NEURAL MODELS FOR INFORMATION RETRIEVAL

November, 2017

BHASKAR MITRA

Principal Applied Scientist
Microsoft AI and Research

Research Student
Dept. of Computer Science
University College London

Microsoft

UCL

This talk is based on work done in collaboration with

Nick Craswell, Fernando Diaz, Emine Yilmaz, Rich Caruana, Eric Nalisnick, Hamed Zamani, Christophe Van Gysel, Nicola Cancedda, Matteo Venanzi, Saurabh Tiwary, Xia Song, Laura Dietz, Federico Nanni, Matt Magnusson, Roy Rosemarin, Grzegorz Kukla, Piotr Grudzien, and many others

For a general overview of neural IR refer to the manuscript under review for

Foundations and Trends® in Information Retrieval

Pre-print is available for free download

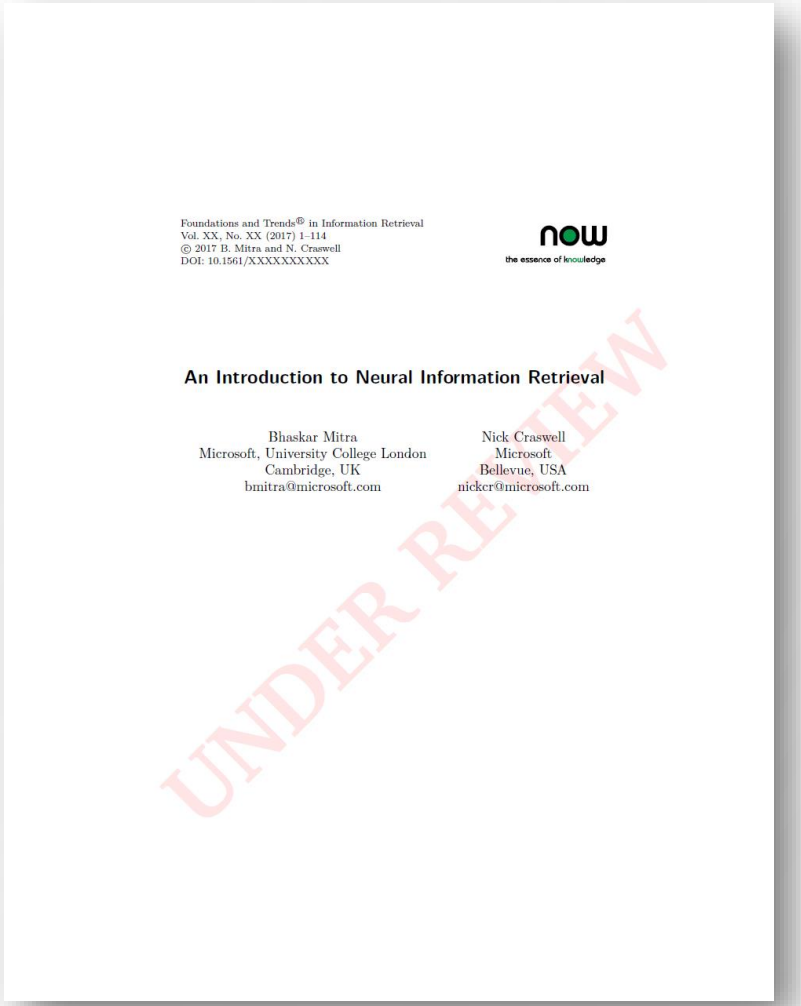
http://bit.ly/neuralir-intro

Final manuscript may contain additional content and changes

Or check out the presentations from these recent tutorials,

WSDM 2017 tutorial: http://bit.ly/NeuIRTutorial-WSDM2017

SIGIR 2017 tutorial: http://nn4ir.com/



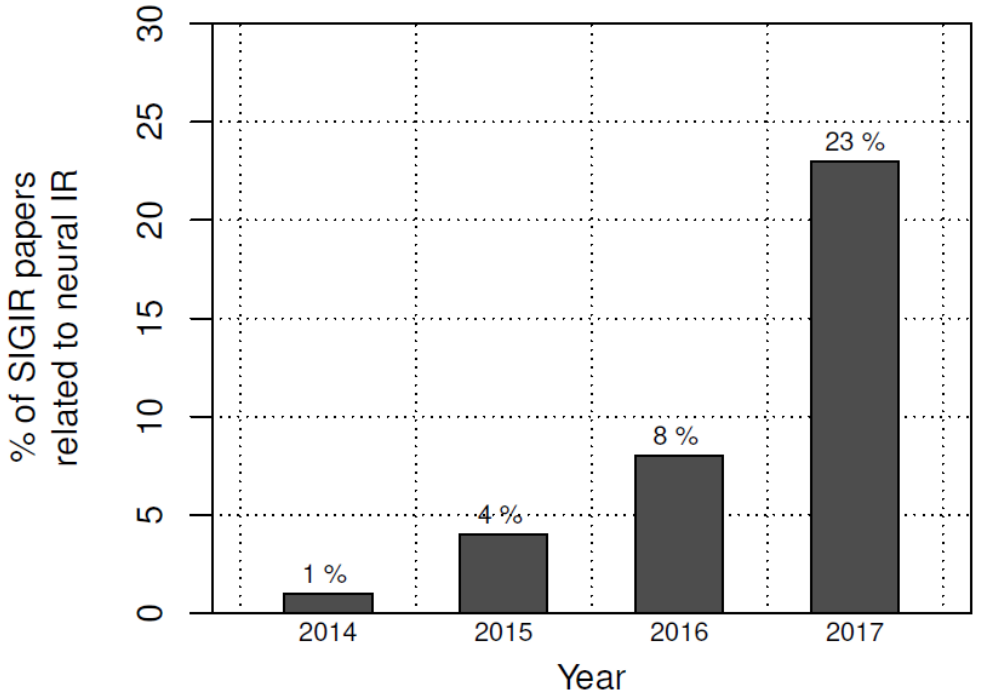
NEURAL NETWORKS

Amazingly successful on many difficult application areas

Dominating multiple fields:

2011	2013	2015	2017
speech	vision	NLP	IR?

Each application is different, motivates new innovations in machine learning

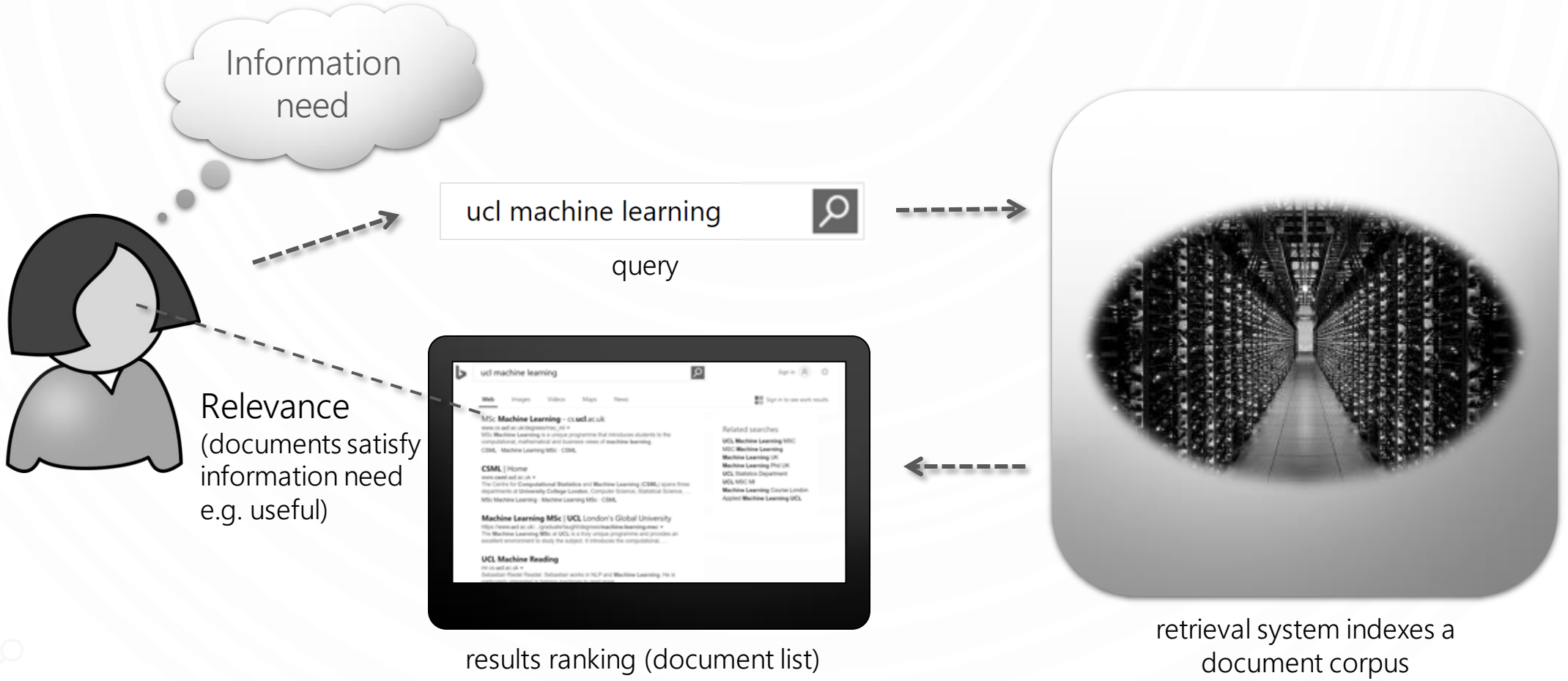


Source: (Mitra and Craswell, 2018)
https://www.microsoft.com/en-us/research/...

Our research:

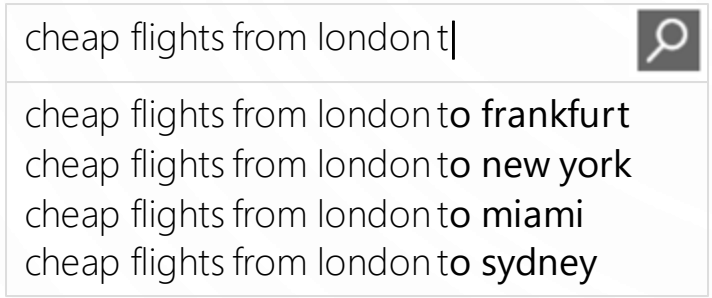
Novel representation learning methods and neural architectures motivated by specific needs and challenges of IR tasks

IR TASK MAY REFER TO DOCUMENT RANKING

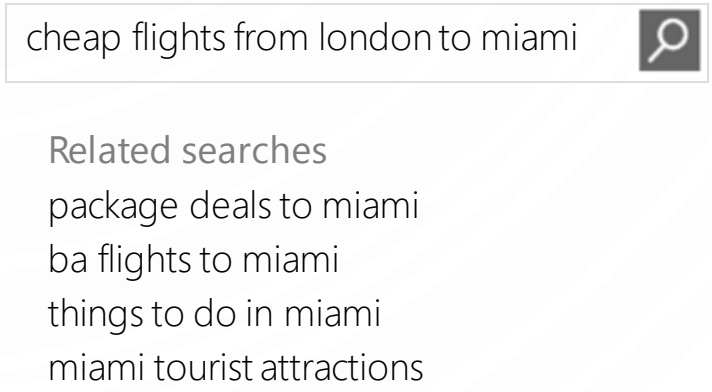


BUT IT MAY ALSO REFER TO…

QUERY AUTO-COMPLETION



NEXT QUERY SUGGESTION

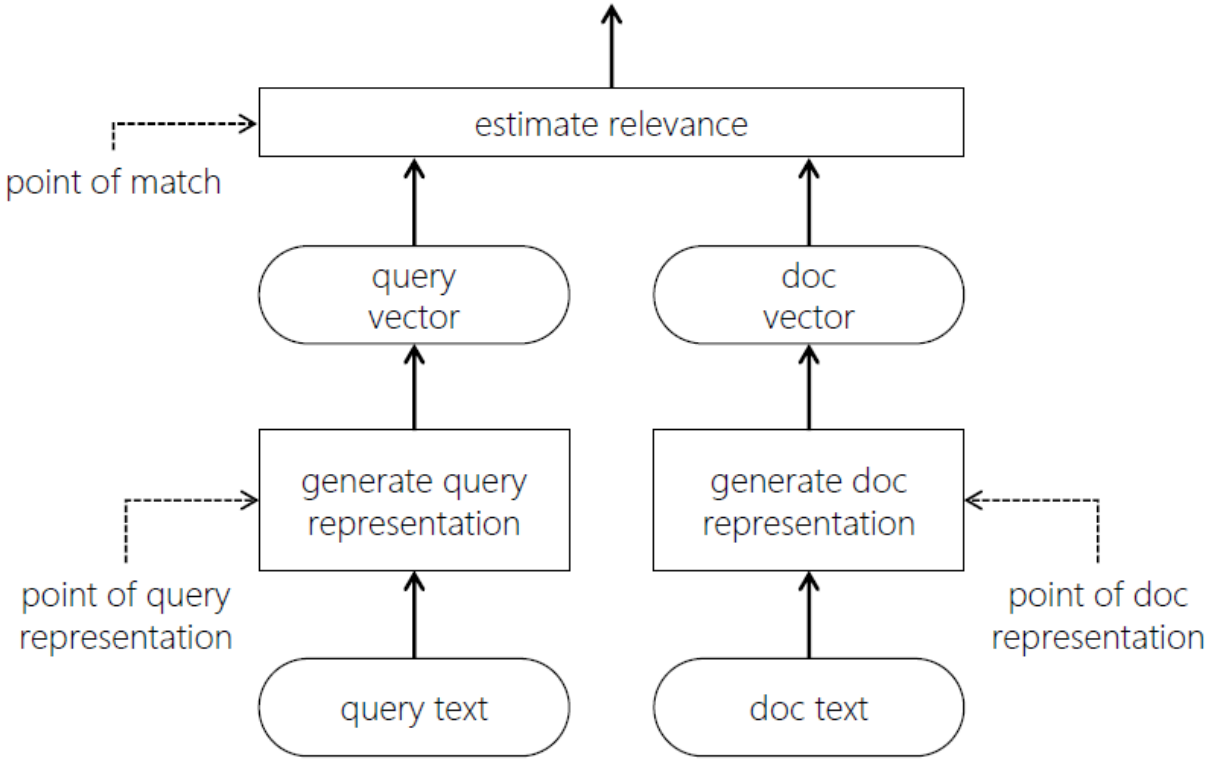


ANATOMY OF AN IR MODEL

IR in three simple steps:

