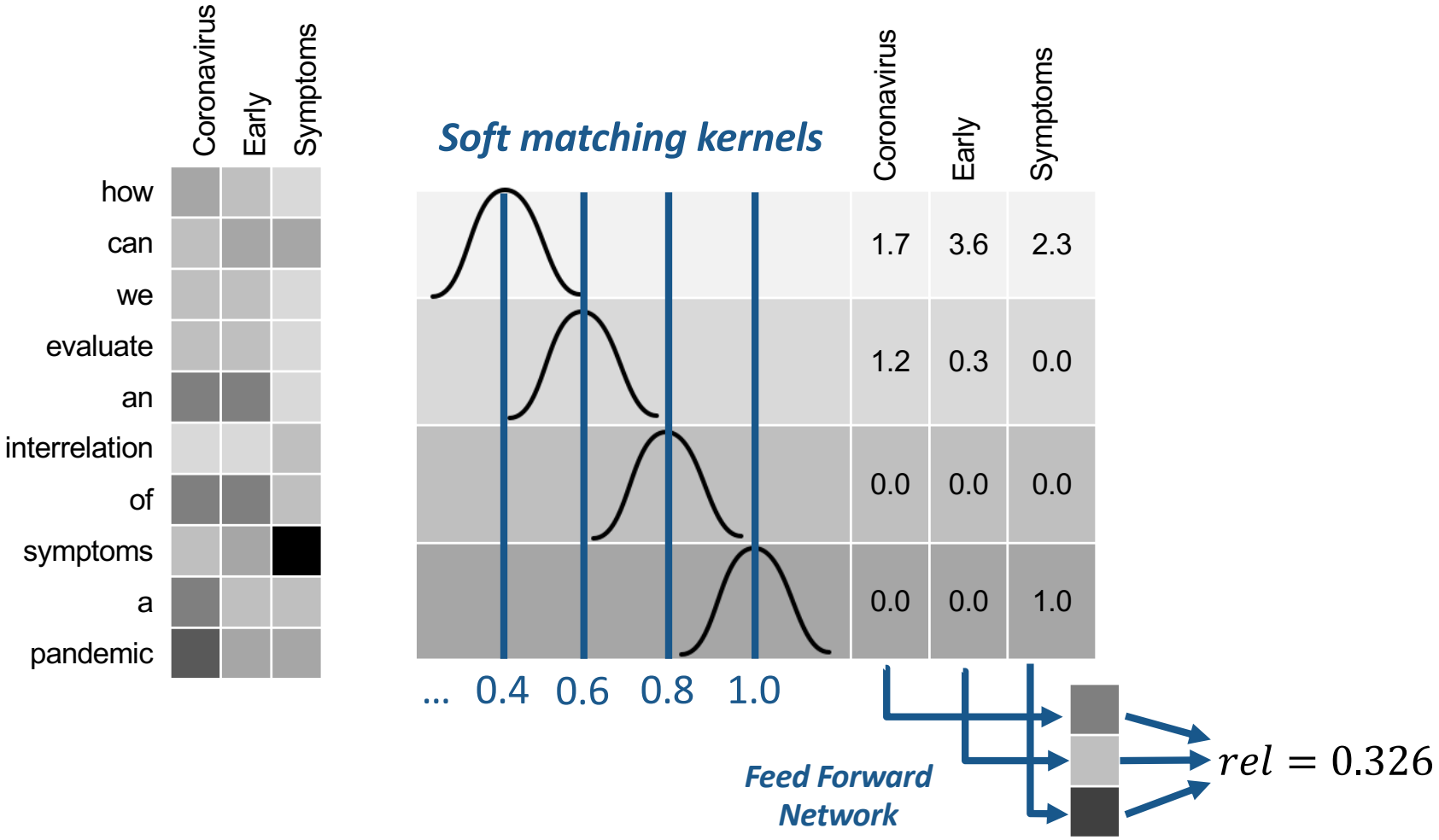




Xiong et al. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. SIGIR 2017.

Interaction Aggregation

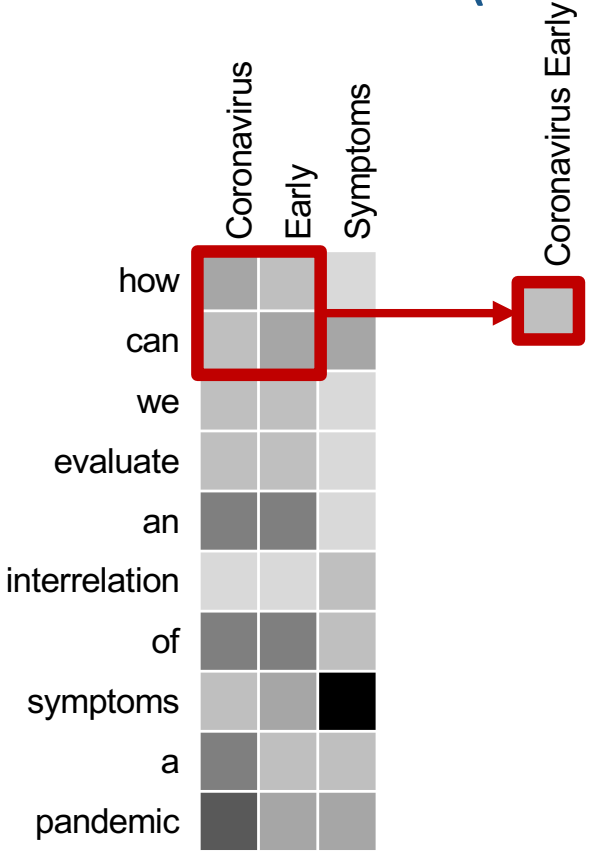
Soft Bucketing - KNRM (Kernel-based Neural Ranking Model)



Xiong et al. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. SIGIR 2017.

Interaction Aggregation

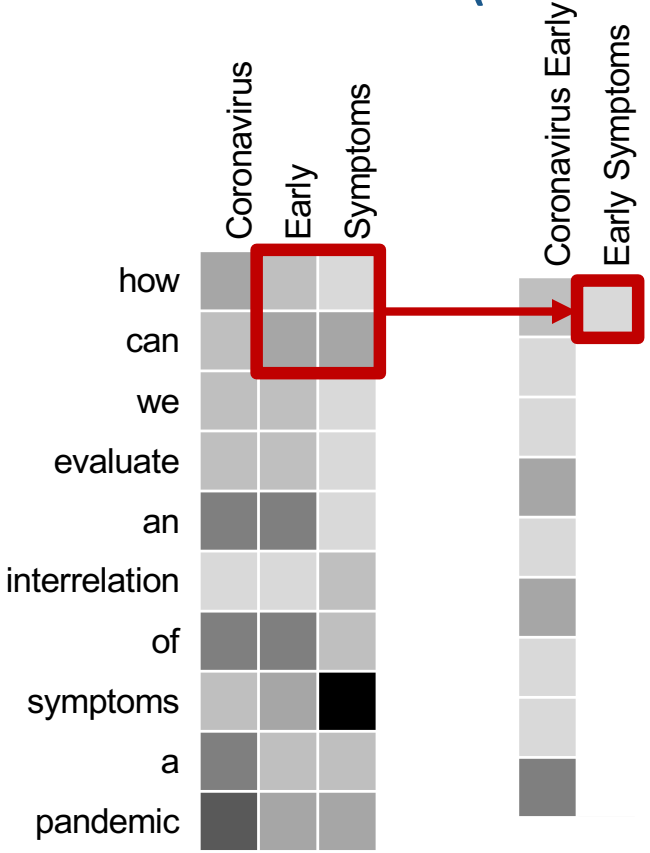
Convolution - PACRR (Position-Aware Convolutional-Recurrent Relevance)



Hui et al. PACRR: A Position-Aware Neural IR Model for Relevance Matching. EMNLP 2017.

Interaction Aggregation

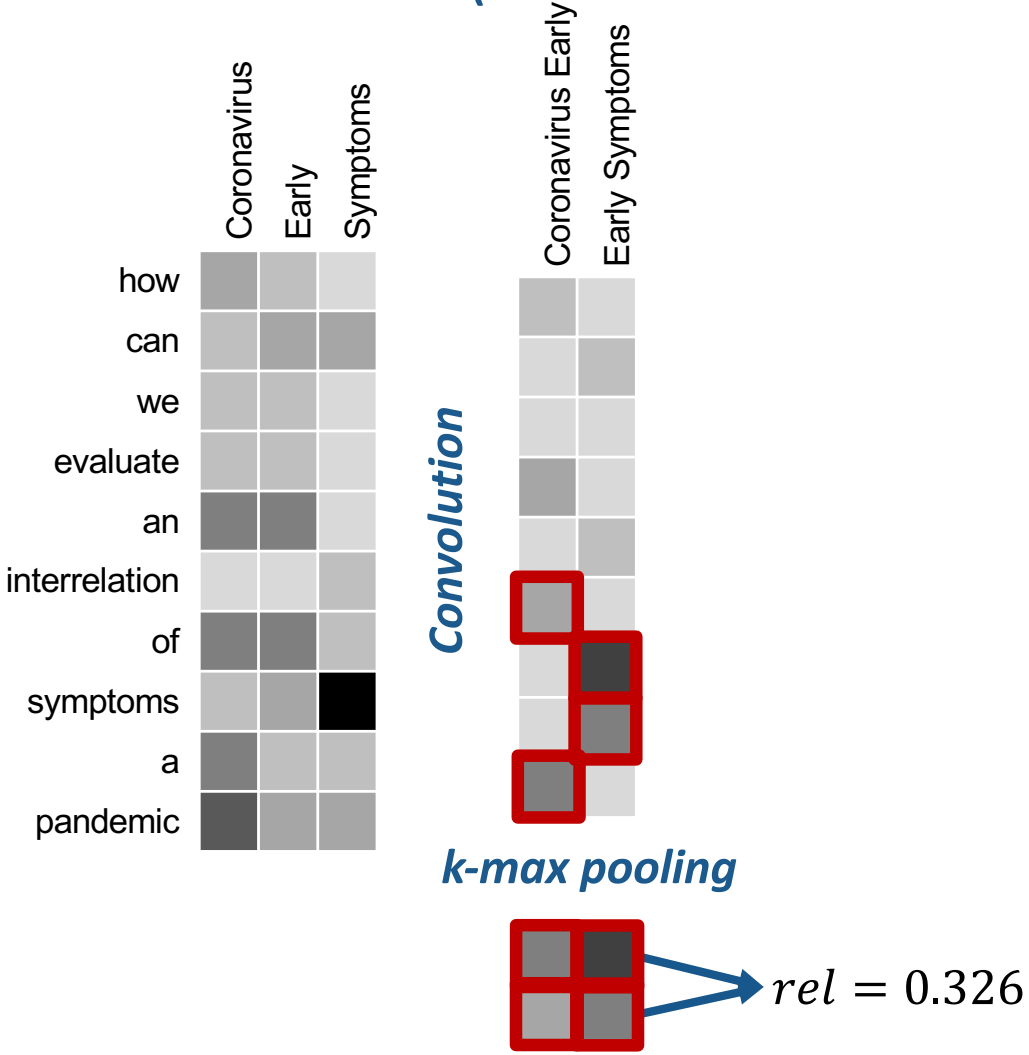
Convolution - PACRR (Position-Aware Convolutional-Recurrent Relevance)



Hui et al. PACRR: A Position-Aware Neural IR Model for Relevance Matching. EMNLP 2017.

Interaction Aggregation

Convolution - PACRR (Position-Aware Convolutional-Recurrent Relevance)



Hui et al. PACRR: A Position-Aware Neural IR Model for Relevance Matching. EMNLP 2017.

Interaction Aggregation

Interaction Models

- Soft term matching: DRMM, KNRM, PACRR
- Bucketing: DRMM, KNRM
- Proximity Matching: PACRR

nDCG@20

Model	WT 2012	WT 2013	WT 2014
DRMM	0.197	0.228	0.300
KNRM	0.222	0.251	0.324
PACRR	**0.243**	**0.295**	**0.339**

Results reported by Hui, et al. 2017. (PACRR)

OpenNIR

Università di Pisa | University of Glasgow

A toolkit for training and evaluating neural IR models

Historically: Script-based

Today: Can use with PyTerrier!

https://OpenNIR.net/

IR LAB GEORGETOWN UNIVERSITY

MacAvaney. OpenNIR: A Complete Neural Ad-Hoc Ranking Pipeline. WSDM 2020.