1. Generate input (query or prefix) representation
2. Generate candidate (document or suffix or query) representation
3. Estimate relevance based on input and candidate representations

Neural networks can be useful for one or more of these steps

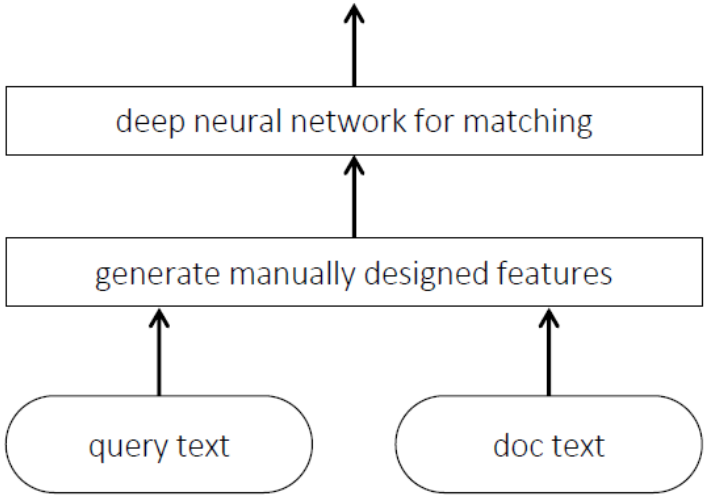


NEURAL NETWORKS CAN HELP WITH…

Learning a matching function on top of traditional feature based representation of query and document

But it can also help with learning good representations of text to deal with vocabulary mismatch

In this part of the talk, we focus on learning good vector representations of text for retrieval



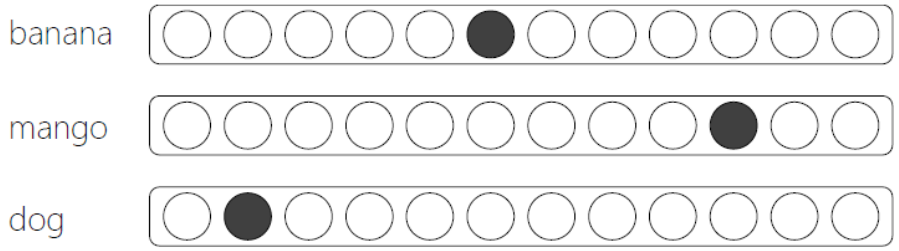
A QUICK REFRESHER ON

VECTOR SPACE REPRESENTATIONS

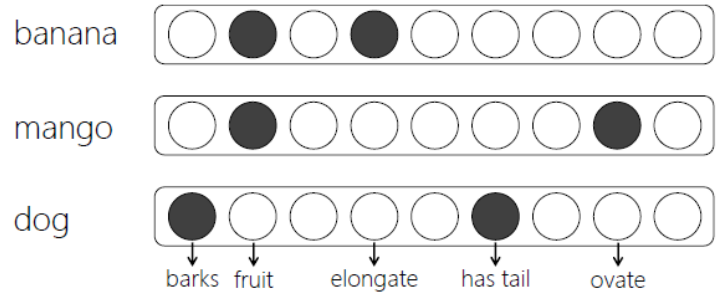
Under local representation the terms banana, mango, and dog are distinct items

But distributed representation (e.g., project items to a feature space) may recognize that banana and mango are both fruits, but dog is different

Important note: the choice of features defines what items are similar



(a) Local representation



(b) Distributed representation

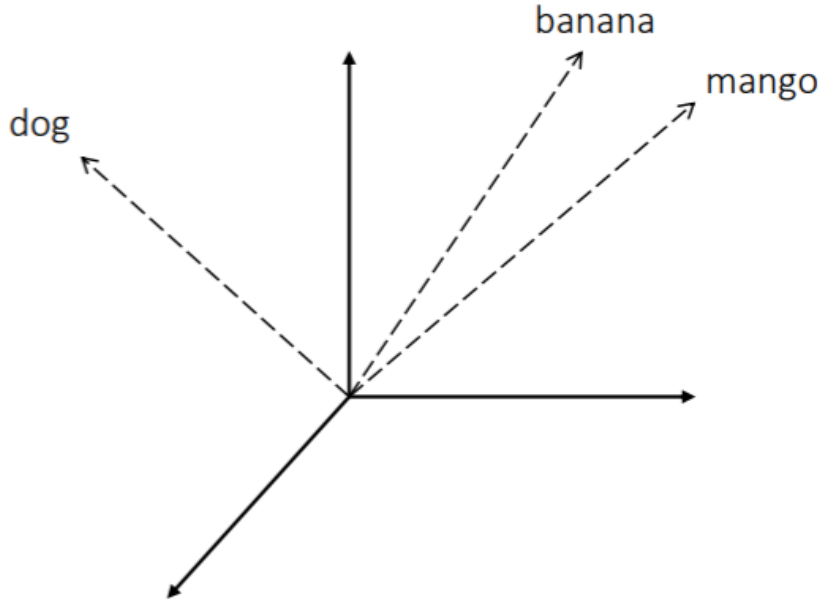
A QUICK REFRESHER ON

VECTOR SPACE REPRESENTATIONS

An *embedding* is a new space such that the properties of, and the relationships between, the items are preserved

Compared to original feature space an embedding space may have one or more of the following:

- Less number of dimensions
- Less sparseness
- Disentangled principle components



NOTIONS OF SIMILARITY

Is “Seattle” more similar to…

“Sydney” (similar *type*)

Or

“Seahawks” (similar *topic*)

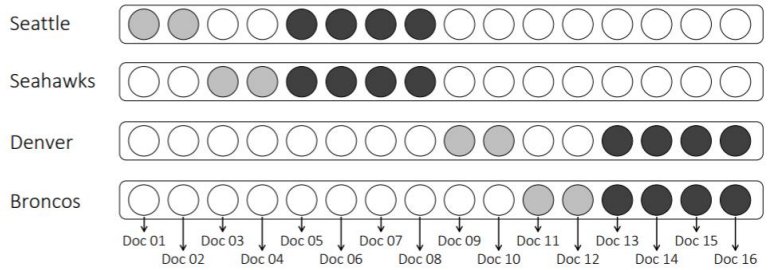
Depends on what feature space you choose

NOTIONS OF SIMILARITY

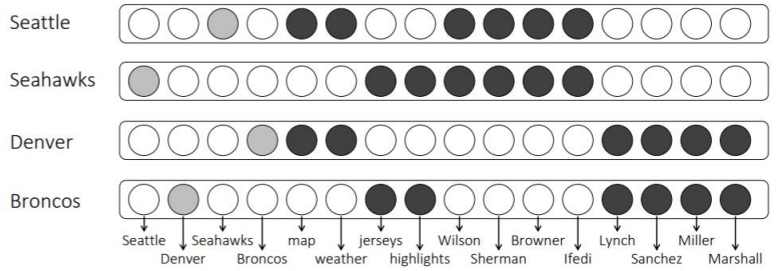
Consider the following toy corpus…

doc 01	Seattle map	doc 09	Denver map
doc 02	Seattle weather	doc 10	Denver weather
doc 03	Seahawks jerseys	doc 11	Broncos jerseys
doc 04	Seahawks highlights	doc 12	Broncos highlights
doc 05	Seattle Seahawks Wilson	doc 13	Denver Broncos Lynch
doc 06	Seattle Seahawks Sherman	doc 14	Denver Broncos Sanchez
doc 07	Seattle Seahawks Browner	doc 15	Denver Broncos Miller
doc 08	Seattle Seahawks Ifedi	doc 16	Denver Broncos Marshall

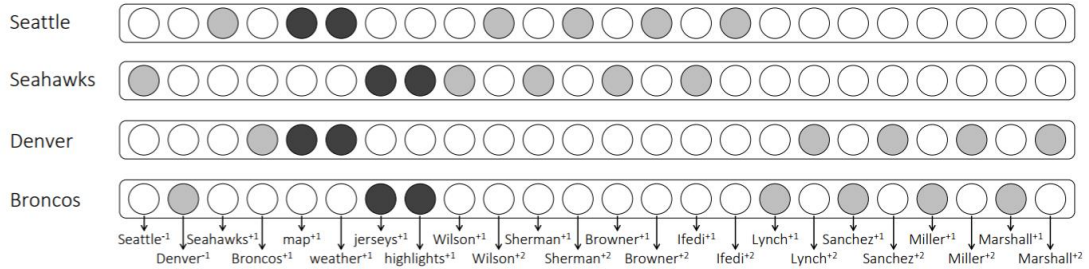
Now consider the different vector representations of terms you can derive from this corpus and how the items that are similar differ in these vector spaces



(a) "In-documents" features



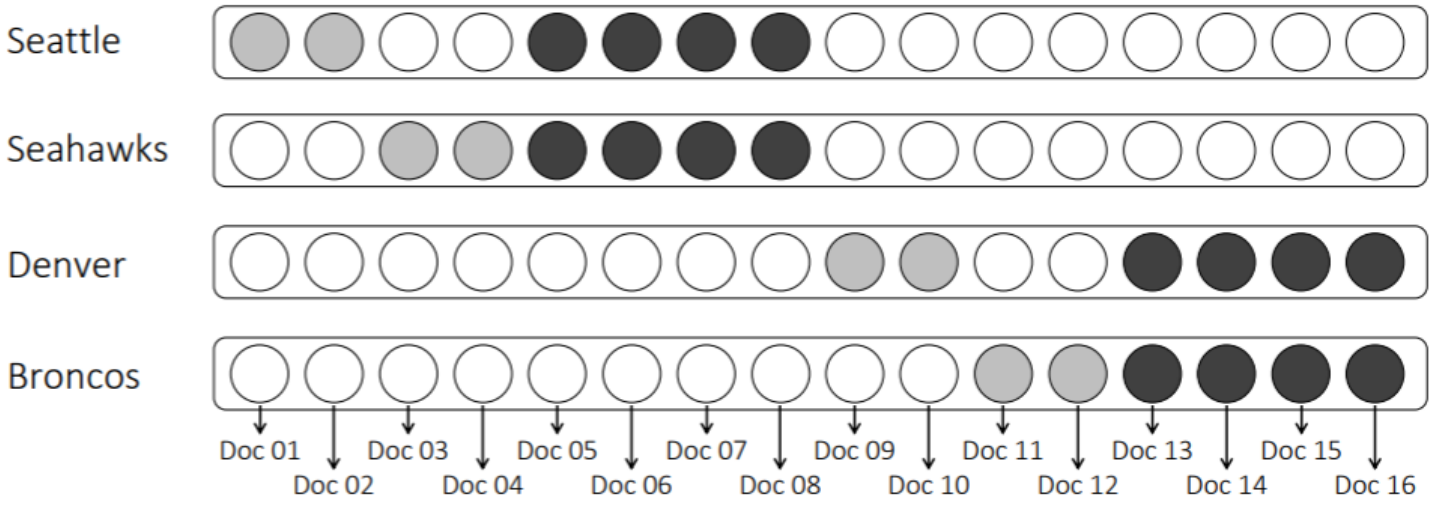
(b) "Neighbouring terms" features



(c) "Neighbouring terms w/ distances" features

NOTIONS OF SIMILARITY

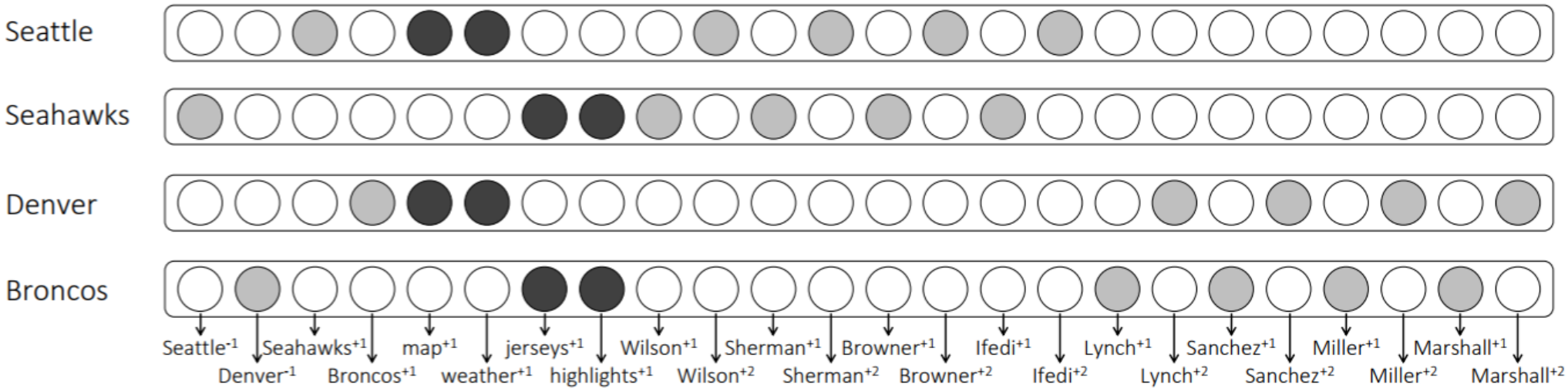
doc 01	Seattle map	doc 09	Denver map
doc 02	Seattle weather	doc 10	Denver weather
doc 03	Seahawks jerseys	doc 11	Broncos jerseys
doc 04	Seahawks highlights	doc 12	Broncos highlights
doc 05	Seattle Seahawks Wilson	doc 13	Denver Broncos Lynch
doc 06	Seattle Seahawks Sherman	doc 14	Denver Broncos Sanchez
doc 07	Seattle Seahawks Browner	doc 15	Denver Broncos Miller
doc 08	Seattle Seahawks Ifedi	doc 16	Denver Broncos Marshall



(a) "In-documents" features

NOTIONS OF SIMILARITY

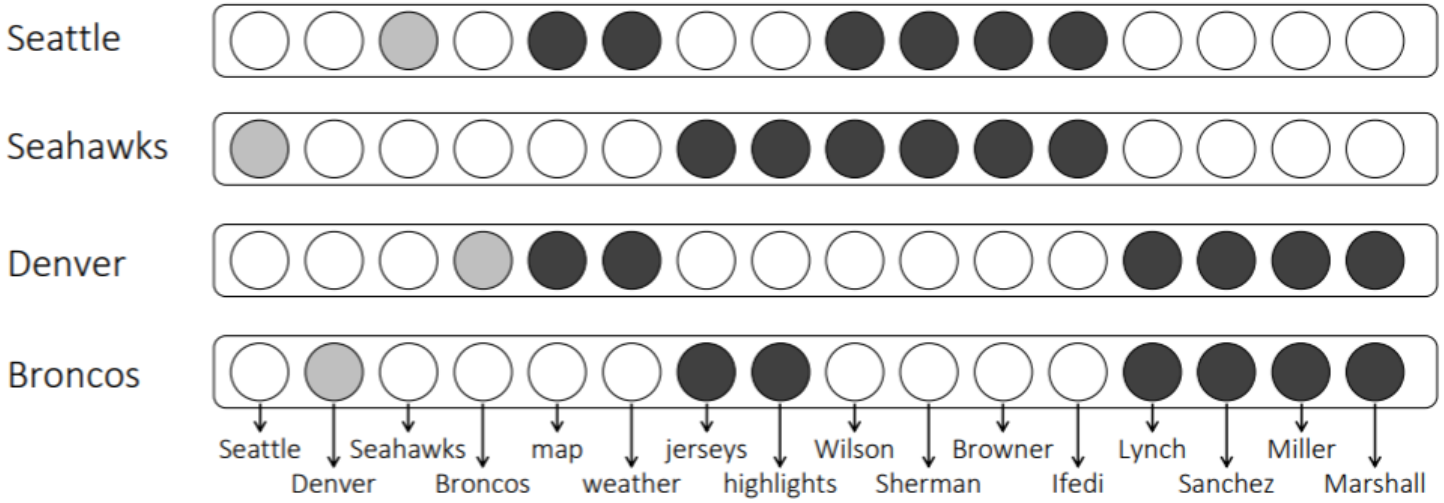
doc 01	Seattle map	doc 09	Denver map
doc 02	Seattle weather	doc 10	Denver weather
doc 03	Seahawks jerseys	doc 11	Broncos jerseys
doc 04	Seahawks highlights	doc 12	Broncos highlights
doc 05	Seattle Seahawks Wilson	doc 13	Denver Broncos Lynch
doc 06	Seattle Seahawks Sherman	doc 14	Denver Broncos Sanchez
doc 07	Seattle Seahawks Browner	doc 15	Denver Broncos Miller
doc 08	Seattle Seahawks Ifedi	doc 16	Denver Broncos Marshall



(c) "Neighbouring terms w/ distances" features

NOTIONS OF SIMILARITY

doc 01	Seattle map	doc 09	Denver map
doc 02	Seattle weather	doc 10	Denver weather
doc 03	Seahawks jerseys	doc 11	Broncos jerseys
doc 04	Seahawks highlights	doc 12	Broncos highlights
doc 05	Seattle Seahawks Wilson	doc 13	Denver Broncos Lynch
doc 06	Seattle Seahawks Sherman	doc 14	Denver Broncos Sanchez
doc 07	Seattle Seahawks Browner	doc 15	Denver Broncos Miller
doc 08	Seattle Seahawks Ifedi	doc 16	Denver Broncos Marshall



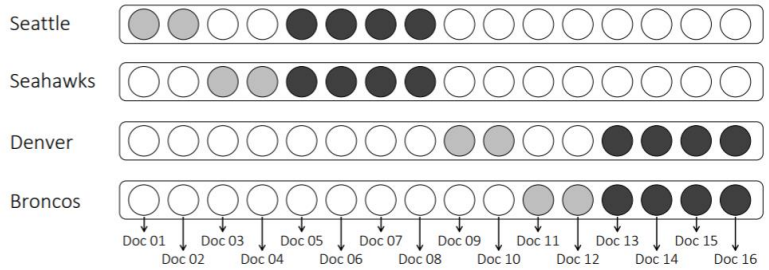
(b) "Neighbouring terms" features

NOTIONS OF SIMILARITY

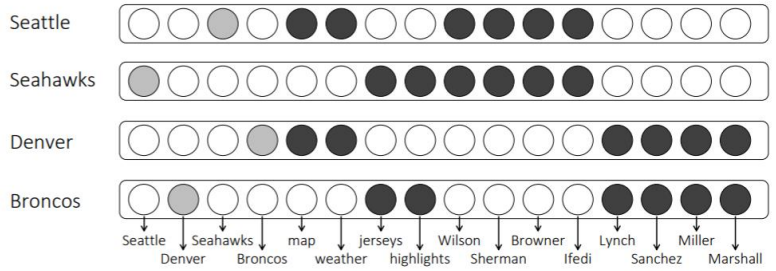
Consider the following toy corpus…

doc 01	Seattle map	doc 09	Denver map
doc 02	Seattle weather	doc 10	Denver weather
doc 03	Seahawks jerseys	doc 11	Broncos jerseys
doc 04	Seahawks highlights	doc 12	Broncos highlights
doc 05	Seattle Seahawks Wilson	doc 13	Denver Broncos Lynch
doc 06	Seattle Seahawks Sherman	doc 14	Denver Broncos Sanchez
doc 07	Seattle Seahawks Browner	doc 15	Denver Broncos Miller
doc 08	Seattle Seahawks Ifedi	doc 16	Denver Broncos Marshall

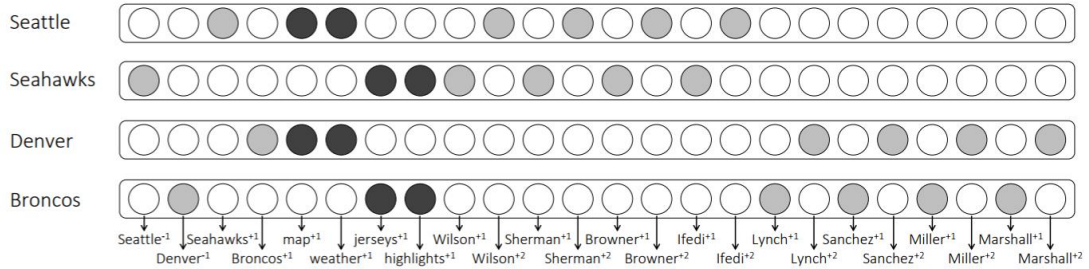
Now consider the different vector representations of terms you can derive from this corpus and how the items that are similar differ in these vector spaces



(a) "In-documents" features



(b) "Neighbouring terms" features



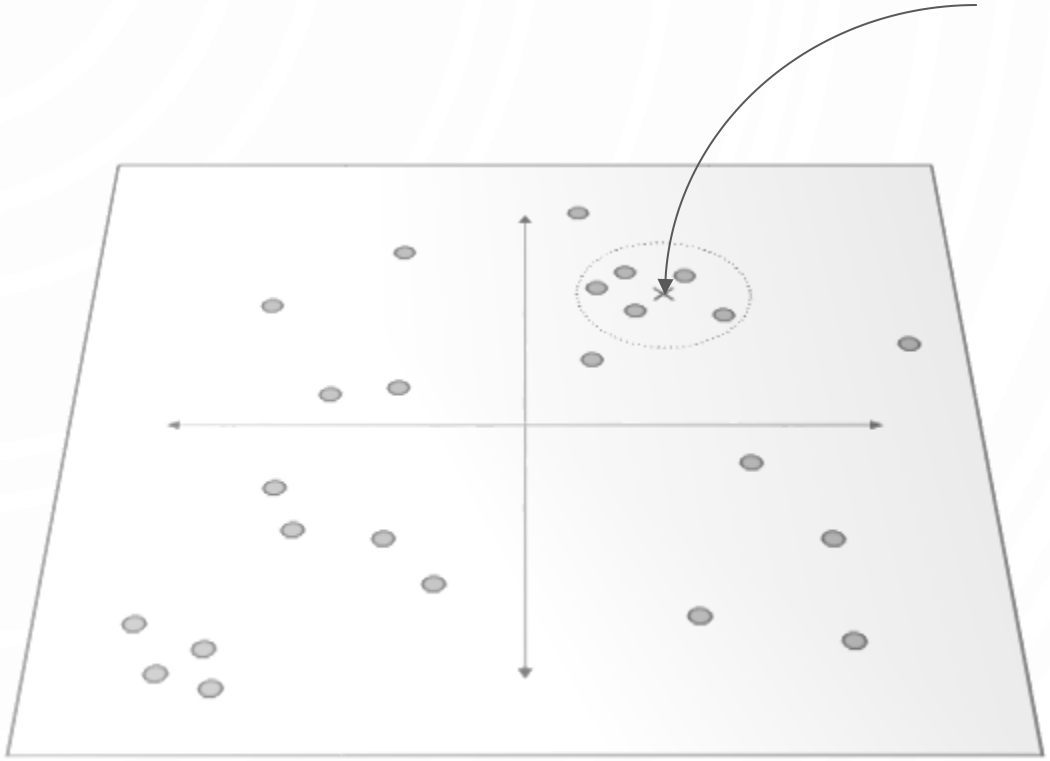
(c) "Neighbouring terms w/ distances" features

RETRIEVAL USING EMBEDDINGS

Given an input the retrieval model predicts a point in the embedding space

Items close to this point in the embedding space are retrieved as results

Relevant items should be similar to /near each other in the embedding space



SO THE DESIRED NOTION OF SIMILARITY SHOULD DEPEND ON THE TARGET TASK

DOCUMENT RANKING

cheap flights to london

✓ budget flights to london

✗ cheap flights to sydney

✗ hotels in london

QUERY AUTO-COMPLETION

cheap flights to |

✓ cheap flights to **london**

✓ cheap flights to **sydney**

✗ cheap flights to **big ben**

NEXT QUERY SUGGESTION

cheap flights to london

✓ budget flights to london

✓ hotels in london

✗ cheap flights to sydney

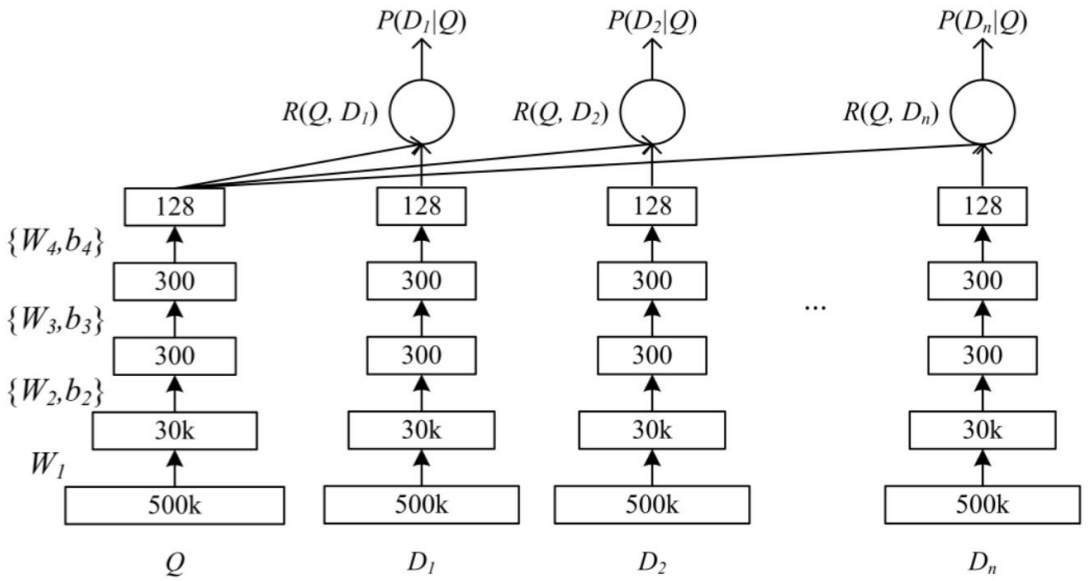
Next, we take a sample model and show how the same model captures different notions of similarity based on the data it is trained on

DEEP SEMANTIC SIMILARITY MODEL (DSSM)

Siamese network with two deep sub-models

Projects input and candidate texts into embedding space

Trained by maximizing cosine similarity between correct input-output pairs



$$\mathcal{L}_{dssm}(q, d^+, D^-) = -log\left(\frac{e^{\gamma \cdot cos\left(\vec{q},\vec{d^+}\right)}}{\sum_{d \in D} e^{\gamma \cdot cos\left(\vec{q},\vec{d}\right)}}\right)$$

$$\text{where,} \quad D = \{d^+\} \cup D^-$$

(Huang et al., 2013)
https://dl.acm.org/citation...