Interaction Aggregation - Practical

Università di Pisa | University of Glasgow

Ranker: Which model to use?

Which doc text field to use?

```
drmm = onir_pt.reranker('drmm', 'wordvec_hash', text_field='abstract')
knrm = onir_pt.reranker('knrm', 'wordvec_hash', text_field='abstract')
pacrr = onir_pt.reranker('pacrr', 'wordvec_hash', text_field='abstract')
```

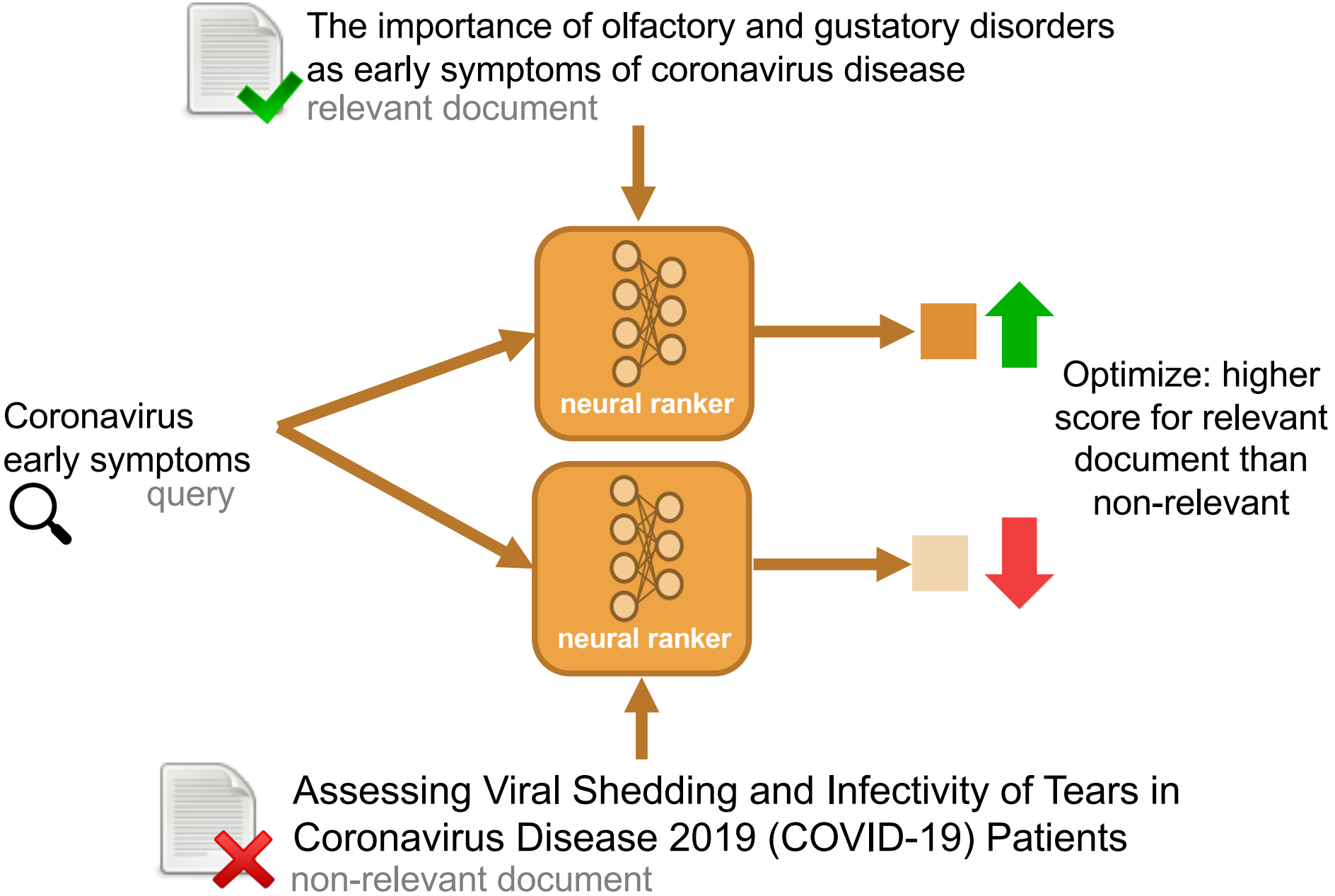
Vocab: Which word embeddings to use?

```
br = pt.BatchRetrieve(index) % 100
pipeline = br >> pt.text.get_text(dataset, 'abstract') >> reranker
```


```
pt.Experiment(
    [br, drmm_pipeline, knrm_pipeline, pacrr_pipeline],
    dataset.get_topics('title'),
    dataset.get_qrels(),
    names=['DPH', 'DPH >> DRMM', 'DPH >> KNRM', 'DPH >> PACRR'],
    eval_metrics=[RR, P@5, nDCG@10, 'mrt']
)
```

	name	recip_rank	P_5	ndcg_cut_10	mrt
0	DPH	0.767259	0.684	0.584309	33.919908
1	DPH >> DRMM	0.515536	0.420	0.377125	88.168377
2	DPH >> KNRM	0.488852	0.340	0.329785	79.464181
3	DPH >> PACRR	0.623139	0.532	0.457561	89.766008

Pairwise Training



Pairwise Training

Training data sources:

- Annotated collections
- Behavioral data
- Weak supervision / relevance transfer

Graphic from trec.nist.gov

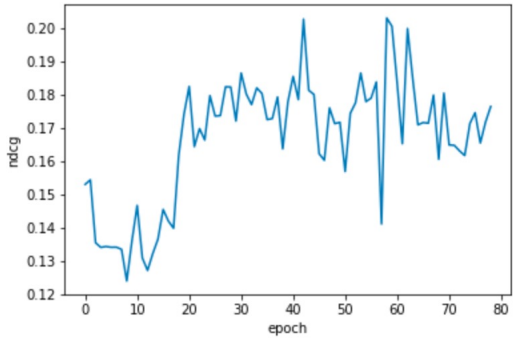
Pairwise Training

Ranker: Train model using medical subset of MS MARCO [1,2]

```
train_ds = pt.datasets.get_dataset('irds:msmarco-passage/train/medical')
```

```
fit_res = knrm.fit(
    train_ds.get_topics(),
    train_ds.get_qrels(),
    valid_ds.get_topics(),
    valid_ds.get_qrels())
```

```
plt.plot(fit_res['ndcg'])
```



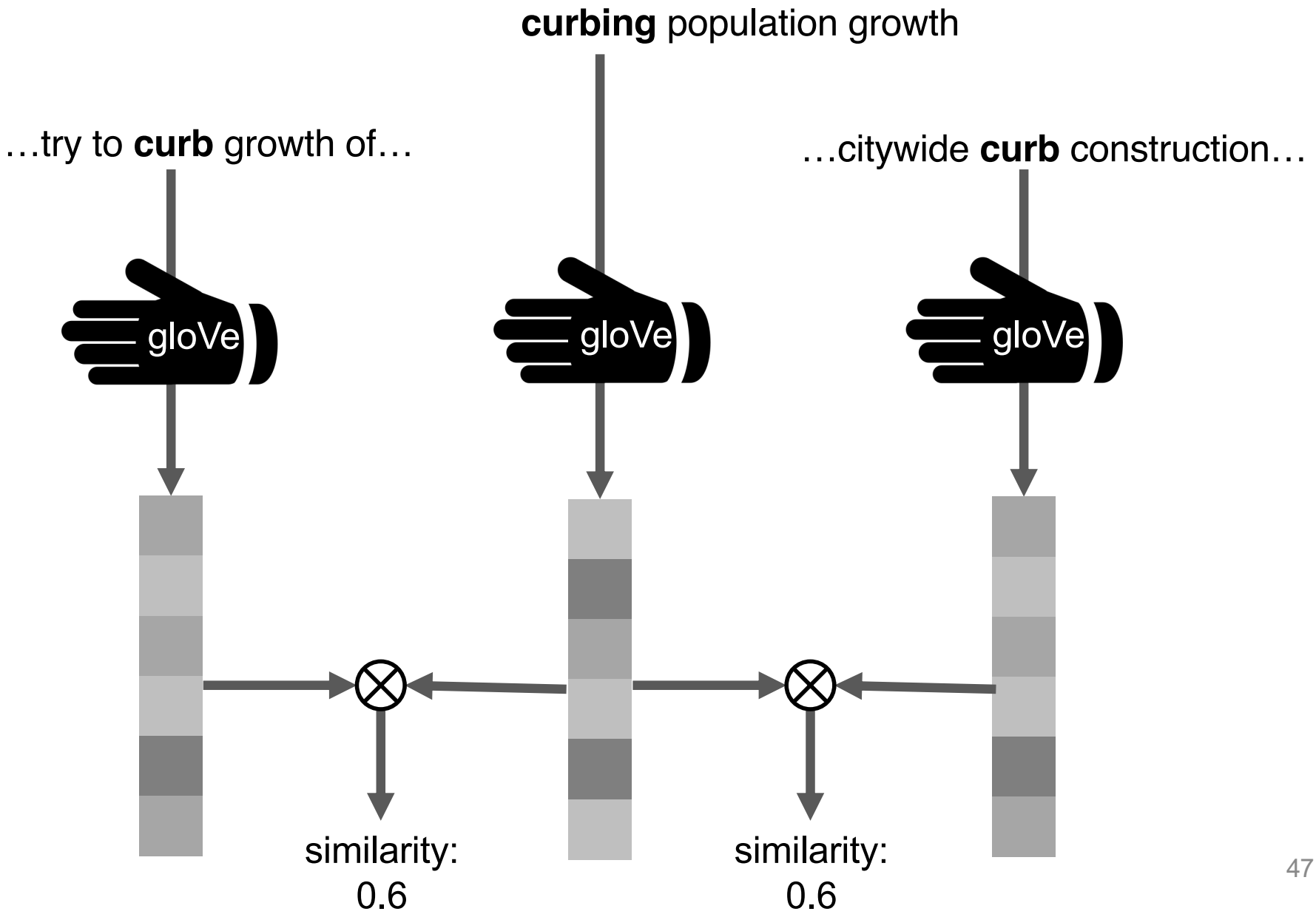
	name	map	recip_rank	ndcg	ndcg_cut_10	mrt
0	DPH	0.075329	0.767259	0.164584	0.584309	30.752108
1	DPH >> KNRM	0.075105	0.810732	0.164964	0.619237	77.134939

[1] Bajaj et al. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset}. InCoCo@NeuIPS 2016.
[2] MacAvaney et al. SLEDGE-Z: A Zero-Shot Baseline for COVID-19 Literature Search. EMNLP 2020.