DSSM TRAINED ON DIFFERENT TYPES OF DATA

Trained on pairs of…	Sample training data	Useful for?	Paper
Query and document titles	<"things to do in seattle", "seattle tourist attractions">	Document ranking	(Shen et al., 2014) https://dl.acm.org/citation…
Query prefix and suffix	<"things to do in", "seattle">	Query auto-completion	(Mitra and Craswell, 2015) https://dl.acm.org/citation…
Consecutive queries in user sessions	<"things to do in seattle", "space needle">	Next query suggestion	(Mitra, 2015) https://dl.acm.org/citation…

Each model captures a different notion of similarity / regularity in the learnt embedding space

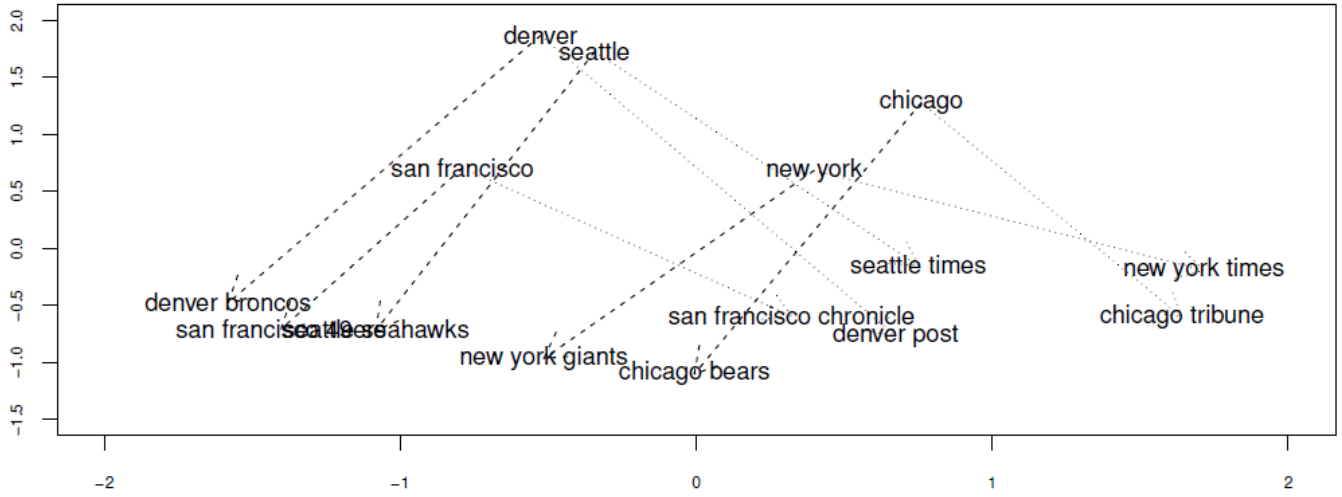
DIFFERENT REGULARITIES IN DIFFERENT EMBEDDING SPACES

Nearest neighbors for "seattle" and "taylor swift" based on two DSSM models – one trained on query-document pairs and the other trained on query prefix-suffix pairs

seattle		**taylor swift**	
Query-Document	Prefix-Suffix	Query-Document	Prefix-Suffix
weather seattle	chicago	taylor swift.com	lady gaga
seattle weather	san antonio	taylor swift lyrics	meghan trainor
seattle washington	denver	how old is taylor swift	megan trainor
ikea seattle	salt lake city	taylor swift twitter	nicki minaj
west seattle blog	seattle wa	taylor swift new song	anna kendrick

DIFFERENT REGULARITIES IN DIFFERENT EMBEDDING SPACES

The DSSM trained on session query pairs can capture regularities in the query space (similar to word2vec for terms)



Groups of similar search intent transitions from a query log

soundcloud	→	www.soundcloud.com
coasthills coop	→	www.coasthills.coop
american express	→	www.barclaycardus.com login
duke energy bill pay	→	www.duke-energy.com pay my bill
cool math games	→	www.coolmath.com
majesty shih tzu	→	what is a majesty shih tzu
hard drive dock	→	what is a hard drive dock
lugia in leaf green	→	where is lugia in leaf green
red river log jam	→	what is th red river log jam
prowl	→	what does prowl mean
rottweiler	→	rottweiler facebook
sundry	→	sundry expense
elections	→	florida governor race 2014
pleurisy	→	pleurisy shoulder pain
elections	→	2014 rowan county election results
cna classes	→	cna classes in lexington tennessee
container services inc	→	container services ringgold ga
enclosed trailers for sale	→	enclosed trailers for sale north carolina
firewood for sale	→	firewood for sale in asheboro nc
us senate race in colorado	→	us senate race in georgia
siol	→	facebook
cowboy bebop	→	facebook
mr doob	→	google
great west 100 west 29th	→	facebook
avatar dragons	→	youtube

DSSM TRAINED ON SESSION QUERY PAIRS ALLOWS FOR ANALOGIES OVER SHORT TEXT!

Query vector	Nearest neighbour
$vector(\text{“chicago”}) + vector(\text{“newspaper”})$	$vector(\text{“chicago suntimes”})$
$vector(\text{“new york”}) + vector(\text{“newspaper”})$	$vector(\text{“new york times”})$
$vector(\text{“san francisco”}) + vector(\text{“newspaper”})$	$vector(\text{“la times”})$
$vector(\text{“beyonce”}) + vector(\text{“pictures”})$	$vector(\text{“beyonce images”})$
$vector(\text{“beyonce”}) + vector(\text{“videos”})$	$vector(\text{“beyonce videos”})$
$vector(\text{“beyonce”}) + vector(\text{“net worth”})$	$vector(\text{“jaden smith net worth”})$
$vector(\text{“www.facebook.com”}) - vector(\text{“facebook”}) + vector(\text{“twitter”})$	$vector(\text{“www.twitter.com”})$
$vector(\text{“www.facebook.com”}) - vector(\text{“facebook”}) + vector(\text{“gmail”})$	$vector(\text{“www.googlemail.com”})$
$vector(\text{“www.facebook.com”}) - vector(\text{“facebook”}) + vector(\text{“hotmail”})$	$vector(\text{“www.hotmail.xom”})$
$vector(\text{“how tall is tom cruise”}) - vector(\text{“tom cruise”}) + vector(\text{“tom selleck”})$	$vector(\text{“how tall is tom selleck”})$
$vector(\text{“how old is gwen stefani”}) - vector(\text{“gwen stefani”}) + vector(\text{“meghan trainor”})$	$vector(\text{“how old is meghan trainor”})$
$vector(\text{“how old is gwen stefani”}) - vector(\text{“gwen stefani”}) + vector(\text{“ariana grande”})$	$vector(\text{“how old is ariana grande 2014”})$
$vector(\text{“university of washington”}) - vector(\text{“seattle”}) + vector(\text{“chicago”})$	$vector(\text{“chicago state university”})$
$vector(\text{“university of washington”}) - vector(\text{“seattle”}) + vector(\text{“denver”})$	$vector(\text{“university of colorado”})$
$vector(\text{“university of washington”}) - vector(\text{“seattle”}) + vector(\text{“detroit”})$	$vector(\text{“northern illinois university”})$

WHEN IS IT PARTICULARLY IMPORTANT TO THINK ABOUT NOTIONS OF SIMILARITY?

If you are using pre-trained embeddings, instead of learning the text representations in an end-to-end model for the target task

USING PRE-TRAINED WORD EMBEDDINGS FOR DOCUMENT RANKING

Non-matching terms "population" and "area" indicate first passage is more relevant to the query "Albuquerque"

Use word2vec embeddings to compare every query and document terms

$$sim(q,d) = cos(\vec{v}_q, \vec{v}_d) = \frac{\vec{v}_q^{\intercal}\vec{v}_d}{\|\vec{v}_q\|\|\vec{v}_d\|}$$

$$\text{where,} \quad \vec{v}_q = \frac{1}{|q|}\sum_{t_q \in q}\frac{\vec{v}_{t_q}}{\|\vec{v}_{t_q}\|}$$

$$\vec{v}_d = \frac{1}{|d|}\sum_{t_d \in d}\frac{\vec{v}_{t_d}}{\|\vec{v}_{t_d}\|}$$

Albuquerque is the most populous city in the U.S. state of New Mexico. The high-altitude city serves as the county seat of Bernalillo County, and it is situated in the central part of the state, straddling the Rio Grande. The city population is 557,169 as of the July 1, 2014, population estimate from the United States Census Bureau, and ranks as the 32nd-largest city in the U.S. The Metropolitan Statistical Area (or MSA) has a population of 902,797 according to the United States Census Bureau's most recently available estimate for July 1, 2013.

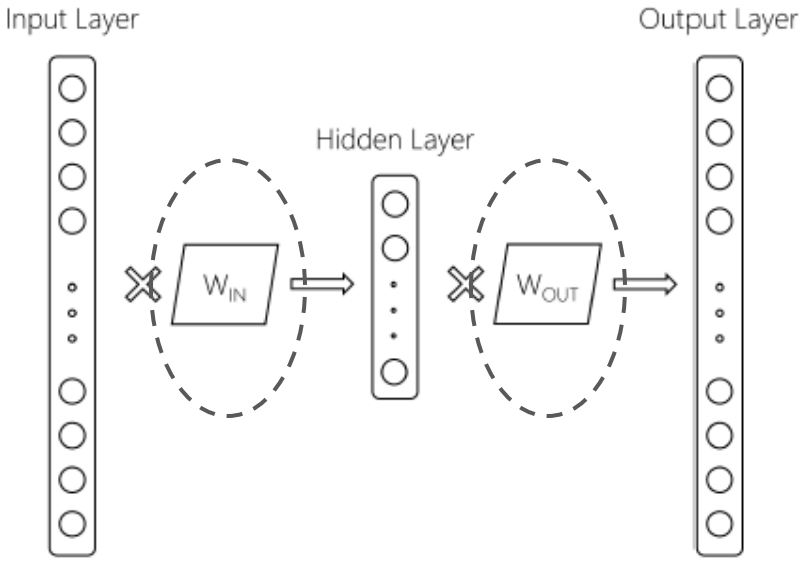
Passage *about* Albuquerque

Allen suggested that they could program a BASIC interpreter for the device; after a call from Gates claiming to have a working interpreter, MITS requested a demonstration. Since they didn't actually have one, Allen worked on a simulator for the Altair while Gates developed the interpreter. Although they developed the interpreter on a simulator and not the actual device, the interpreter worked flawlessly when they demonstrated the interpreter to MITS in Albuquerque, New Mexico in March 1975; MITS agreed to distribute it, marketing it as Altair BASIC.

Passage *not about* Albuquerque

DUAL EMBEDDING SPACE MODEL (DESM)

…but what if I told you that everyone using word2vec is throwing half the model away?



IN-OUT similarity captures a more Topical notion of term-term relationship compared to IN-IN and OUT-OUT

yale			seahawks		
IN-IN	OUT-OUT	IN-OUT	IN-IN	OUT-OUT	IN-OUT
yale	yale	yale	seahawks	seahawks	seahawks
harvard	uconn	faculty	49ers	broncos	highlights
nyu	harvard	aumni	broncos	49ers	jerseys
cornell	tulane	orientation	packers	nfl	tshirts
tulane	nyu	haven	nfl	packers	seattle
tufts	tufts	graduate	steelers	steelers	hats

Better to represent query terms using IN embeddings and document terms using OUT embeddings

(Mitra et al., 2016)
https://arxiv.org/abs/1602.01137

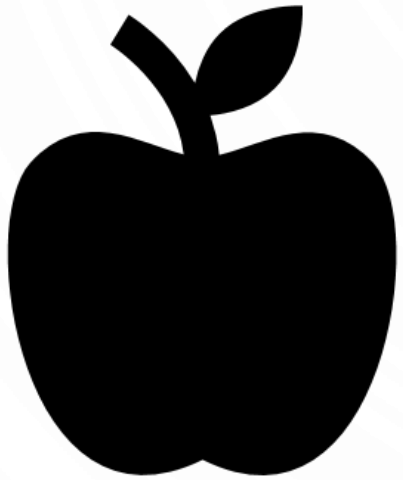
GET THE DATA

Download

IN+OUT Embeddings for 2.7M words trained on 600M+ Bing queries

https://www.microsoft.com/en-us/download/details.aspx?id=52597

BUT TERMS ARE INHERENTLY AMBIGUOUS



or

TOPIC-SPECIFIC TERM REPRESENTATIONS

Terms can take different meanings in different context – global representations likely to be coarse and inappropriate under certain topics

Global model likely to focus more on learning accurate representations of popular terms

Often impractical to train on full corpus – without topic specific sampling important training instances may be ignored

global	local
cutting	tax
squeeze	deficit
reduce	vote
slash	budget
reduction	reduction
spend	house
lower	bill
halve	plan
soften	spend
freeze	billion

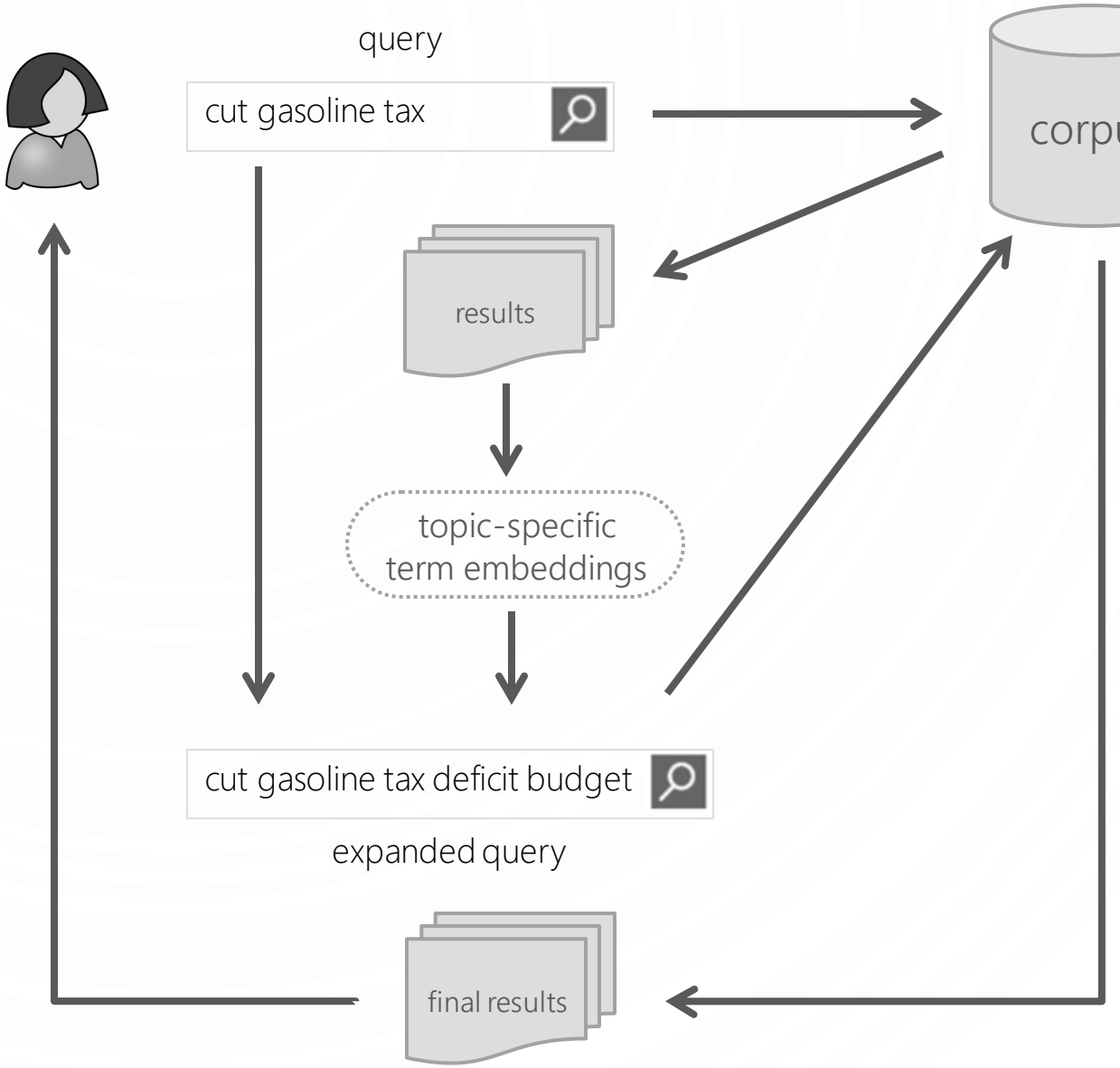
Figure 3: Terms similar to 'cut' for a word2vec model trained on a general news corpus and another trained only on documents related to 'gasoline tax'.

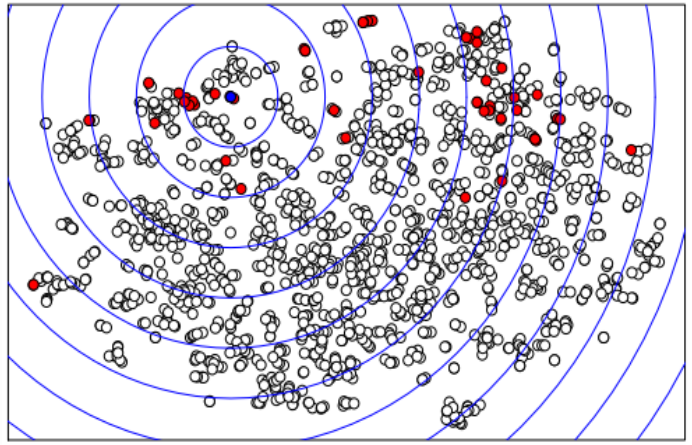
TOPIC-SPECIFIC TERM EMBEDDINGS FOR QUERY EXPANSION

Use documents from first round of retrieval to learn a query-specific embedding space

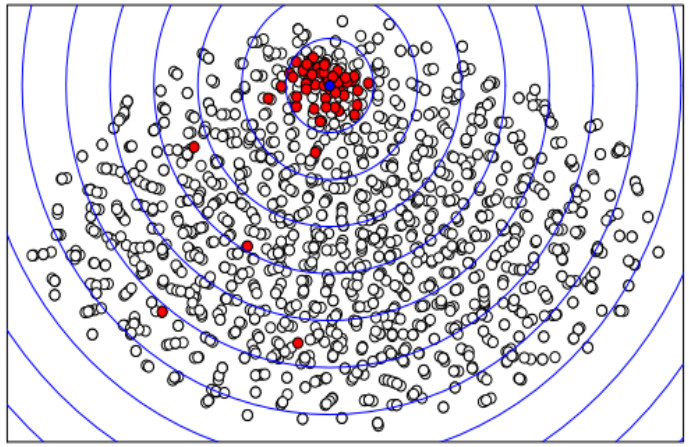
Use learnt embeddings to find related terms for query expansion for second round of retrieval

(Diaz et al., 2015)
http://anthology.aclweb.org/...



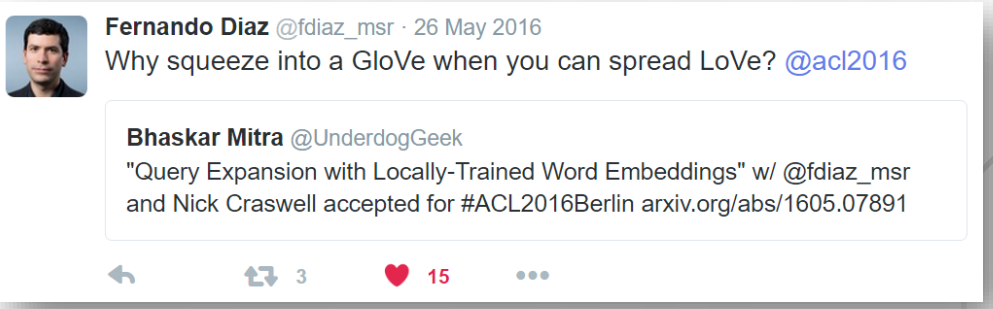


global



local

Figure 5: Global versus local embedding of highly relevant terms. Each point represents a candidate expansion term. Red points have high frequency in the relevant set of documents. White points have low or no frequency in the relevant set of documents. The blue point represents the query. Contours indicate distance from the query.



Microsoft

UCL

Now, let's talk about deep neural network models for document ranking…

CHALLENGES IN SHORT VS. LONG TEXT RETRIEVAL

Short-text

Vocabulary mismatch more serious problem

Long-text

Documents contain mixture of many topics

Matches in different parts of a long document contribute unequally

Term proximity is an important consideration

A TALE OF TWO QUERIES

"PEKAROVIC LAND COMPANY"

Hard to learn good representation for rare term *pekarovic*

But easy to estimate relevance based on patterns of exact matches

Proposal: Learn a neural model to estimate relevance from patterns of exact matches

"WHAT CHANNEL SEAHAWKS ON TODAY"

Target document likely contains *ESPN* or *sky sports* instead of *channel*

An embedding model can associate *ESPN* in document to *channel* in query

Proposal: Learn embeddings of text and match query with document in the embedding space

The Duet Architecture

Use neural networks to model both functions and learn their parameters jointly

THE DUET ARCHITECTURE

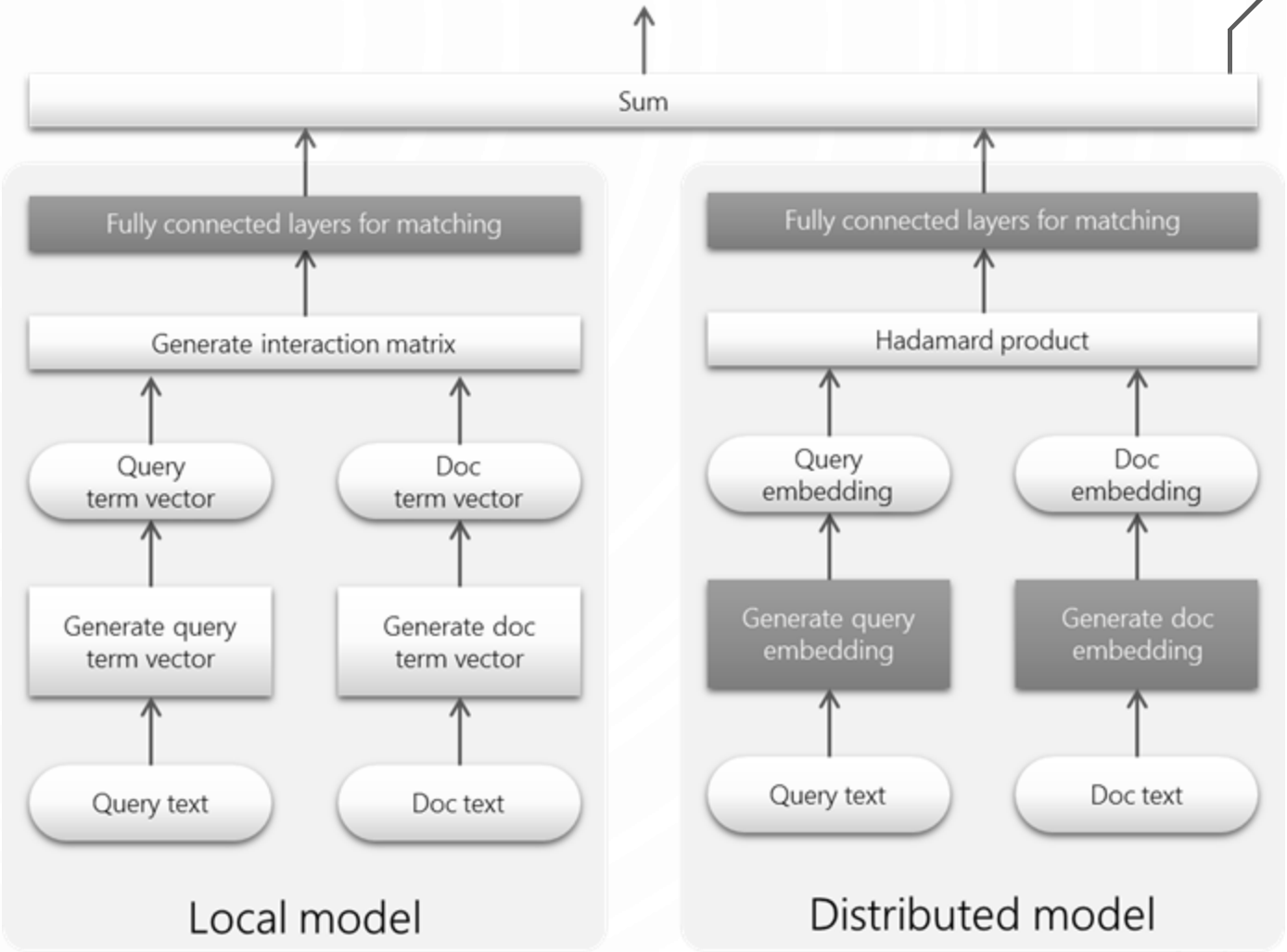
Linear combination of two models trained jointly on labelled query-document pairs

$$f(\mathbf{Q}, \mathbf{D}) = f_\ell(\mathbf{Q}, \mathbf{D}) + f_d(\mathbf{Q}, \mathbf{D})$$

Local model operates on lexical interaction matrix, and Distributed model projects text into an embedding space for matching

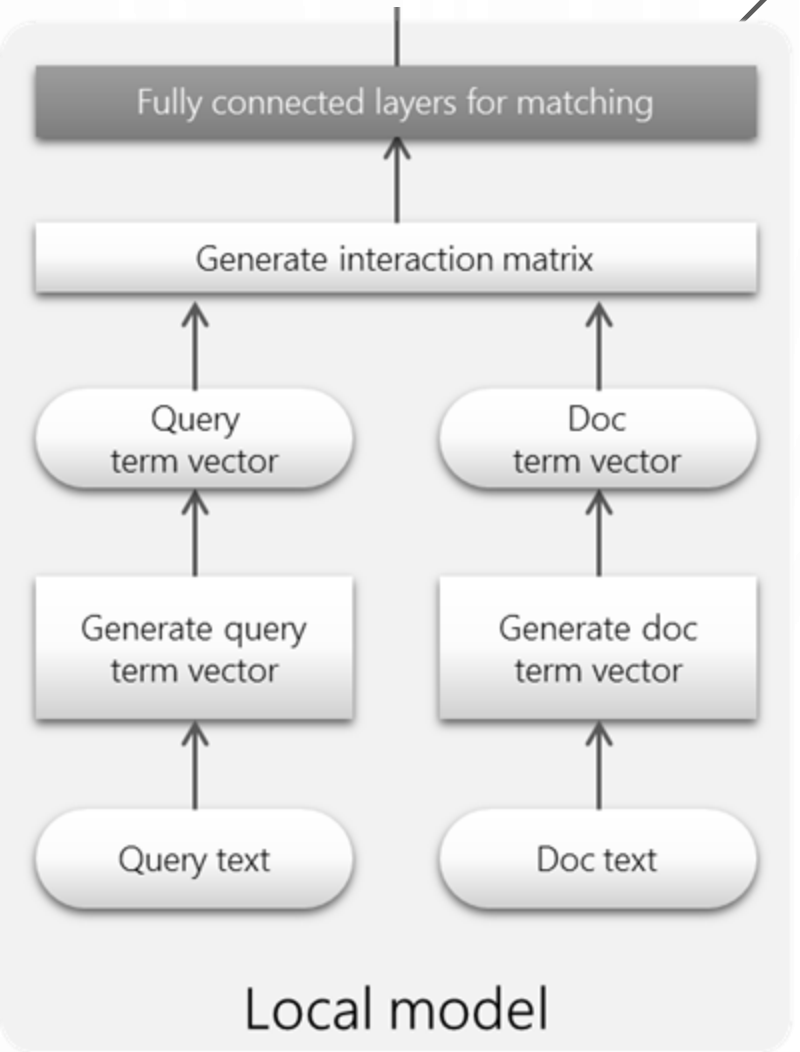
(Mitra et al., 2017)
http://anthology.aclweb.org/...