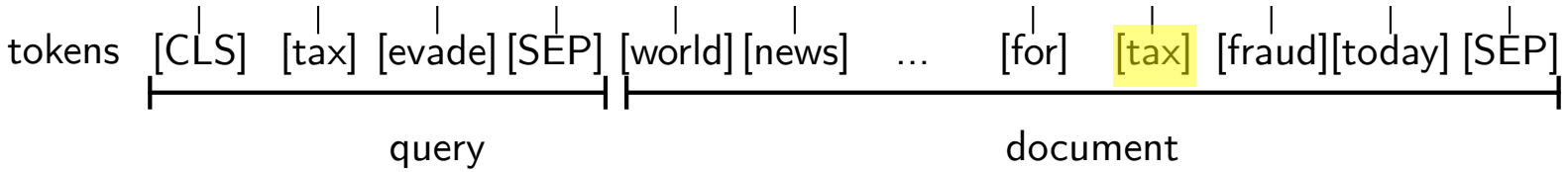
Part 3B

CONTEXTUALIZED LANGUAGE MODELS FOR RE-RANKING

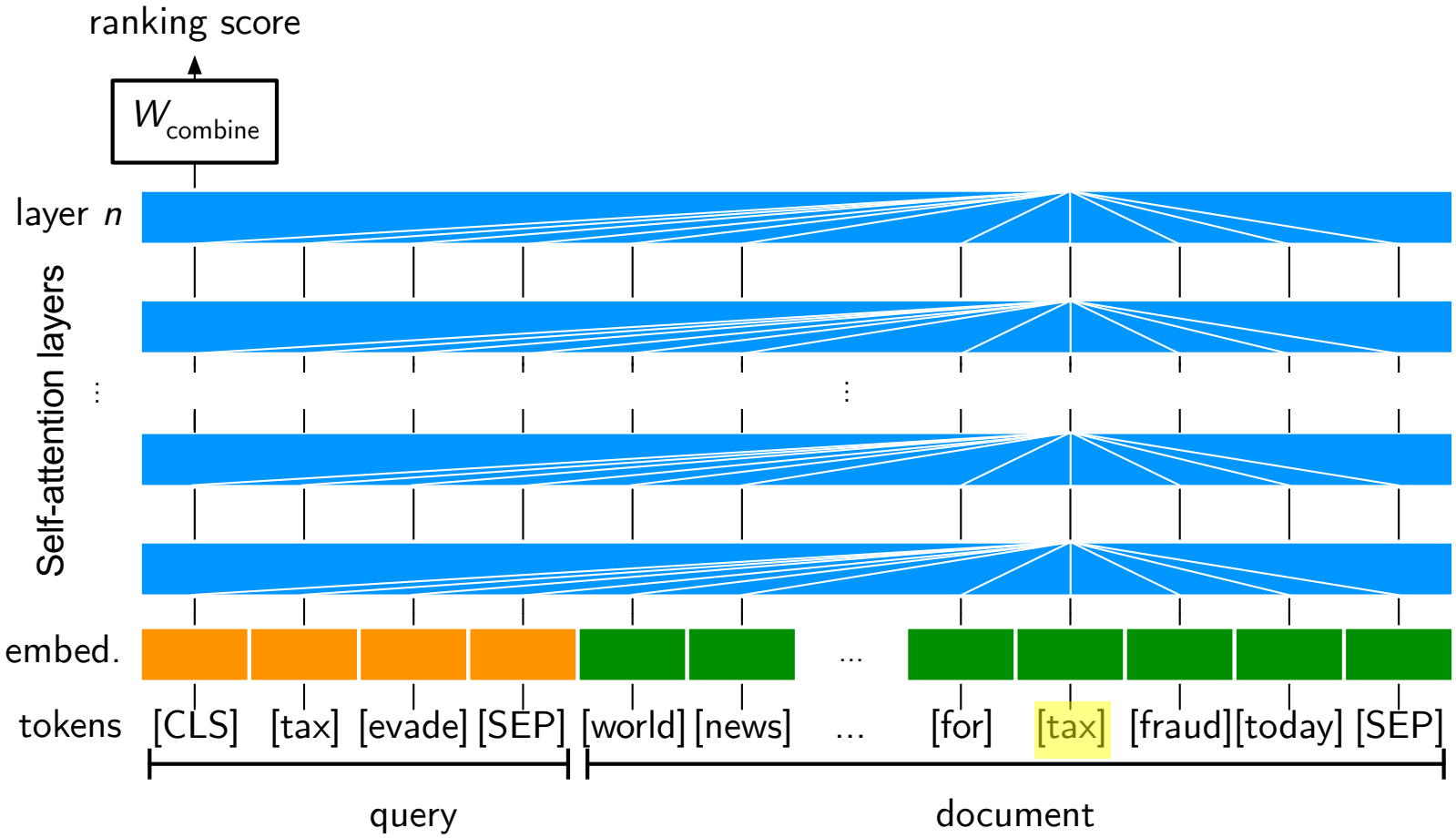
Static Word Vectors



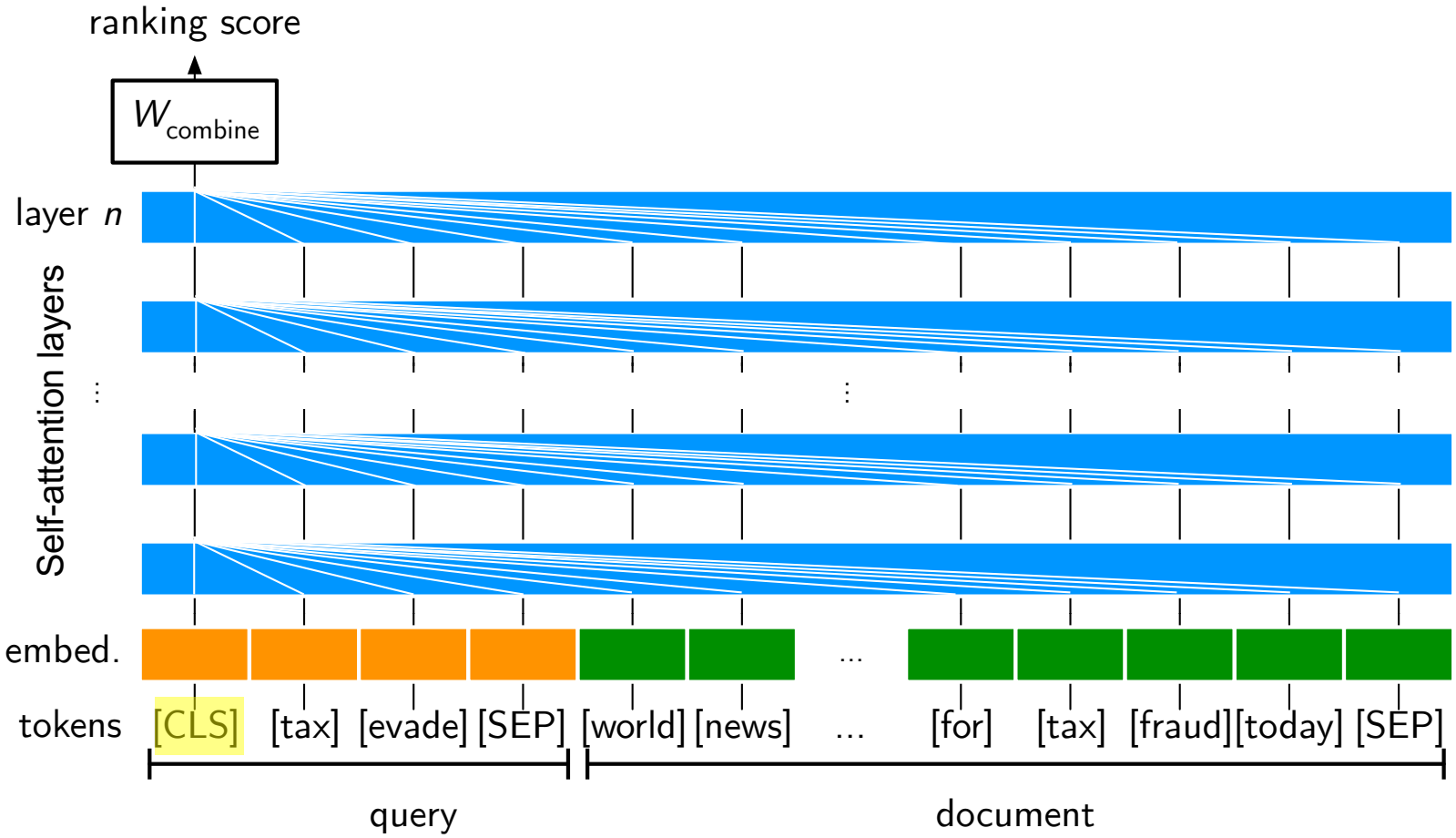
BERT & Self-Attention Networks



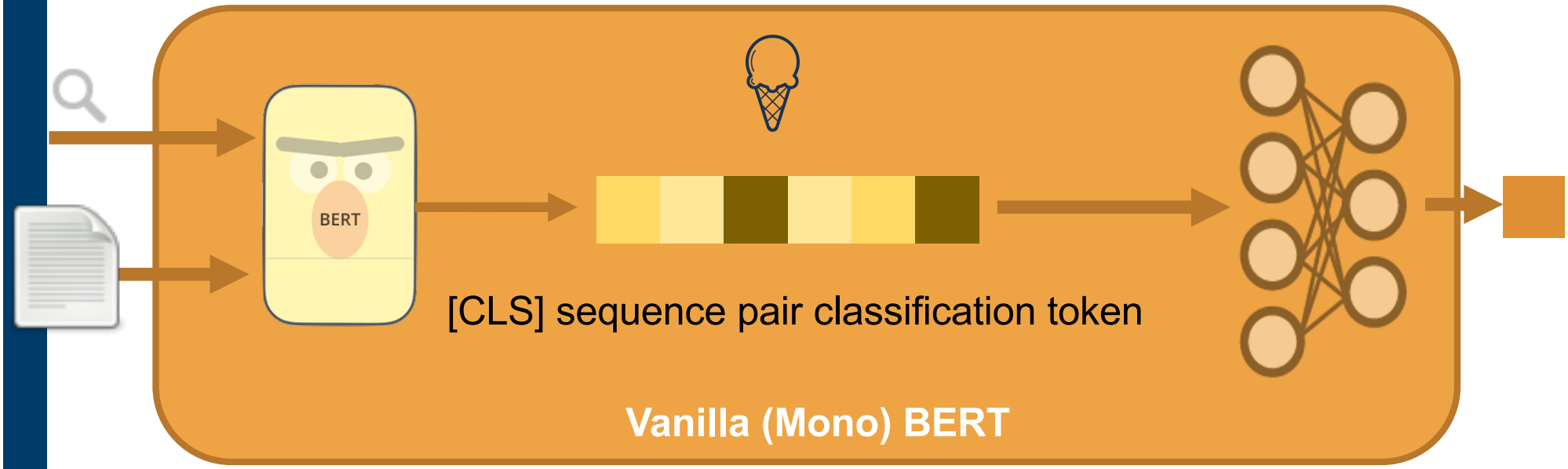
BERT & Self-Attention Networks



BERT & Self-Attention Networks

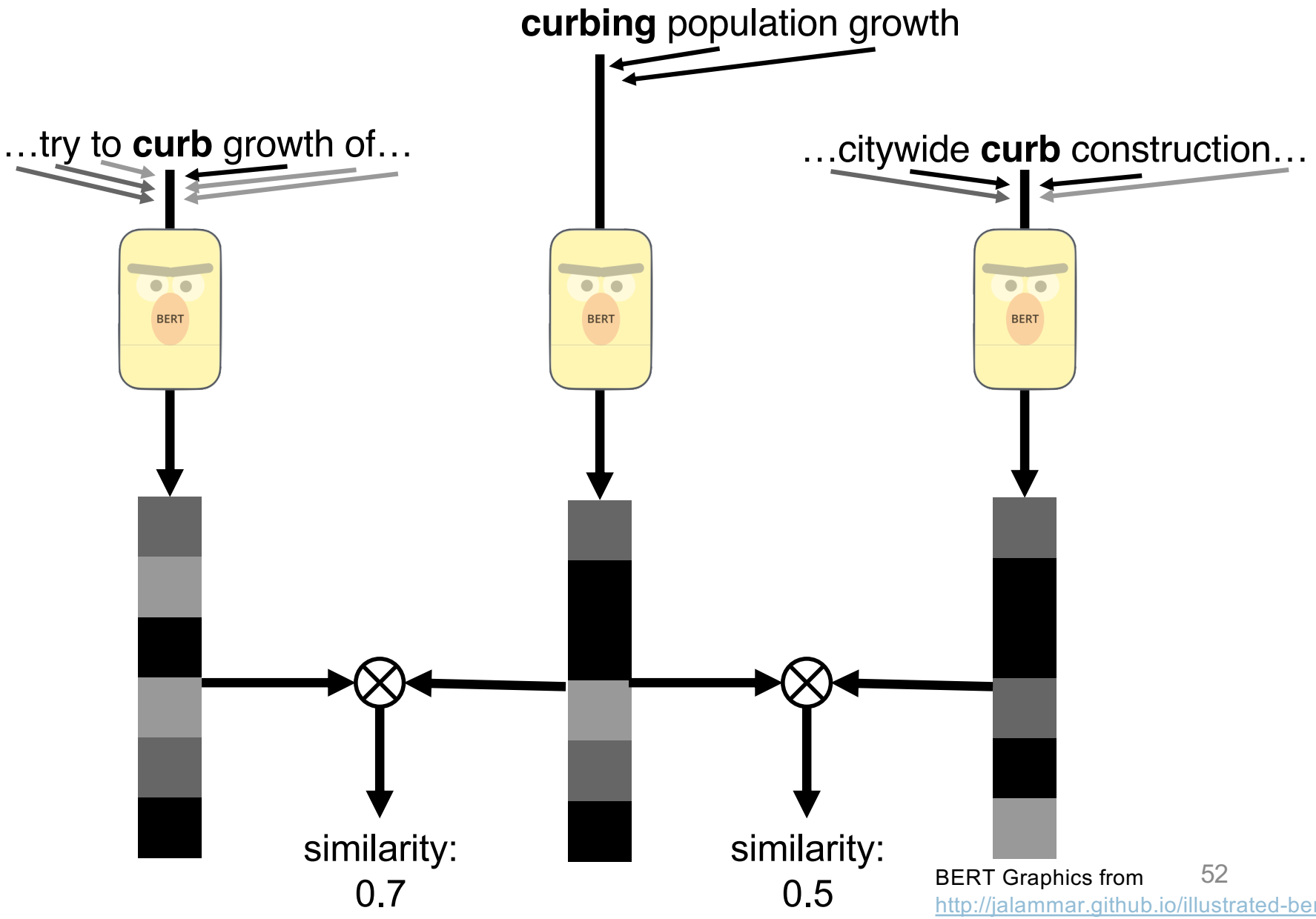


Vanilla BERT



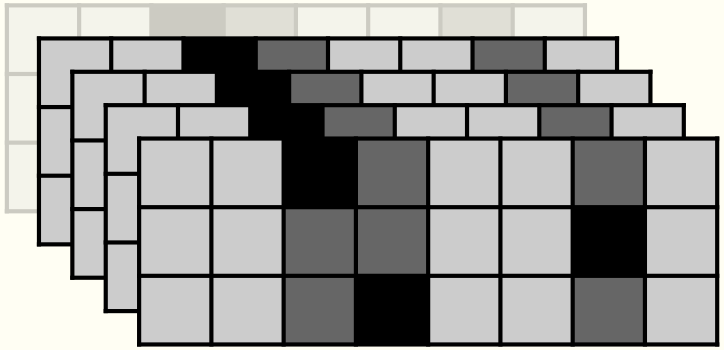
BERT Graphics from Jay Alammar, http://jalammar.github.io/illustrated-bert/