(Nanni et al., 2017)
https://dl.acm.org/citation...

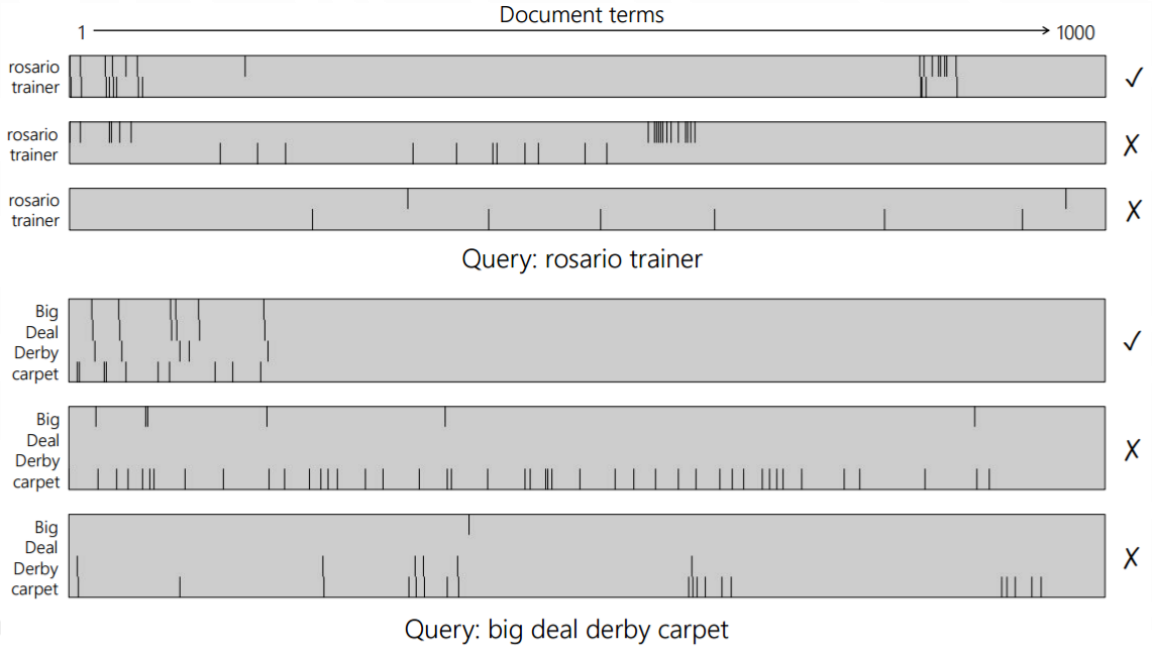


LOCAL SUB-MODEL

Focuses on patterns of exact matches of query terms in document



INTERACTION MATRIX OF QUERY AND DOCUMENT TERMS



$$X_{i,j} = \begin{cases} 1, if\ q_i = d_j \\ 0, otherwise \end{cases}$$

In relevant documents,

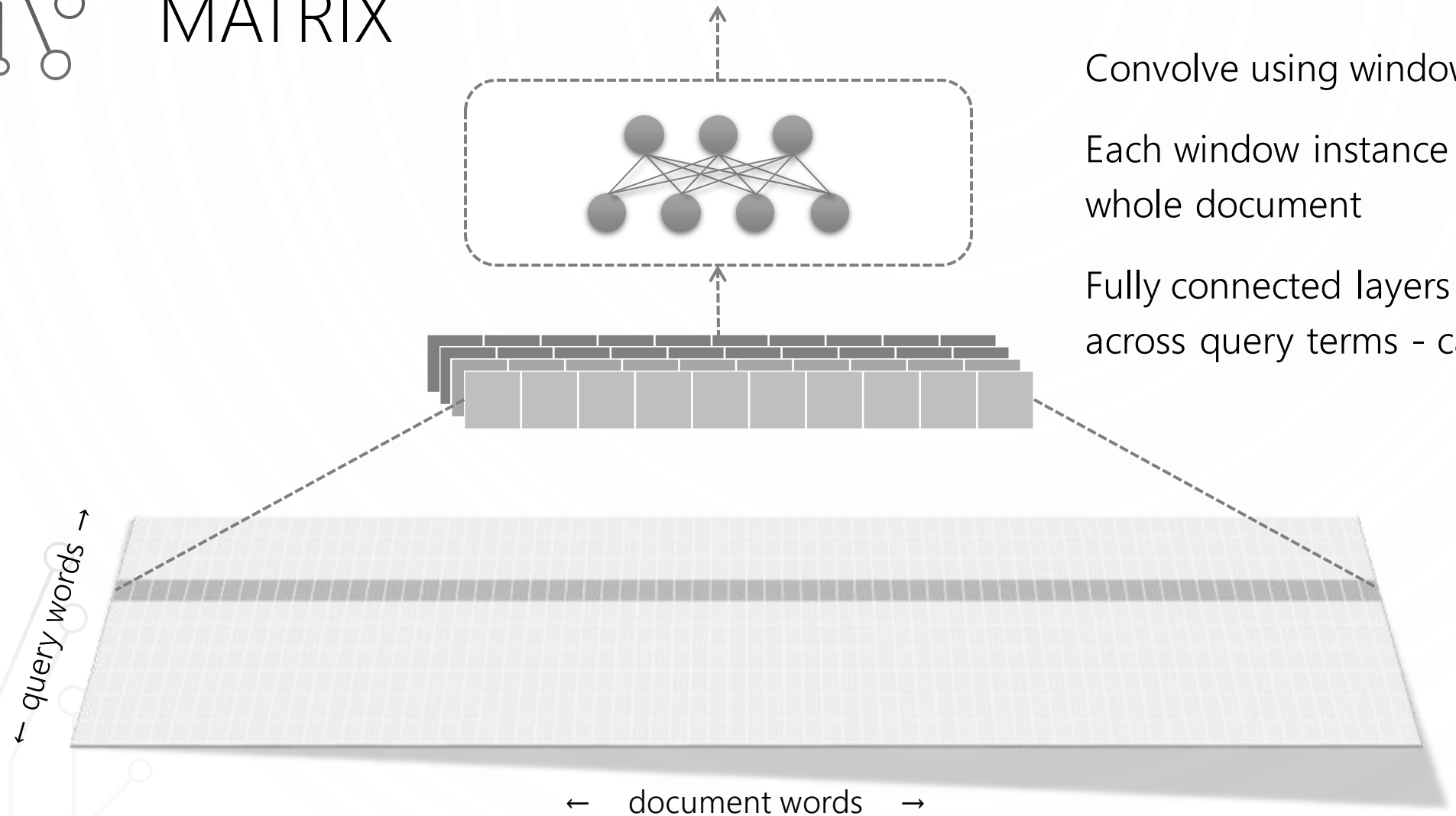
→Many matches, typically in clusters

→Matches localized early in document

→Matches for all query terms

→In-order (phrasal) matches

ESTIMATING RELEVANCE FROM INTERACTION MATRIX



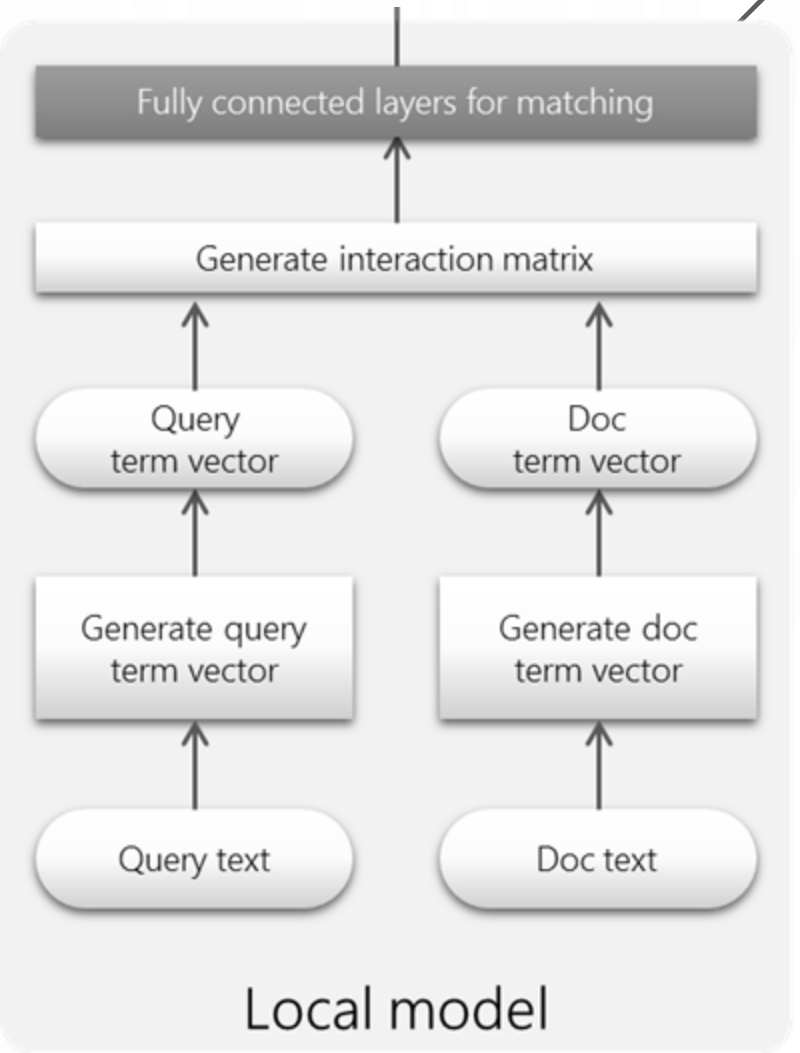
Convolve using window of size $n_d \times 1$

Each window instance compares a query term w/ whole document

Fully connected layers aggregate evidence across query terms - can model phrasal matches