Contextualized Word Vectors



BERT Graphics from http://jalammar.github.io/illustrated-bert/

BERT consists of multiple layers. We build a similarity matrix for each layer.

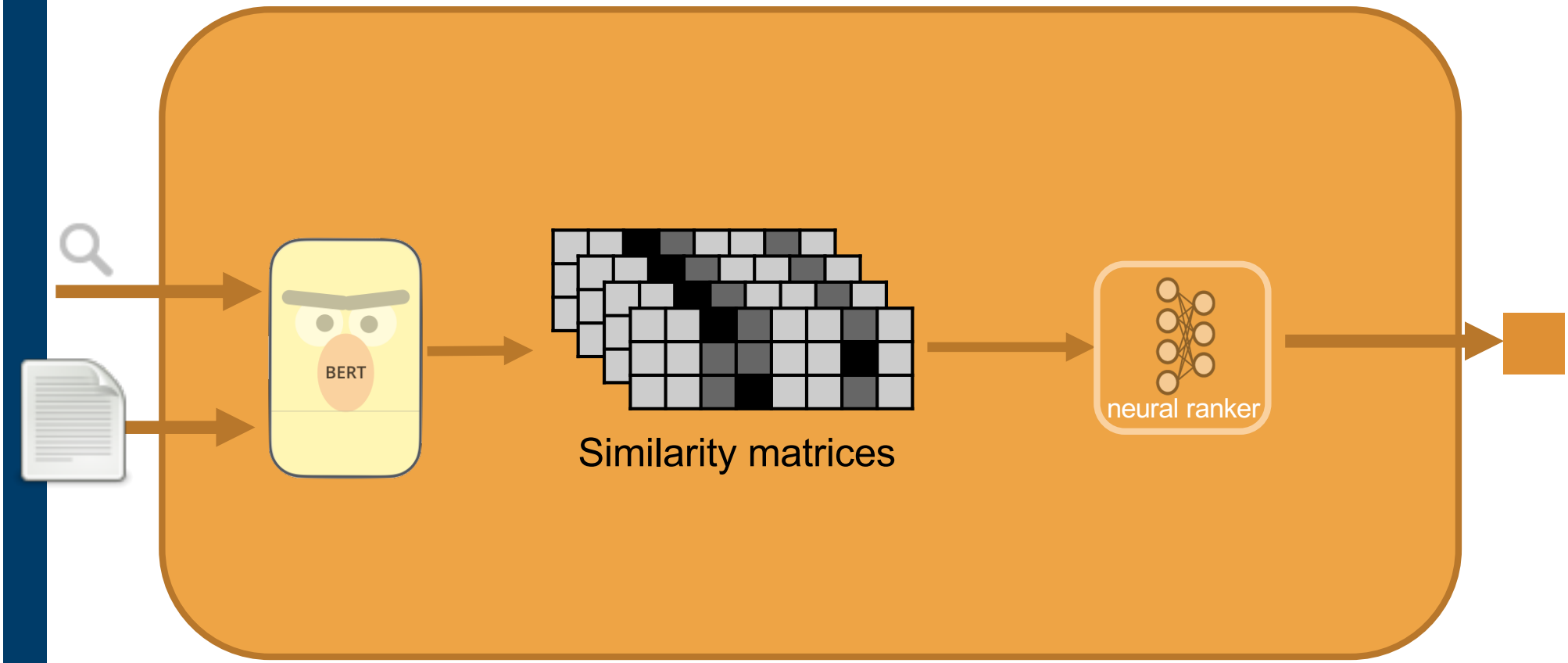


$BERT_{BASE}$: 13 Layers

* Including 1 input layer (non-transformed)

BERT Graphics from Jay Alammar, http://jalammar.github.io/illustrated-bert/

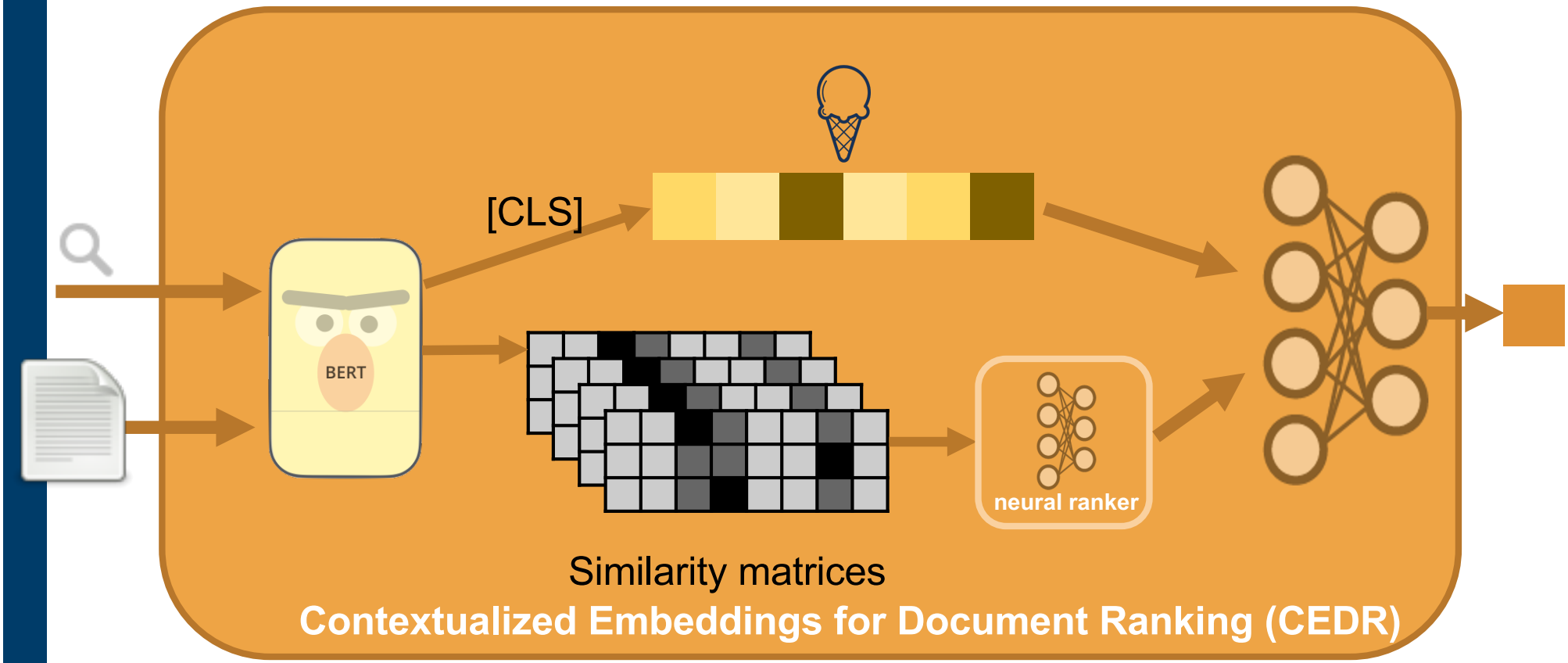
Contextualized Word Vectors



MacAvaney et al. CEDR. SIGIR 2019.

BERT Graphics from Jay Alammar, http://jalammar.github.io/illustrated-bert/

CEDR



MacAvaney et al. CEDR. SIGIR 2019.

BERT Graphics from Jay Alammar, http://jalammar.github.io/illustrated-bert/

BERT Practical

Replace KNRM's vocabulary with BERT

```
bert_knrm = onir_pt.reranker('knrm', 'bert', text_field='title_abstract')
```

Vanilla model that uses BERT's [CLS] for ranking

```
vbert = onir_pt.reranker('vanilla_transformer', 'bert', text_field='title_abstract')
cedr_knrm = onir_pt.reranker('cedr_knrm', 'bert', text_field='title_abstract')
```

CEDR KNRM model (using both KNRM and [CLS])

	name	map	ndcg	ndcg_cut_10	mrt
0	BM25	0.077880	0.177728	0.644374	36.795420
1	BM25 >> BERT	0.065846	0.160599	0.416803	1847.491717
2	BM25 >> BERT KNRM	0.059786	0.151265	0.277798	1906.480065
3	BM25 >> CEDR KNRM	0.067430	0.165405	0.507656	1917.520533

You still need to train these models!

BERT Practical

Università di Pisa | University of Glasgow

Load trained model from URL

```
bert = onir_pt.reranker.from_checkpoint('https://macavaney.us/scibert-medmarco.tar.gz',
    text_field='title_abstract',
    expected_md5='854966d0b61543ffffa44cea627ab63b')
```

(optional) verification of download's hash

```
def cat_title_abstract(df):
    df['title_abstract'] = df['title'].str.cat(df['abstract'], sep=' ')
    return df

bert_pipeline = (br >>
                 pt.text.get_text(dataset, ['title', 'abstract']) >>
                 pt.apply.generic(cat_title_abstract) >>
                 bert)
```

Include both the title and abstract

```
pt.Experiment(
    [br, bert_pipeline],
    dataset.get_topics('title'),
    dataset.get_qrels(),
    names=['BM25', 'BM25 >> BERT'],
    eval_metrics=["map", "ndcg", 'ndcg_cut.10', 'mrt']
)
```

TREC COVID results

Title queries

	name	map	ndcg	ndcg_cut_10	mrt
0	BM25	0.073623	0.162657	0.583665	29.487896
1	BM25 >> BERT	0.077678	0.168394	0.650975	1637.205799

Description queries

	name	map	ndcg	ndcg_cut_10	mrt
0	BM25	0.077880	0.177728	0.644374	38.557969
1	BM25 >> BERT	0.085371	0.185821	0.740331	1748.576758

Causal Language Models

BERT is trained with a Masked Language Modeling (MLM) objective.

- Words are masked out in the input and it's trained to fill in the missing words

T5, GPT, etc. are trained with a Causal Language Modeling (CLM) objective.

- Models learn to predict the next word in a sequence.
- This allows them to be used in text generation settings.

Scoring with a CLM

Università di Pisa | University of Glasgow

Main idea: Train a model to generate the text "true" or "false", prompted with the query and document:

> **Query:** coronavirus early symptoms **Document:** Nidovirus subgenomic mRNAs contain a leader sequence derived from the 5′ end of the genome fused to different sequences ('bodies') derived from the 3′ end. Their generation involves a unique mechanism of discontinuous subgenomic RNA synthesis that resembles copy-choice RNA recombination. During this process, the nascent RNA strand is transferred from one site in the template to another, during either plus or minus… **Relevant:**

Ask the model: Should the next word be "true" or "false"?

$$P(rel \mid q, d) \approx P(\text{“}true\text{”} \mid \text{“}Query\text{:”}\ \$q\ \text{“}Document\text{:”}\ \$d\ \text{“}Relevant\text{:”})$$

Train on relevant and non-relevant pairs.

Nogueira et al. Document Ranking with a Pretrained Sequence-to-Sequence Model. arXiv 2020.

MonoT5 Practical

Università di Pisa | University of Glasgow

Building a PyTerrier Neural Re-Ranking Transformer:

Input DataFrame

Batching for efficiency on GPU

Model integration

Output DataFrame

```
class MonoT5ReRanker(TransformerBase):
    def __init__(...):
        ... # sets up model, tokenizer, etc.

    def transform(self, run):
        scores = []
        queries = run['query']
        texts = run[self.text_field]
        with torch.no_grad():
            for start_idx in range(0, len(queries), self.batch_size):
                inp = self._encode(
                    queries[start_idx:start_idx+self.batch_size],
                    texts[start_idx:start_idx+self.batch_size])
                r = self.model(**inp).logits
                r = r[:, 0, (self.REL, self.NREL)]
                scores += F.log_softmax(r, dim=1)[:, 0].cpu().detach().tolist()
        run = run.assign(score=scores)
        run = add_ranks(run)
        return run

    def _encode(self, queries, texts):
        ... # builds model inputs
```

MonoT5 Practical

```
from pyterrier_t5 import MonoT5ReRanker
monoT5 = MonoT5ReRanker()
```

MonoT5 is already implemented by the pyterrier_t5 package

```
br = pt.BatchRetrieve(indexref, wmodel='BM25') % 100
pt.Experiment(
    [br, br >> pt.text.get_text(dataset, 'text') >> monoT5],
    dataset.get_topics(),
    dataset.get_qrels(),
    names=['BM25', 'BM25 >> monoT5'],
    eval_metrics=[MAP, RR, P@10, nDCG@10]
)
```

monoT5 batches: 100% 2325/2325 [01:36<00:00, 24.00it/s]

	name	map	recip_rank	P_10	ndcg_cut_10
0	BM25	0.272523	0.725587	0.352688	0.446609
1	BM25 >> monoT5	0.295071	0.750839	0.413978	0.490703

Part 3C

EFFICIENCY

Problem: Efficiency

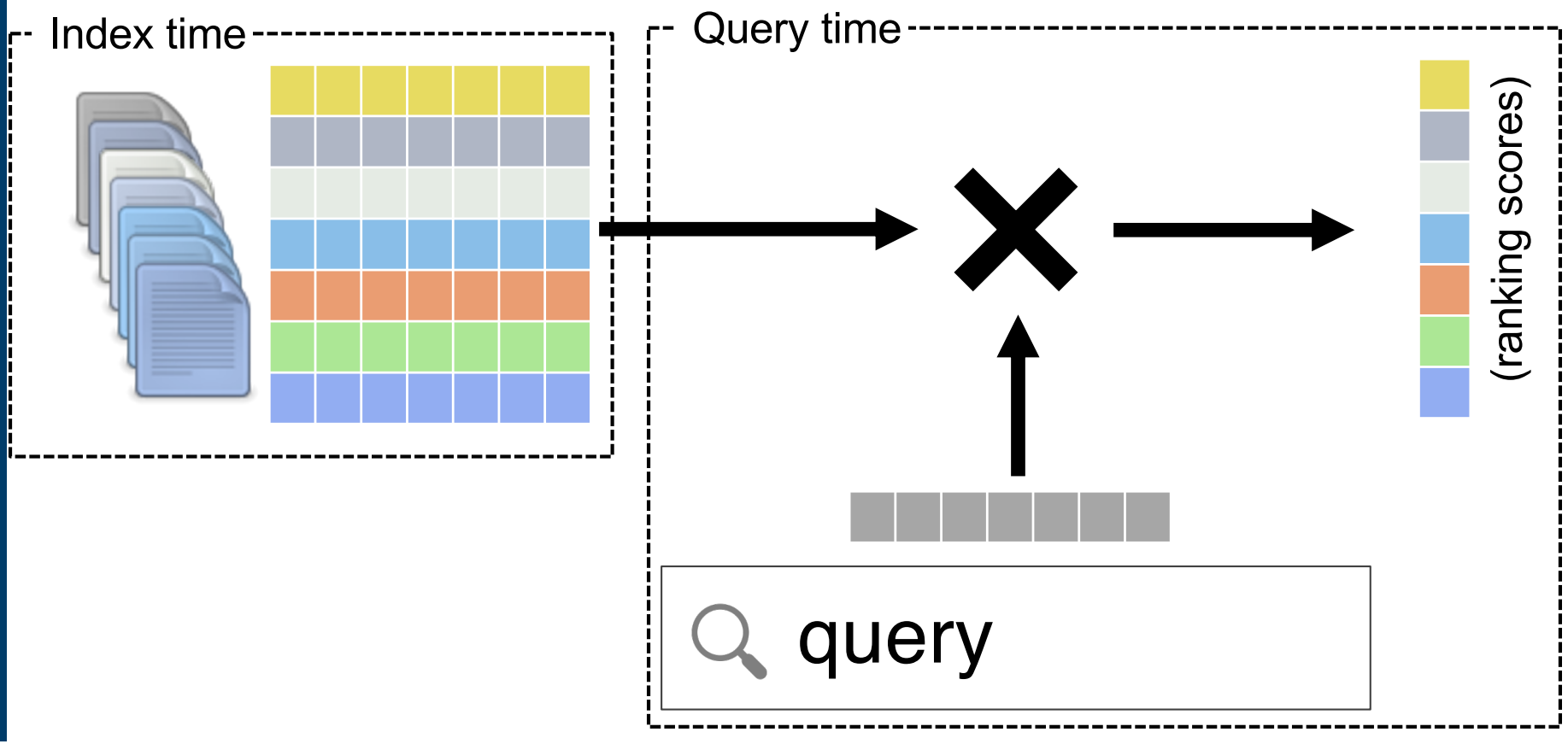
Università di Pisa | University of Glasgow

Using methods like BERT and T5 for ranking is effective, but also very slow.

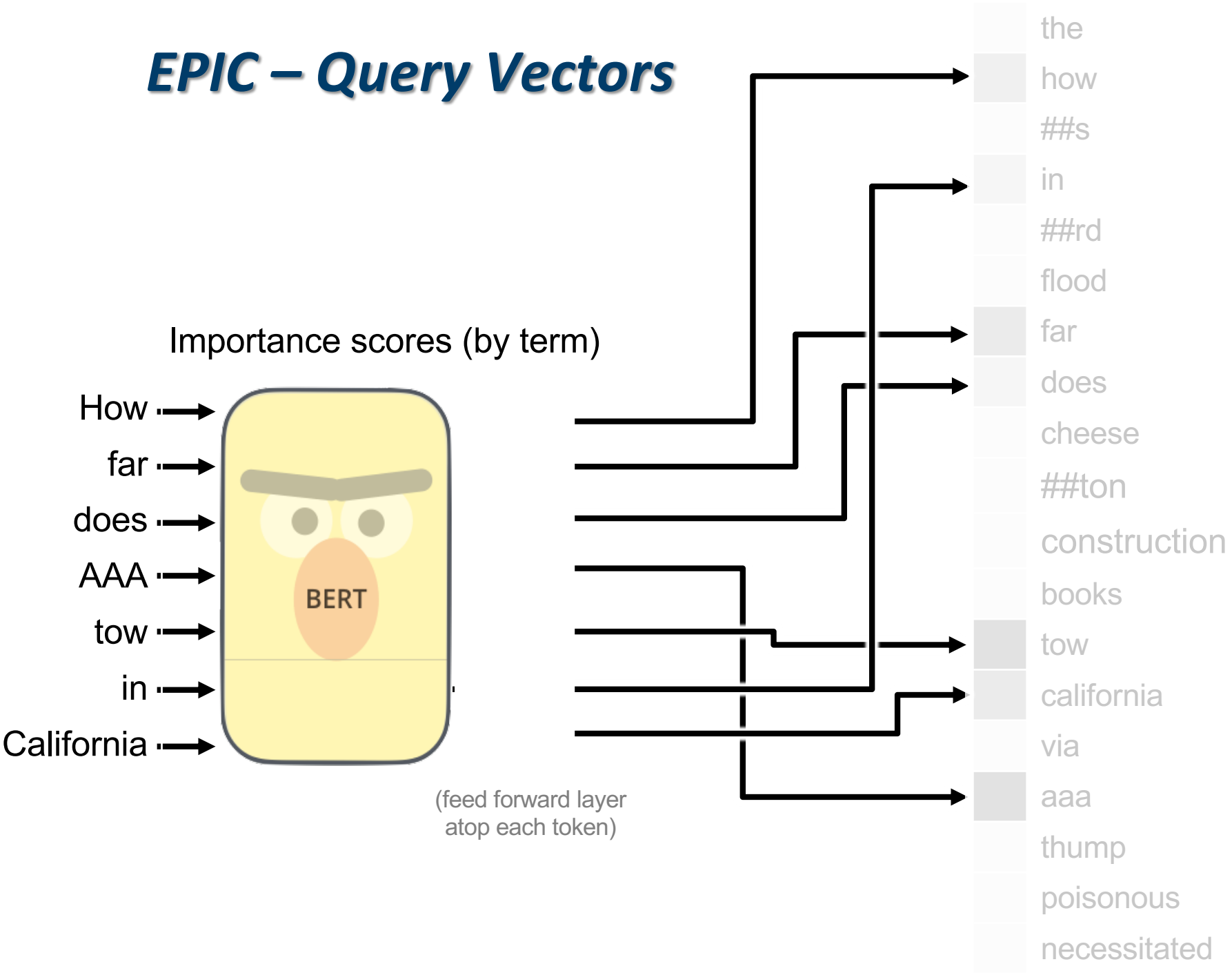
E.g., BERT takes 45x longer than BM25: (1.7 seconds)

	name	map	ndcg	ndcg_cut_10	mrt
0	BM25	0.077880	0.177728	0.644374	38.557969
1	BM25 >> BERT	0.085371	0.185821	0.740331	1748.576758

Main idea: Sparse representations that are fast to score.