LOCAL SUB-MODEL

Focuses on patterns of exact matches of query terms in document



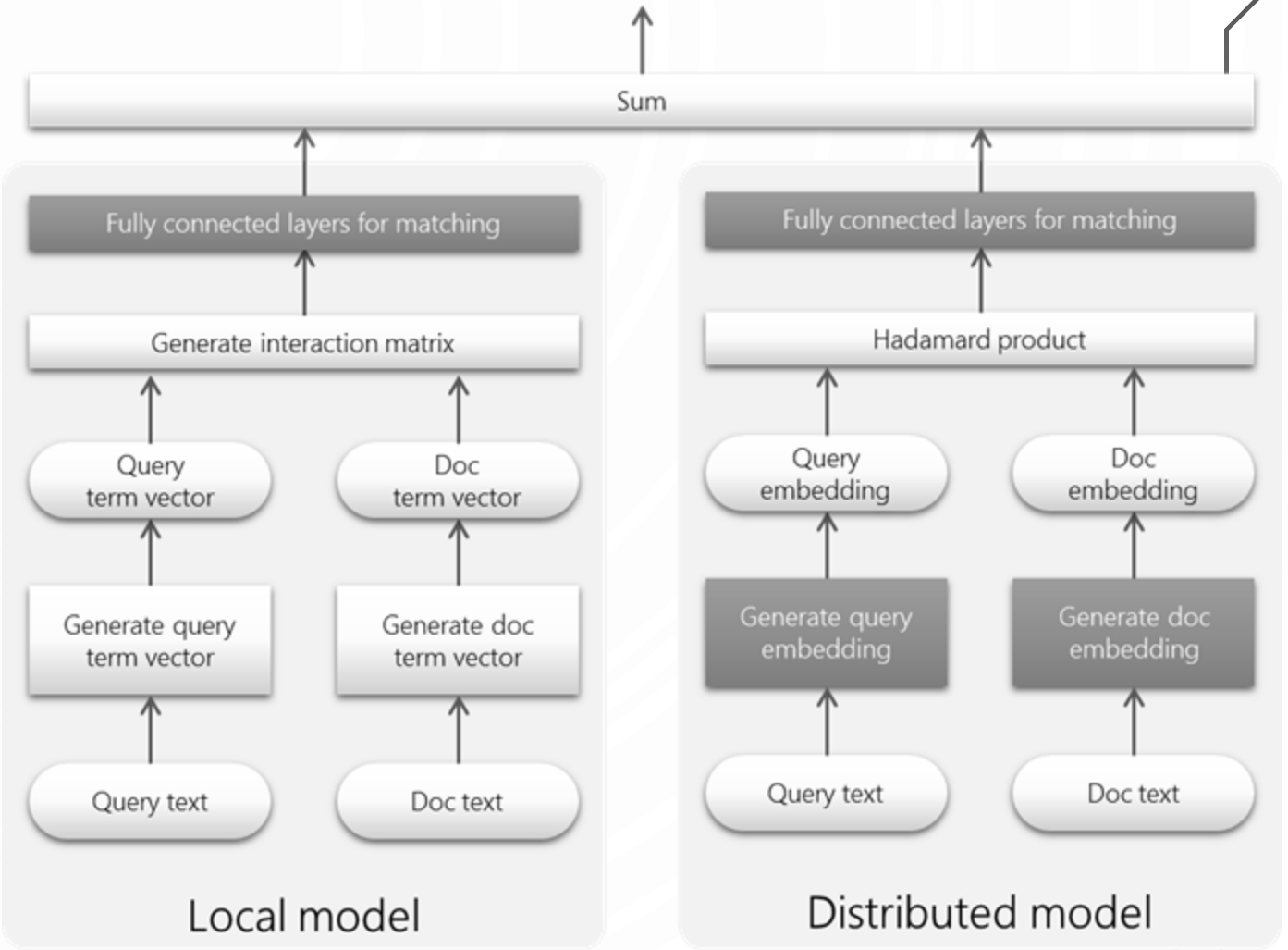
THE DUET ARCHITECTURE

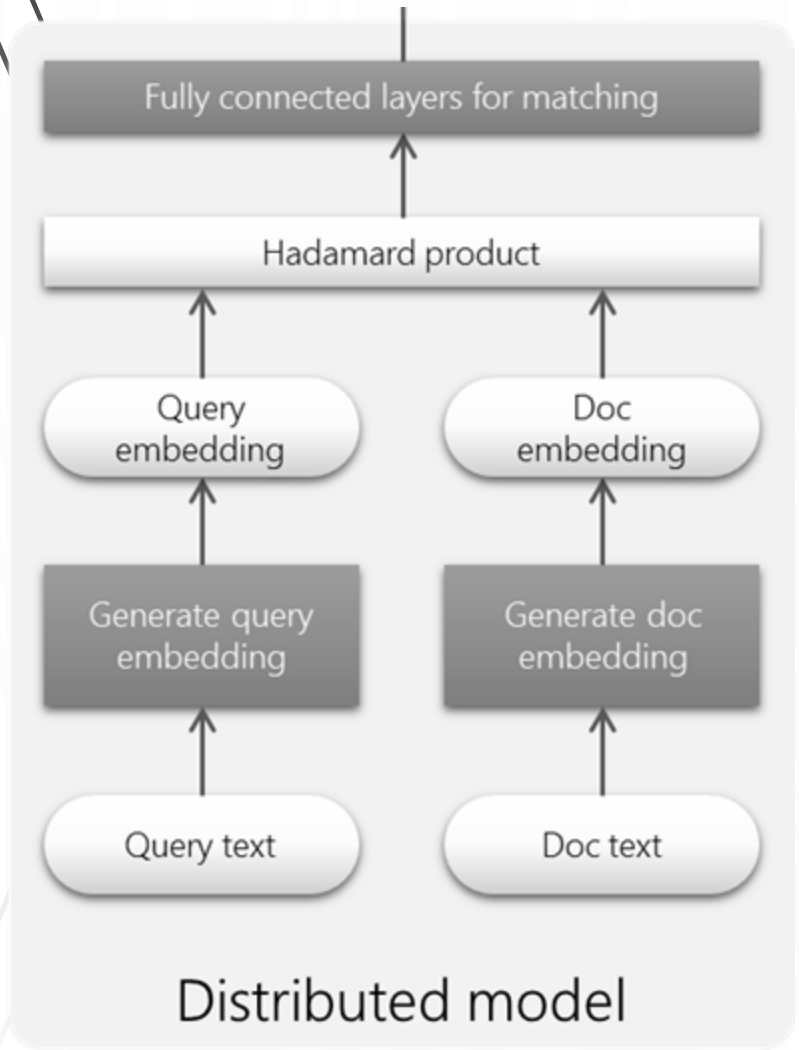
Linear combination of two models trained jointly on labelled query-document pairs

$$f(\mathbf{Q}, \mathbf{D}) = f_\ell(\mathbf{Q}, \mathbf{D}) + f_d(\mathbf{Q}, \mathbf{D})$$

Local model operates on lexical interaction matrix

Distributed model projects *n*-graph vectors of text into an embedding space and then estimates match

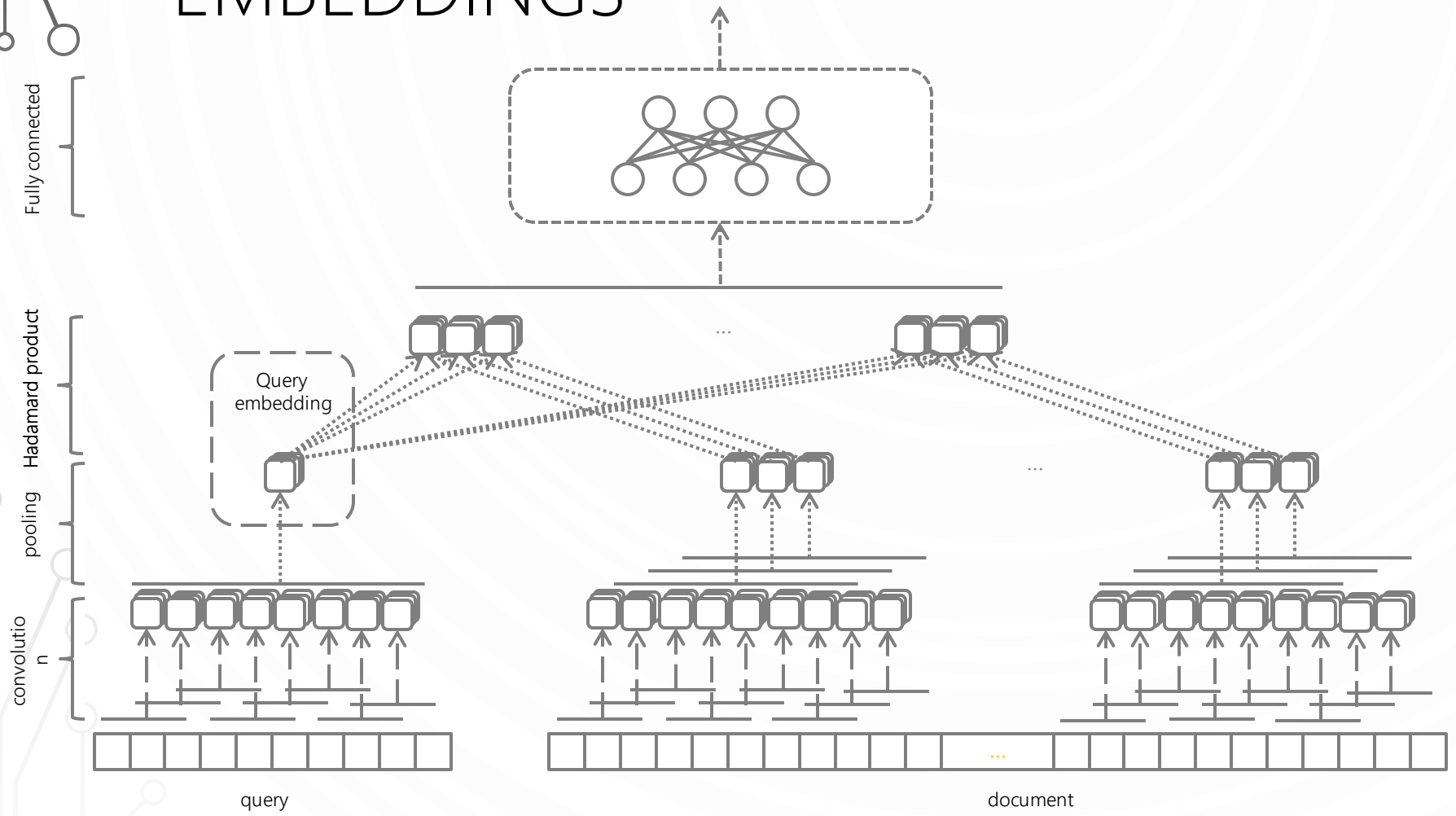




DISTRIBUTED SUB-MODEL

Learns representation of text and matches query with document in the embedding space

ESTIMATING RELEVANCE FROM TEXT EMBEDDINGS



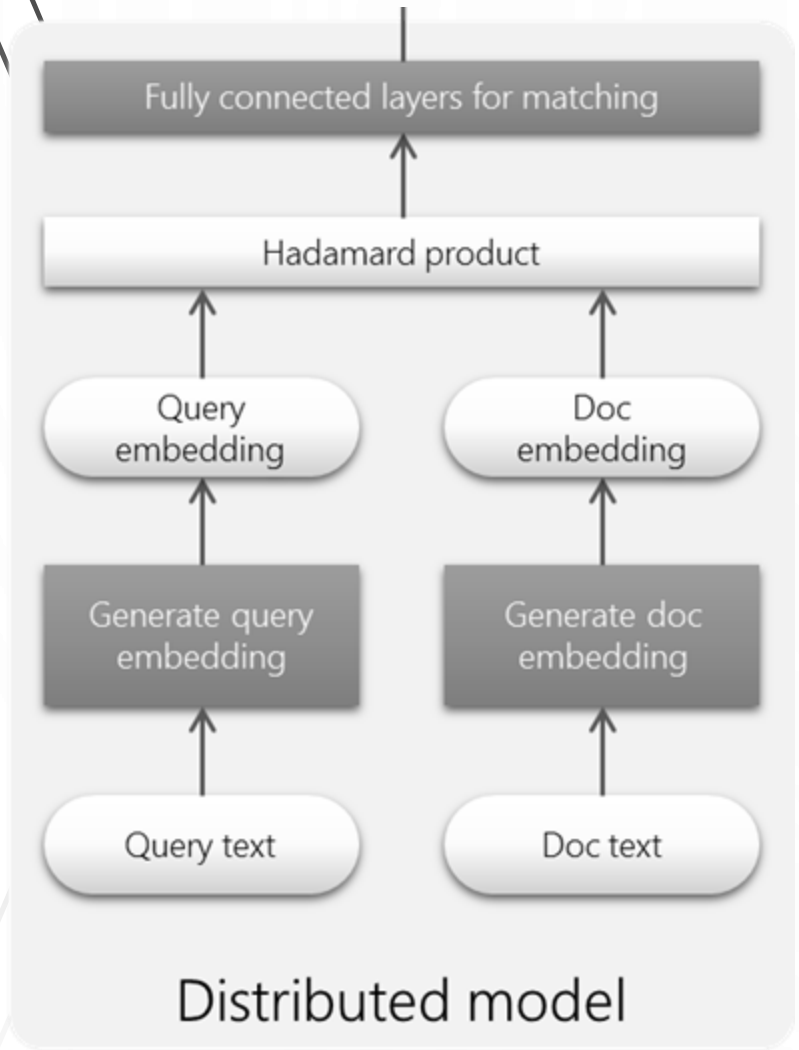
Convolve over query and document terms

Match query with moving windows over document

Learn text embeddings specifically for the task

Matching happens in embedding space

* Network architecture slightly simplified for visualization – refer paper for exact details



DISTRIBUTED SUB-MODEL

Learns representation of text and matches query with document in the embedding space