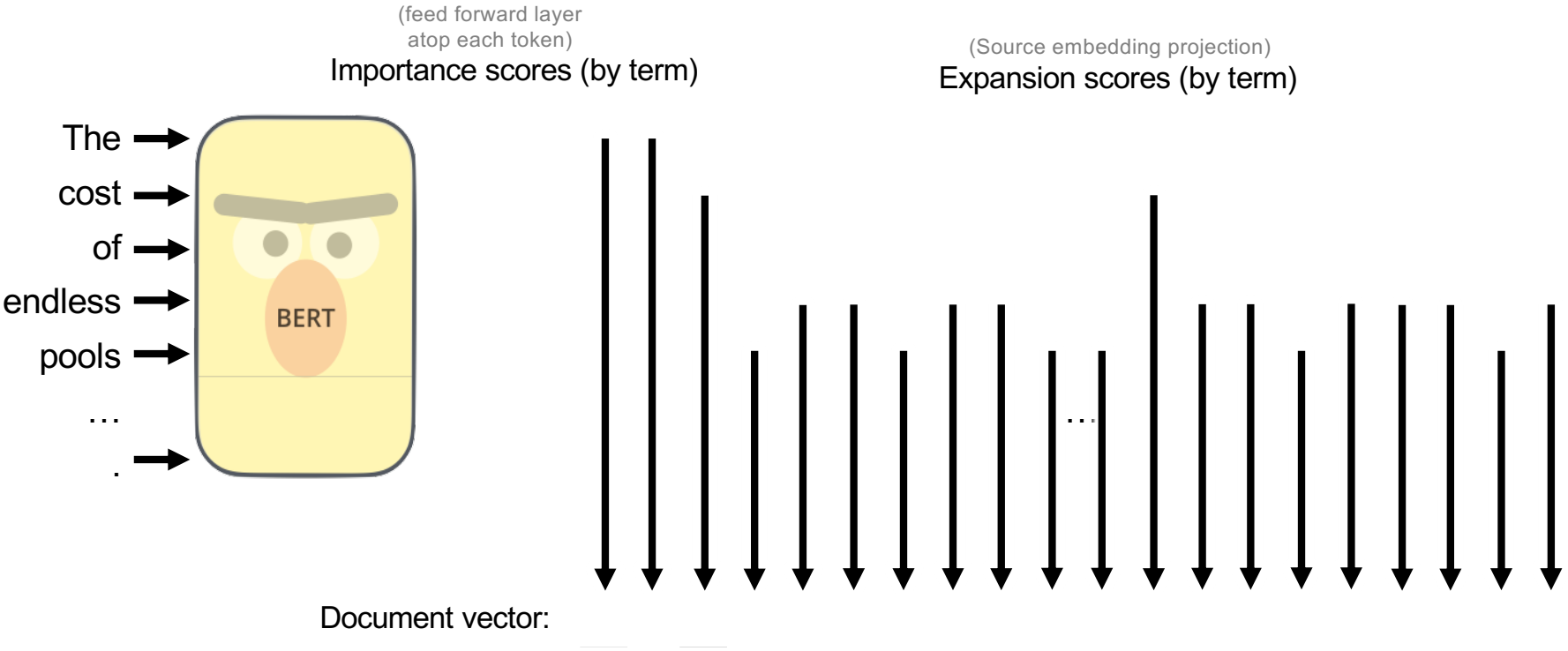


EPIC – Query Vectors



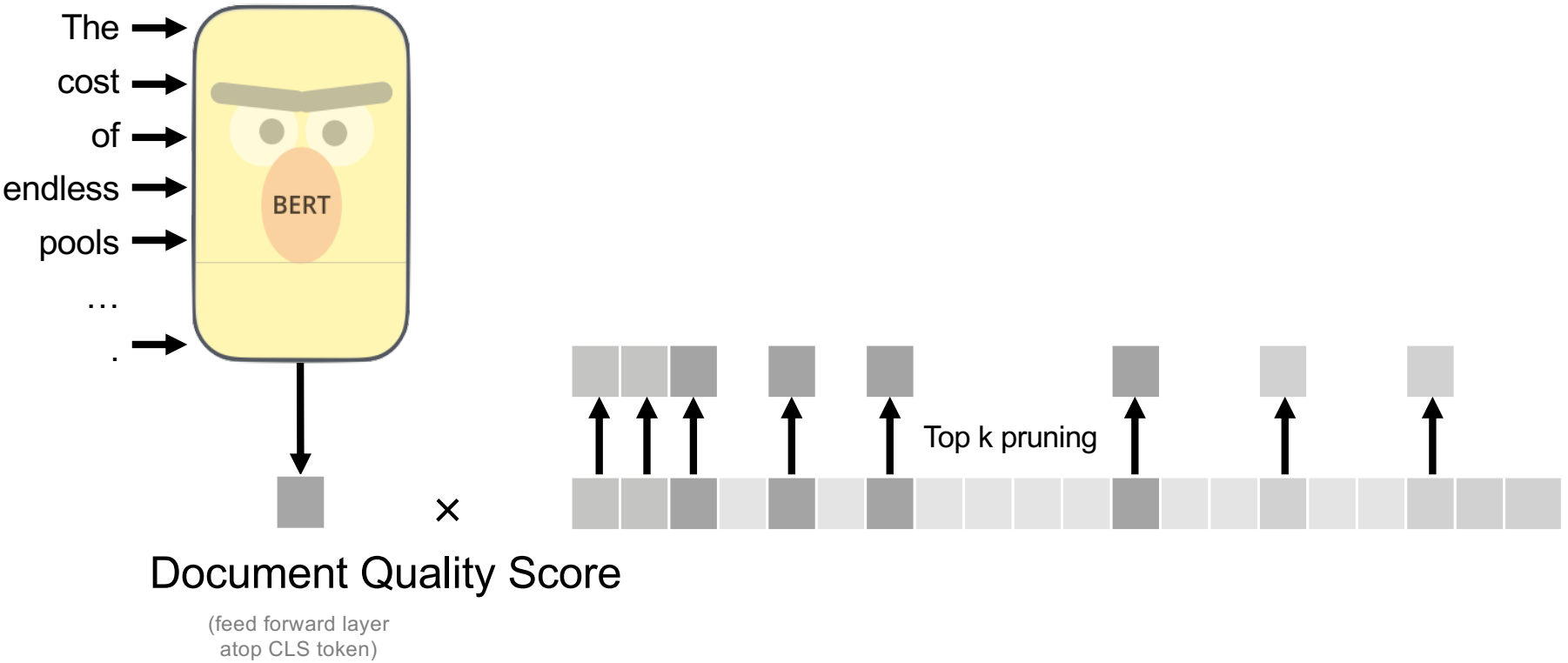
BERT Graphics from Jay Alammar, http://jalammar.github.io/illustrated-bert/

EPIC – Document Vectors



BERT Graphics from Jay Alammar, http://jalammar.github.io/illustrated-bert/

EPIC – Document Vectors



BERT Graphics from Jay Alammar, http://jalammar.github.io/illustrated-bert/

EPIC Practical

Lazy EPIC (not indexed)

```
# Load a version of EPIC trained on the MS-MARCO dataset
lazy_epic = onir_pt.reranker.from_checkpoint(
    'https://macavaney.us/epic.msmarco.tar.gz',
    expected_md5="2f6a16be1a6a63aab1e8fed55521a4db")
```

```
br = pt.BatchRetrieve(index) % 30
pipeline = (br >> pt.text.get_text(dataset, 'abstract')
               >> pt.apply.generic(lambda x: x.rename(columns={'abstract': 'text'}))
               >> lazy_epic)
pt.Experiment(
    [br, pipeline],
    dataset.get_topics('title'),
    dataset.get_qrels(),
    names=['DPH', 'DPH >> EPIC (lazy)'],
    eval_metrics=["recip_rank", "P.5", "mrt"]
)
```

	name	recip_rank	P_5	mrt
0	DPH	0.766833	0.684	36.734074
1	DPH >> EPIC (lazy)	0.817889	0.724	801.524438

EPIC Practical

Università di Pisa | University of Glasgow

EPIC (indexed)

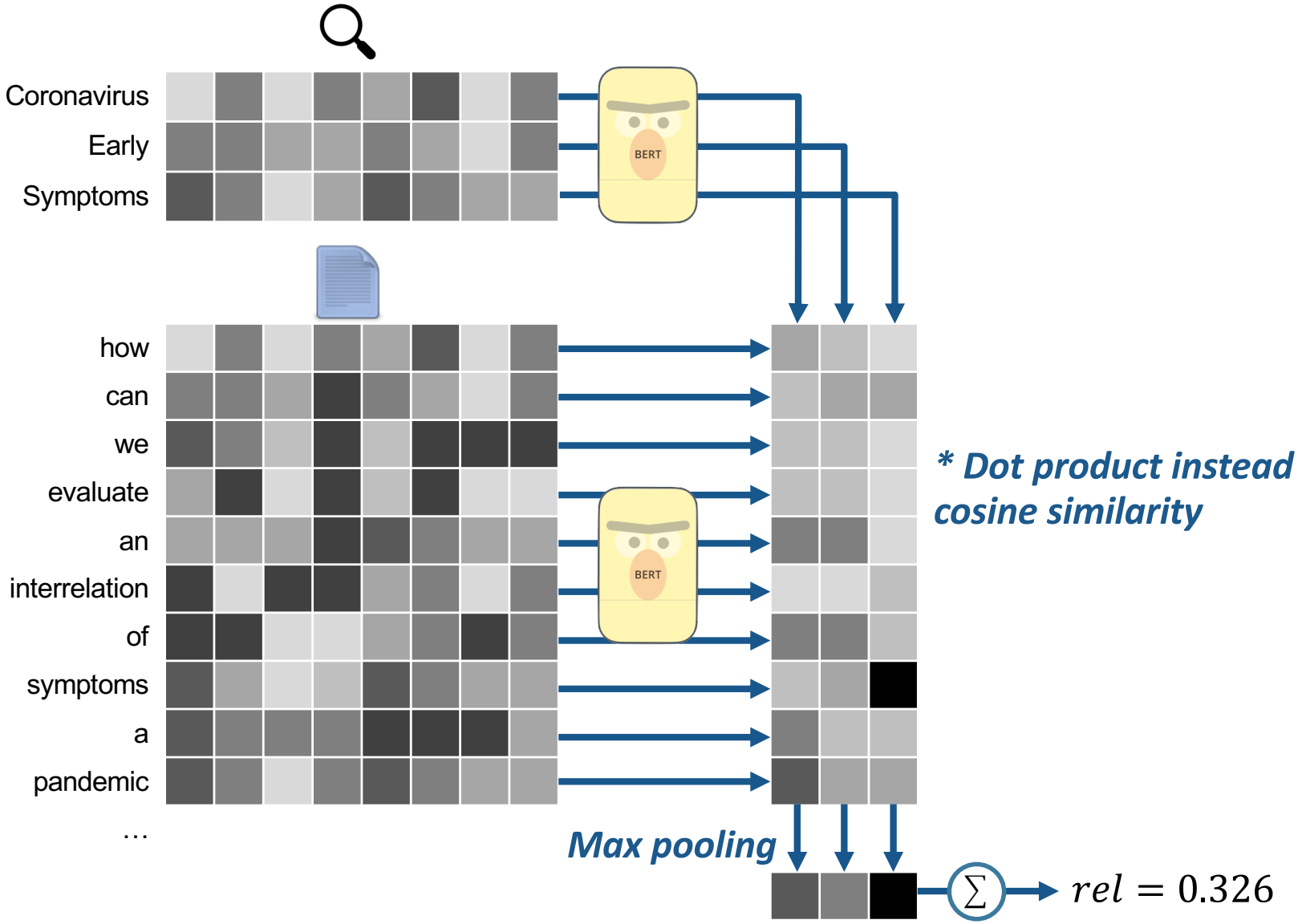
```
indexed_epic = onir_pt.indexed_epic.from_checkpoint(
    'https://macavaney.us/epic.msmarco.tar.gz',
    index_path='./epic_cord19')
```

```
indexed_epic.index(dataset.get_corpus_iter(), fields=('abstract',))
```

```
pipeline = br >> indexed_epic.reranker()
pt.Experiment(
    [br, pipeline],
    dataset.get_topics('title'),
    dataset.get_qrels(),
    names=["DPH", "DPH >> EPIC (indexed)"],
    eval_metrics=["recip_rank", "P.5", "mrt"]
)
```

	name	recip_rank	P_5	mrt
0	DPH	0.766833	0.684	30.500175
1	DPH >> EPIC (indexed)	0.821500	0.700	53.264584

Contextualized Late Interaction over BERT (ColBERT)



Khattab & Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. SIGIR 2020.

ColBERT Re-Ranking practical

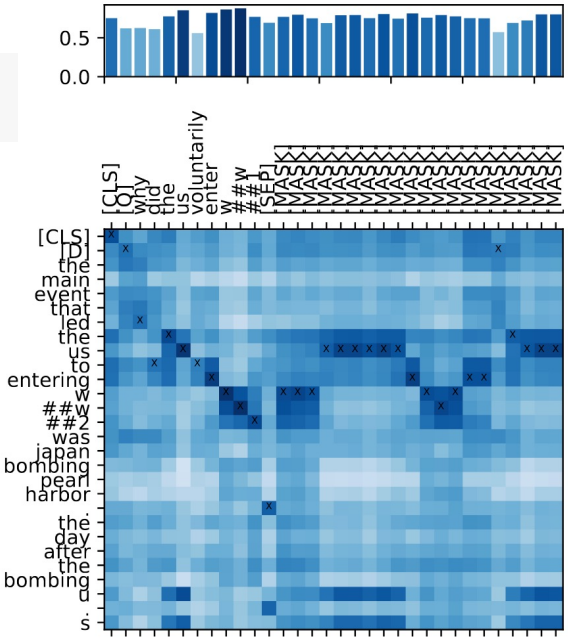
```
colbert_factory = pyterrier_colbert.ranking.ColBERTFactory(
    "http://www.dcs.gla.ac.uk/~craigm/colbert.dnn.zip", None, None)
```

```
colbert = colbert_factory.text_scorer(doc_attr='abstract')
```

```
br = pt.BatchRetrieve(index) % 100
pipeline = br >> pt.text.get_text(dataset, 'abstract') >> colbert
```

	name	map	P_10	ndcg	ndcg_cut_10	mrt
0	DPH	0.068006	0.658	0.165607	0.609058	52.517634
1	DPH >> ColBERT	0.074670	0.752	0.172034	0.690447	615.640942

```
colbert_factory.explain_text(query, text)
```



Macdonald, Tonellotto & Ounis. On Single and Multiple Representations in Dense Passage Retrieval. IIR 2021.

Other Re-Ranking Models

There's a bunch of other great re-ranking models:

- **TK/TKL – Contextualized kernel-based ranking.**
  - Hofstätter et al. Interpretable & time-budget-constrained contextualization for re-ranking. ECAI 2020.
- **PARADE – Handling long documents through passage representation aggregation**
  - Li et al. PARADE: Passage Representation Aggregation for Document Reranking. arXiv 2020.
- **Duet – combine representation and interaction models.**
  - Mitra et al. Learning to match using local and distributed representations of text for web search. WWW 2017.
- **And others…**

Part 3D

NEURAL INDEX AUGMENTATION

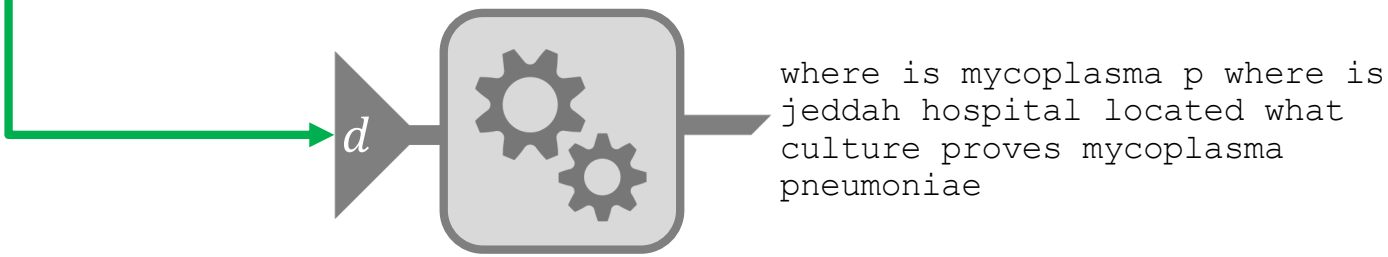
Document Augmentation

E.g., Inverted Index

docno	text
ug7v899j	OBJECTIVE: This retrospective chart review describes the…
02tnwd4m	Inflammatory diseases of the respiratory tract are common…
ejv2xln0	Endothelin-1 (ET-1) is a 21 amino acid peptide with diverse…
…	…

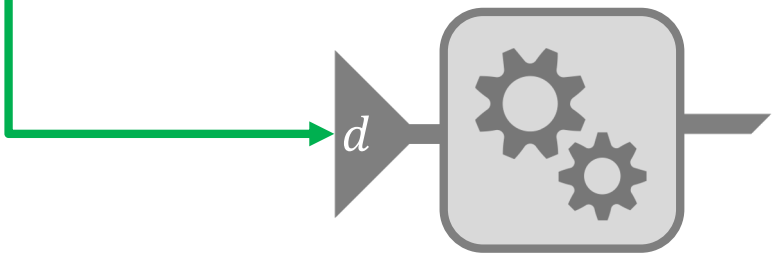
Document Augmentation

docno	text
ug7v899j	OBJECTIVE: This retrospective chart review describes the…
02tnwd4m	Inflammatory diseases of the respiratory tract are common…
ejv2xln0	Endothelin-1 (ET-1) is a 21 amino acid peptide with diverse…
…	…



Document Augmentation

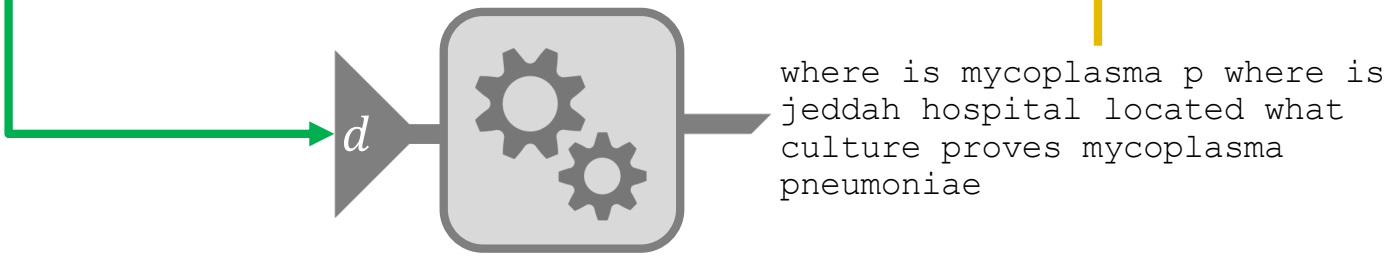
docno	text	newfield
ug7v899j	OBJECTIVE: This retrospective chart review describes the…	where is mycoplasma p where is jeddah…
02tnwd4m	Inflammatory diseases of the respiratory tract are common…	
ejv2xln0	Endothelin-1 (ET-1) is a 21 amino acid peptide with diverse…	
…	…	




```
where is mycoplasma p where is
jeddah hospital located what
culture proves mycoplasma
pneumoniae
```

Document Augmentation

docno	text
ug7v899j	OBJECTIVE: This retrospective chart review describes the… where is mycoplasma p where is jeddah…
02tnwd4m	Inflammatory diseases of the respiratory tract are common…
ejv2xln0	Endothelin-1 (ET-1) is a 21 amino acid peptide with diverse…
…	…



Augmenting the Inverted Index

Università di Pisa | University of Glasgow

We can use neural models to modify the content stored in classical inverted indices.

Two main strategies:

- Emphasizing important terms (DeepCT)
- Generating additional terms to index (doc2query)

DeepCT

Università di Pisa | University of Glasgow

Main idea: Boost the term frequency of important words by repeating them in the text.

Abstract:
Nidovirus subgenomic mRNAs contain a leader sequence derived from the 5′ end of the genome fused to different sequences (‘bodies’) derived from the 3′ end. Their generation involves a unique mechanism of discontinuous subgenomic RNA synthesis that resembles copy-choice RNA recombination. During this process, the nascent RNA strand is transferred from one site in the template to another, during either plus or minus…

Abstract:
Nidovirus Nidovirus Nidovirus subgenomic subgenomic subgenomic mRNAs mRNAs mRNAs contain a leader sequence derived from the 5′ end of the genome fused to different sequences (‘bodies’) derived from the 3′ end. Their generation involves a unique mechanism of discontinuous **subgenomic subgenomic RNA RNA** synthesis that resembles copy-choice **RNA RNA RNA** recombination. During this process, the nascent **RNA RNA RNA** strand is transferred from one site in the template to another, during either plus or minus…

Training: Predict the terms that match relevant query terms.

Dai & Callan. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval. arXiv 2019.

DeepCT practical

```
deepct = pyterrier_deepct.DeepCTTransformer(
    "bert-base-uncased/bert_config.json",
    "marco/model.ckpt-65816")
```

Example DeepCT outputs:

```
df.iloc[0]['text']
```

```
'OBJECTIVE: This retrospective chart review describes the epidemiology and clinical
features of 40 patients with culture-proven Mycoplasma pneumoniae infections at King
Abdulaziz University Hospital, Jeddah, Saudi Arabia. METHODS: Patients with positive
M. pneumoniae cultures from respiratory specimens from January 1997 through December
```

```
deepct.transform(df).iloc[0]["text"]
```

```
'objective objective retrospective retrospective chart chart chart chart chart revie
w describes epidemiology epidemiology epidemiology epidemiology epidemiology epidemi
ology epidemiology epidemiology epidemiology clinical clinical clinical clinical cli
nical clinical clinical clinical features features features features patients patien
```

Doc2Query

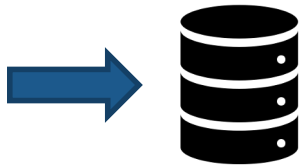
Main idea: Use a causal language model to generate additional text to add to documents when indexing.

At retrieval time, use standard retrieval model (e.g., BM25).

How are the queries generated?

Abstract:
Nidovirus subgenomic mRNAs contain a leader sequence derived from the 5′ end of the genome fused to different sequences (‘bodies’) derived from the 3′ end. Their generation involves a unique mechanism of discontinuous subgenomic RNA synthesis that resembles copy-choice RNA recombination. During this process, the nascent RNA strand is transferred from one site in the template to another, during either plus or minus…

Generated Queries:
where does mrna originate
where does subgenomic rna come from
where is the nidovirus mrna?
when a nidovirus is produced, its genome is quizlet
what is nidovirus mrna
where in a nidovirus can one mrna be derived
where is the leader sequence derived
what types of mrna are found in nidovirus
where are the nidovirus genomes derived from
…



Nogueira & Lin. From doc2query to docTTTTTquery. 2019.

Doc2Query

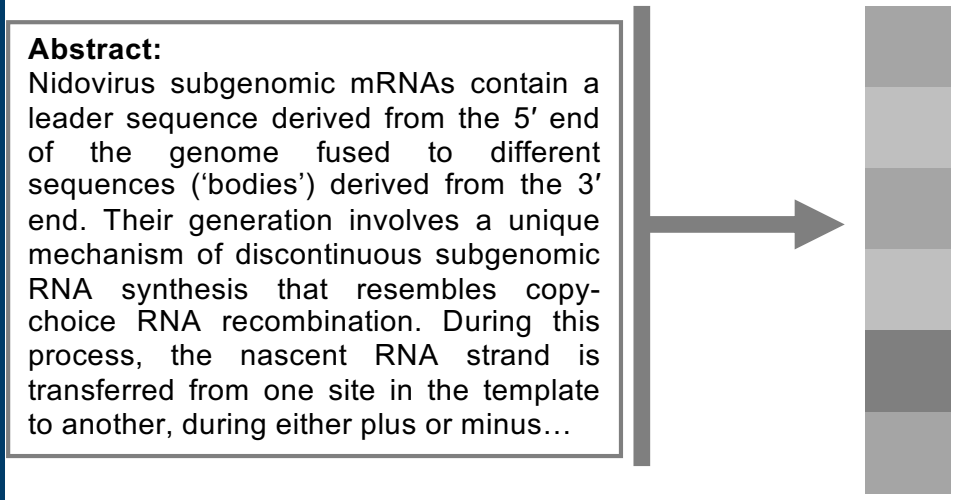
Università di Pisa | University of Glasgow

Training: optimize to generate queries from passages on a collection with many queries (e.g., MS-MARCO)

Most recent version: Use T5 model for generation (docTTTTTquery)

Nogueira & Lin. From doc2query to docTTTTTquery. 2019.

Doc2Query

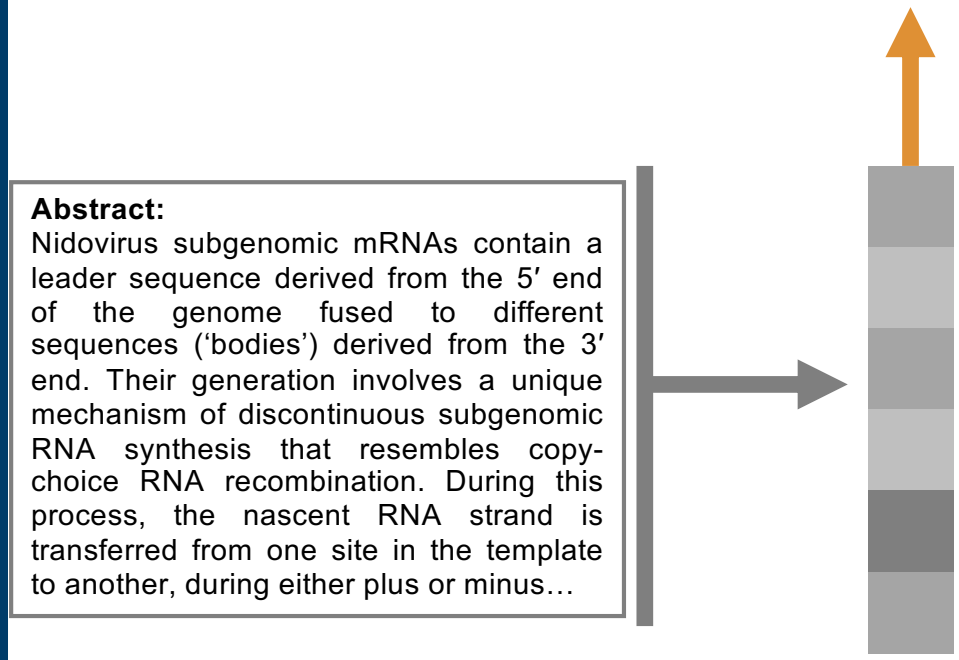


Step 1: Source text encoded

e.g., via RNN or Transformer

Nogueira & Lin. From doc2query to docTTTTTquery. 2019.

Doc2Query

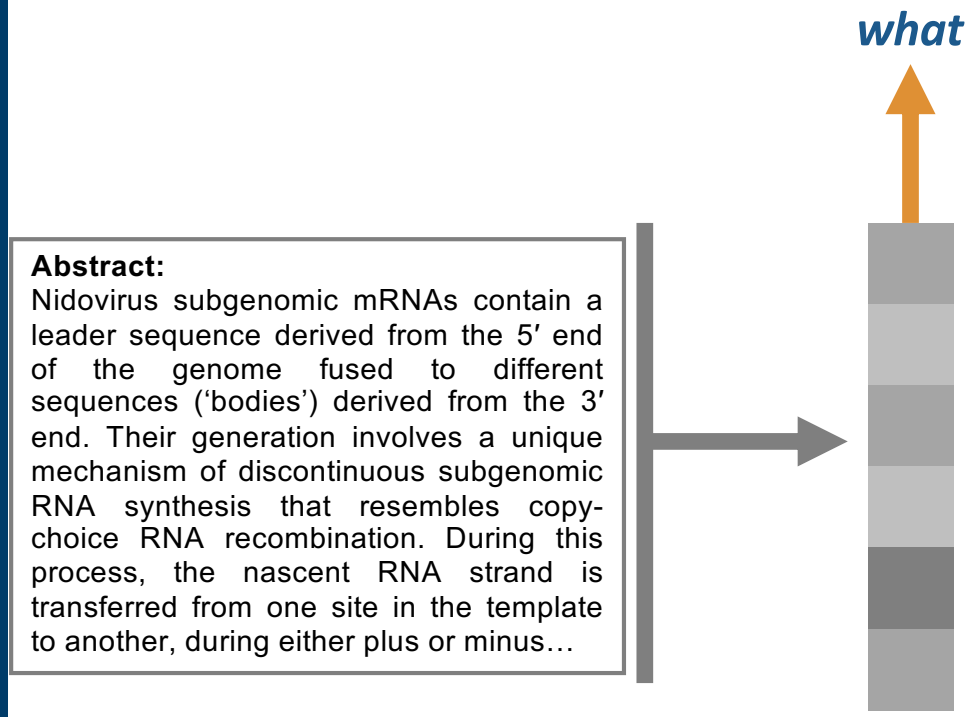


Step 2: Text iteratively generated from encoded document

e.g., via RNN or transformer, using beam search

Nogueira & Lin. From doc2query to docTTTTTquery. 2019.

Doc2Query

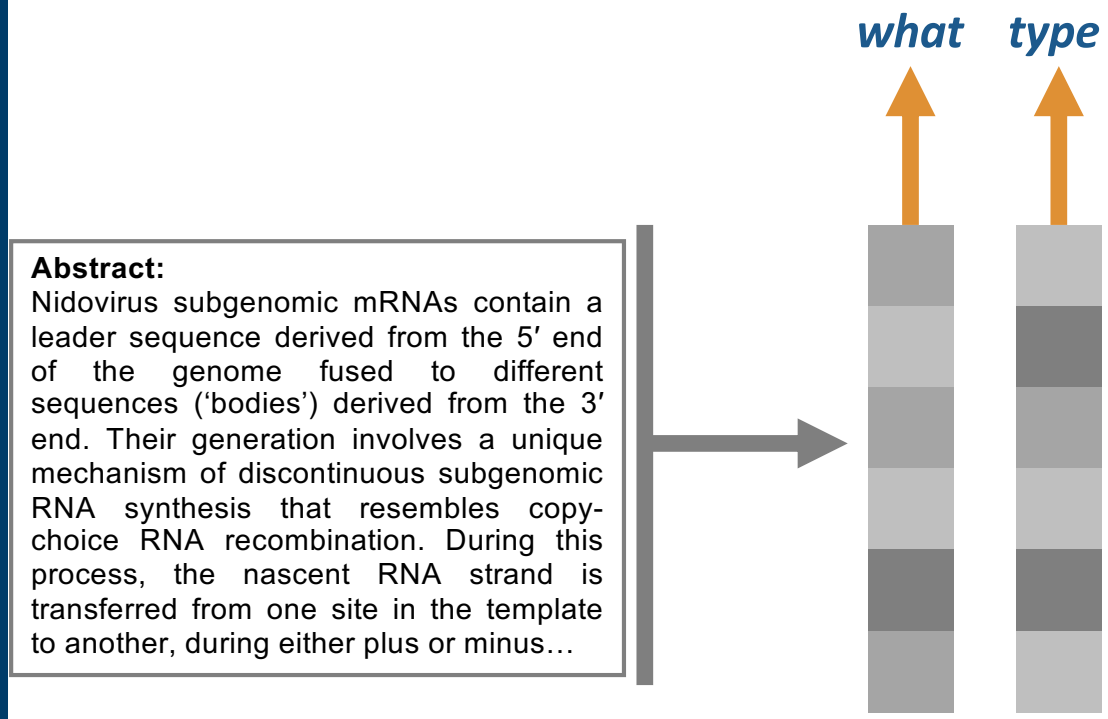


Step 2: Text iteratively generated from encoded document

e.g., via RNN or transformer, using beam search

Nogueira & Lin. From doc2query to docTTTTTquery. 2019.

Doc2Query

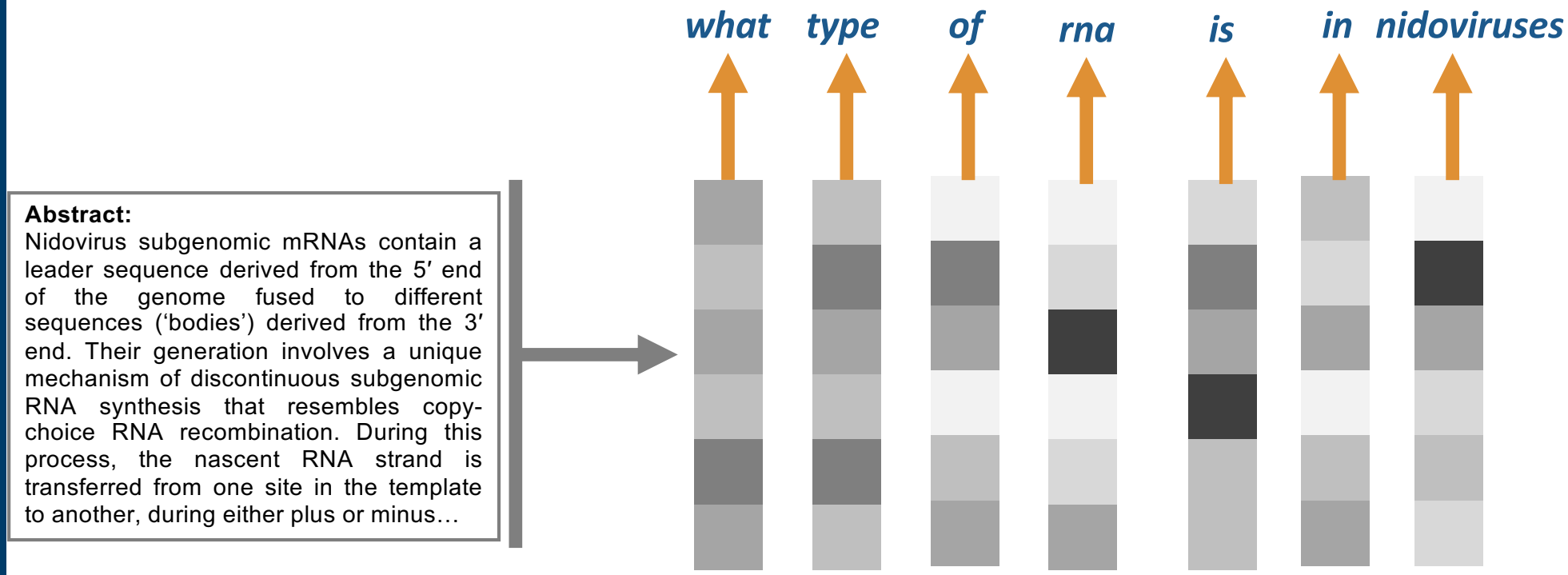


Step 2: Text iteratively generated from encoded document

e.g., via RNN or transformer, using beam search

Nogueira & Lin. From doc2query to docTTTTTquery. 2019.

Doc2Query



Step 2: Text iteratively generated from encoded document

e.g., via RNN or transformer, using beam search

Nogueira & Lin. From doc2query to docTTTTTquery. 2019.

Doc2Query Practical

```
from pyterrier_doc2query import Doc2Query
doc2query = Doc2Query(out_attr="text", batch_size=8)
```

Example doc2query outputs

```
df.iloc[0]['text']
```

```
'OBJECTIVE: This retrospective chart review describes the epidemiology and
clinical features of 40 patients with culture-proven Mycoplasma pneumoniae
infections at King Abdulaziz University Hospital, Jeddah, Saudi Arabia. MET
HODS: Patients with positive M. pneumoniae cultures from respiratory specim
```

```
doc2query.transform(df).iloc[0]['text']
```

```
'what is culture confirmed mycoplasma where is mycoplasma pneumonia located
in saudi arabia? where is mycoplasma cultured'
```

University of Glasgow | Università di Pisa

CIKM 2021 1-5 NOVEMBER

Part 3E

WRAPUP

Summary

Re-ranking approaches continue to be an appealing and active area of research:

- Handle vocabulary mismatch and text semantics
- Straightforward and highly effective
- Query latency can be managed
- Limitations:
  - Documents not retrieved in the first stage cannot be re-ranked
  - A higher-quality initial ranking means you need a lower retrieval threshold (which means less compute)

Index augmentation can bring benefits from deep learning to traditional indices

- Emphasize important terms, predict other terms that could match

PyTerrier Supported Re-Rankers

Università di Pisa | University of Glasgow

Model	Provider	Re-Ranking	Training	Indexing
DRMM	OpenNIR	✅	✅	
KNRM		✅	✅	
ConvKNRM		✅	✅	
PACRR		✅	✅	
MatchPyramid		✅	✅	
Vanilla BERT		✅	✅	
CEDR		✅	✅	
EPIC		✅	✅	✅
monoT5	pyterrier_t5	✅	Coming soon	
duoT5		✅		
ColBERT	pyterrier_colbert	✅		✅ (pt4)
DeepCT	pyterrier_deepct			✅
doc2query	pyterrier_doc2query			✅

Easy to add a PyTerrier wrapper for your model for use in pipelines!

QUESTIONS?

What's the task in the notebooks?

Università di Pisa | University of Glasgow

In the notebooks, you will experience:

- Building untrained re-ranking models
- Using pre-trained BERT and T5 models
- Scoring with EPIC using pre-computed document vecs
- Tuning re-ranking thresholds
- Applying index augmentation approaches

Practical Time

The tutorial Github repo has links to the notebook for Part 3

- https://github.com/terrier-org/cikm2021tutorial
- Press the 

link for each notebook to start a Colab session

Timings:

- Practical – in breakout rooms – 14:30-15:00
- Coffee break: 15:00-15:30
- Part 4 resumes at 15:30

Run 1 times (GMT)

If you are leaving us here, please complete our feedback quiz https://forms.office.com/r/RiYSAxAKhk!

References

DRMM: *Guo, et al. A Deep Relevance Matching Model for Ad-hoc Retrieval. CIKM 2017.*

KNRM: *Xiong, et al. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. SIGIR 2017.*

PACRR: *Hui, et al. PACRR: A Position-Aware Neural IR Model for Relevance Matching. EMNLP 2017.*

BERT: *Devlin, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019.*

CEDR: *MacAvaney, et al. CEDR: Contextualized Embeddings for Document Ranking. SIGIR 2019*.

MonoT5: *Nogueira, et al. Document Ranking with a Pretrained Sequence-to-Sequence Model. arXiv 2020.*

EPIC: *MacAvaney, et al. Expansion via Prediction of Importance with Contextualization. SIGIR 2020.*

ColBERT: *Khattab & Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. SIGIR 2020.*

doc2query: *Nogueira & Lin. From doc2query to docTTTTTquery.*

DeepCT: *Dai & Callan. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval.*