THE DUET MODEL

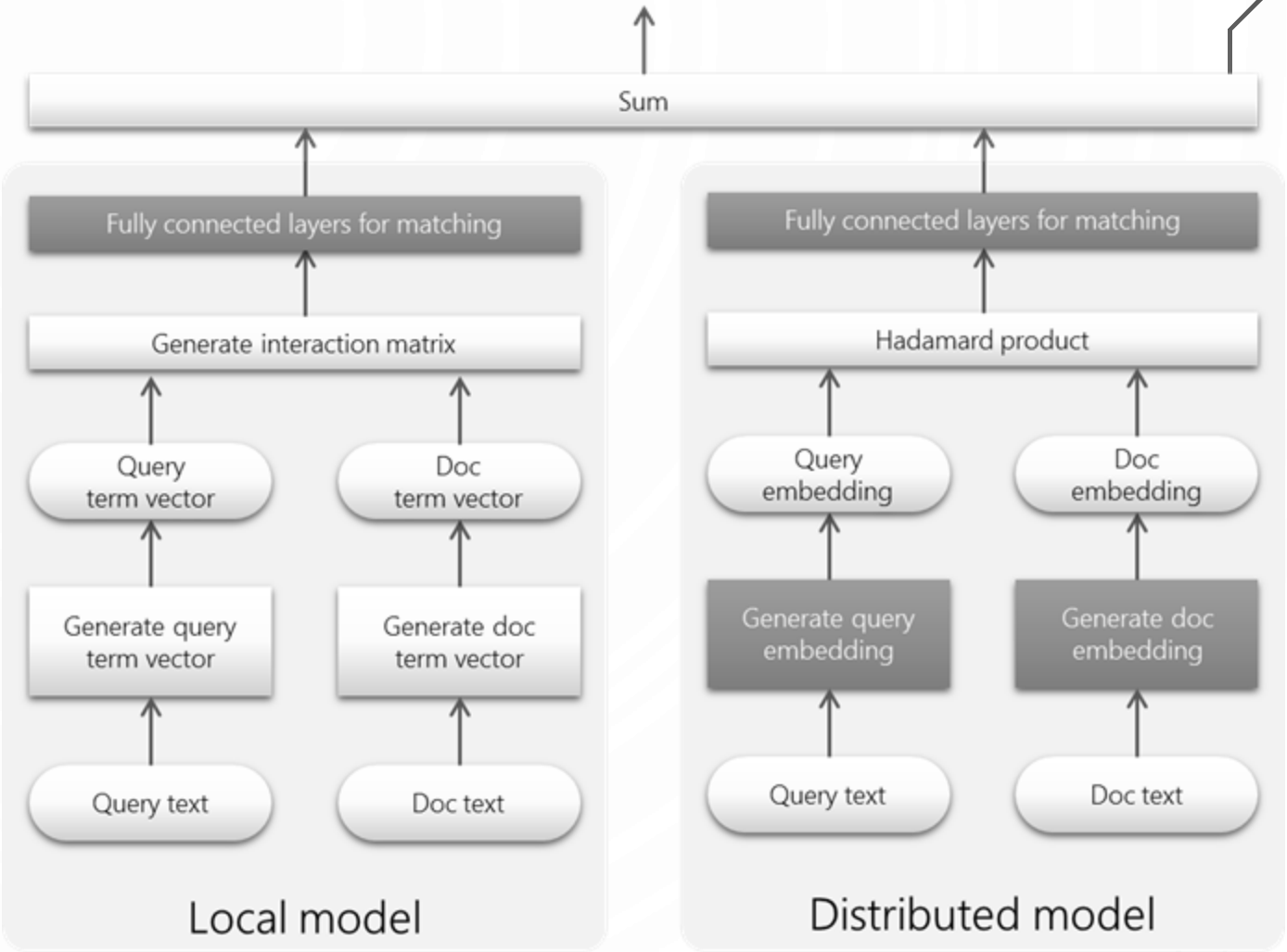
Training sample: $Q, D^+, D_1^- \; D_2^- \; D_3^- \; D_4^-$

$D^+ = Document\ rated\ Excellent\ or\ Good$
$D^- = Document\ 2\ ratings\ worse\ than\ D^+$

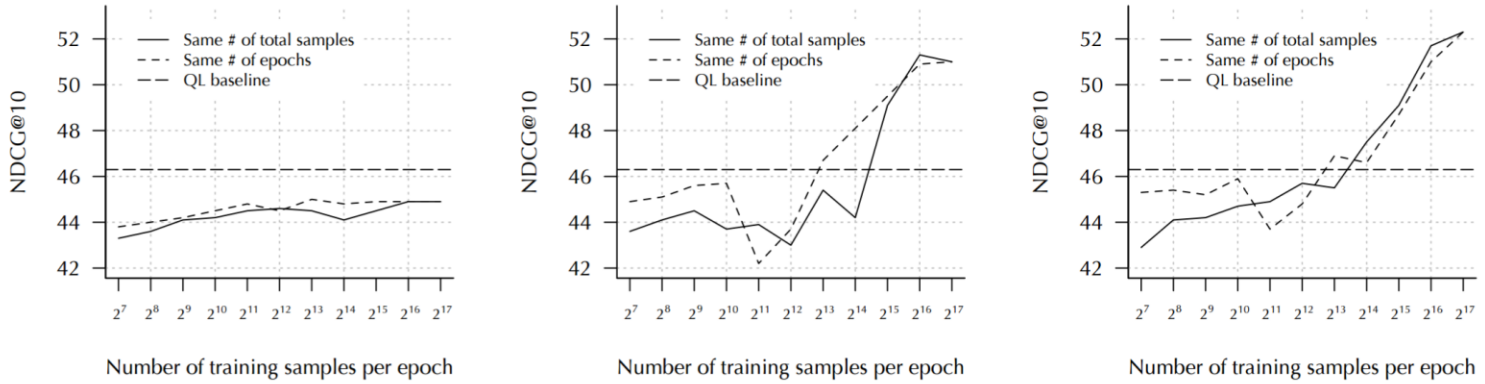
$$p(\mathbf{D}^*|\mathbf{Q}) = \frac{\exp(f(\mathbf{Q}, \mathbf{D}^*))}{\sum_{\mathbf{D} \in \mathcal{N}} \exp(f(\mathbf{Q}, \mathbf{D}))}$$

Optimize cross-entropy loss

Implemented using CNTK (GitHub link)



EFFECT OF TRAINING DATA VOLUME



(a) Local model (b) Distributed model (c) Duet model

Figure 7: We study the performance of our model variants when trained with different size datasets. For every, dataset size we train two models – one for exactly one epoch and another one with multiple epochs such that the total number of training samples seen by the model during training is 131,072.

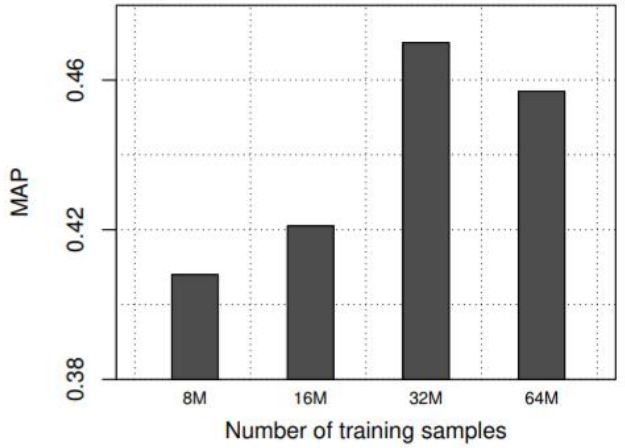


Figure 1: Effect of training data size on the performance of the Duet model. Training on four folds, reporting MAP on holdout fold.

Key finding: large quantity of training data necessary for learning good representations, less impactful for training local model

If we classify models by query level performance there is a clear clustering of lexical (local) and semantic (distributed) models

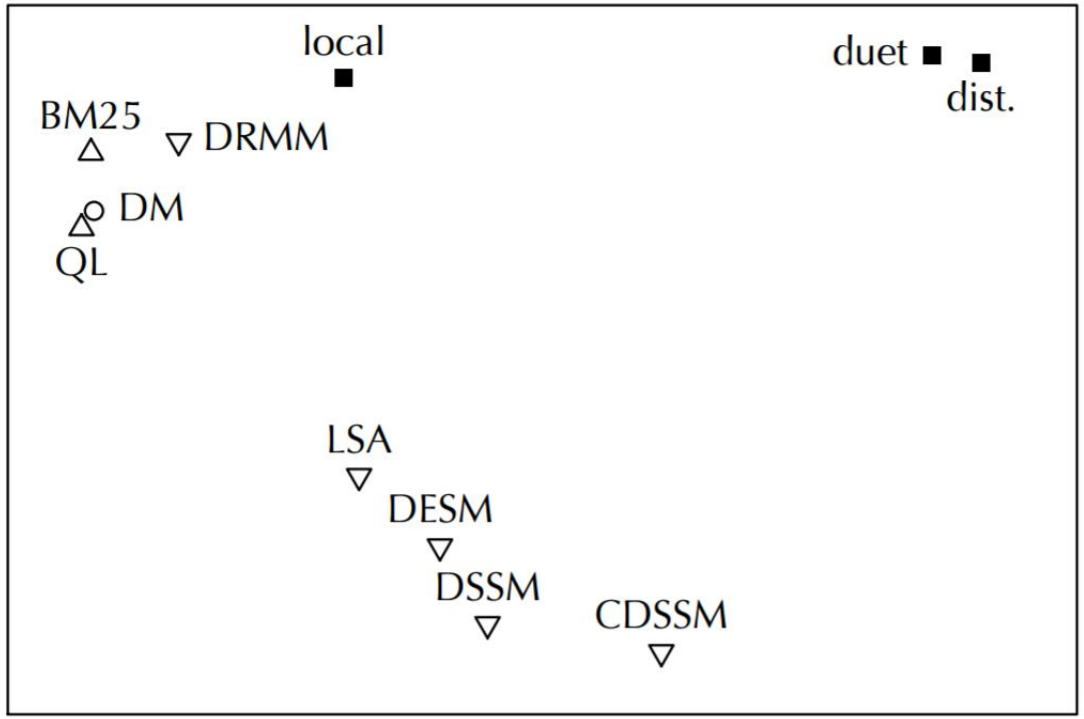


Figure 6: Principal component analysis of models based on retrieval performance across testing queries. Models using exact term matches (△), proximity (○), and inexact matches (▽) are presented. Our models are presented as black squares.

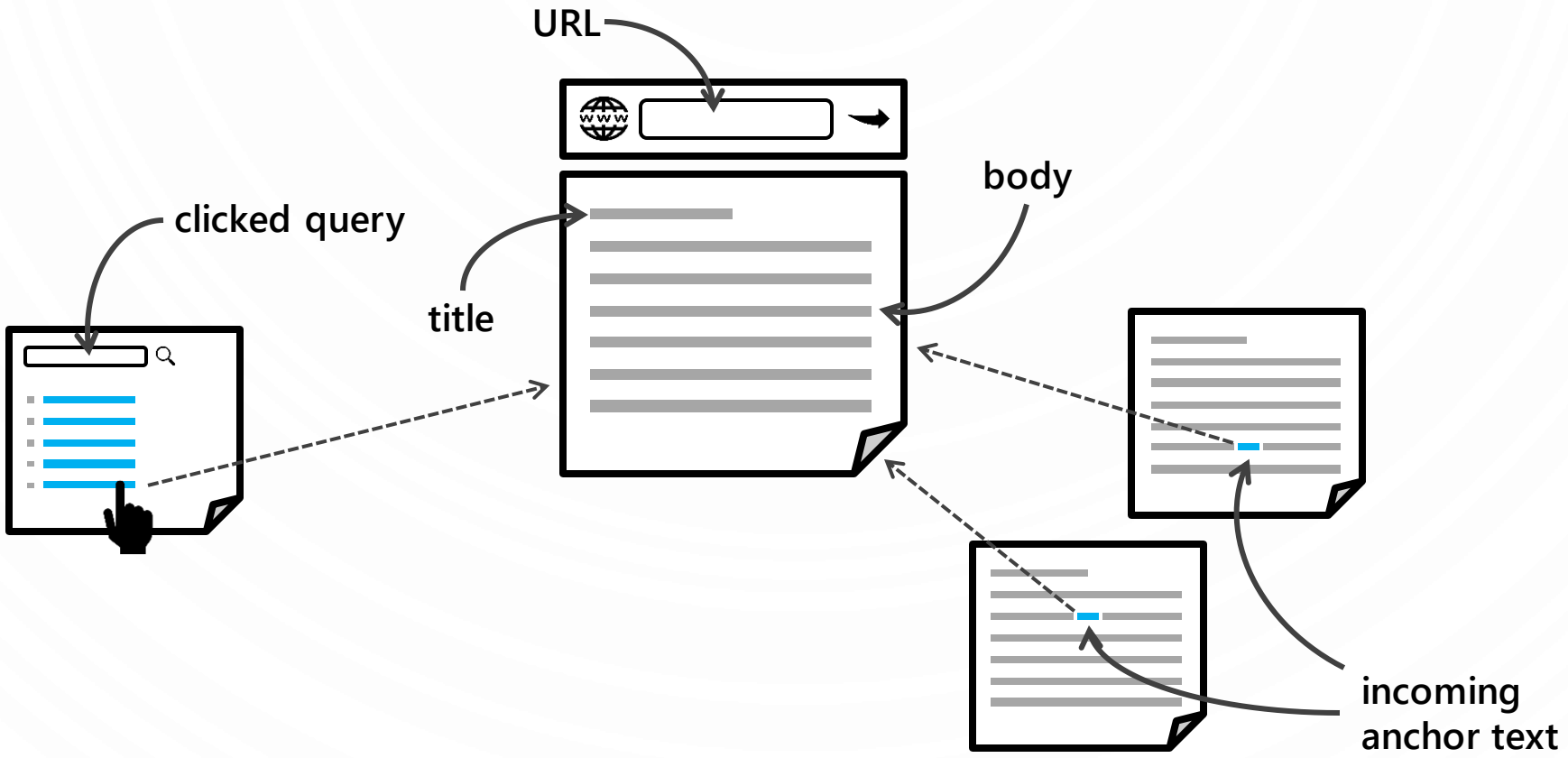
GET THE CODE

Download

Implemented using CNTK python API

https://github.com/bmitra-msft/NDRM/blob/master/notebooks/Duet.ipynb

BUT WEB DOCUMENTS ARE MORE THAN JUST BODY TEXT…



Microsoft

UCL

EXTENDING NEURAL RANKING MODELS TO MULTIPLE DOCUMENT FIELDS

BM25 → BM25F

Neural ranking model → ?

RANKING DOCUMENTS WITH MULTIPLE FIELDS

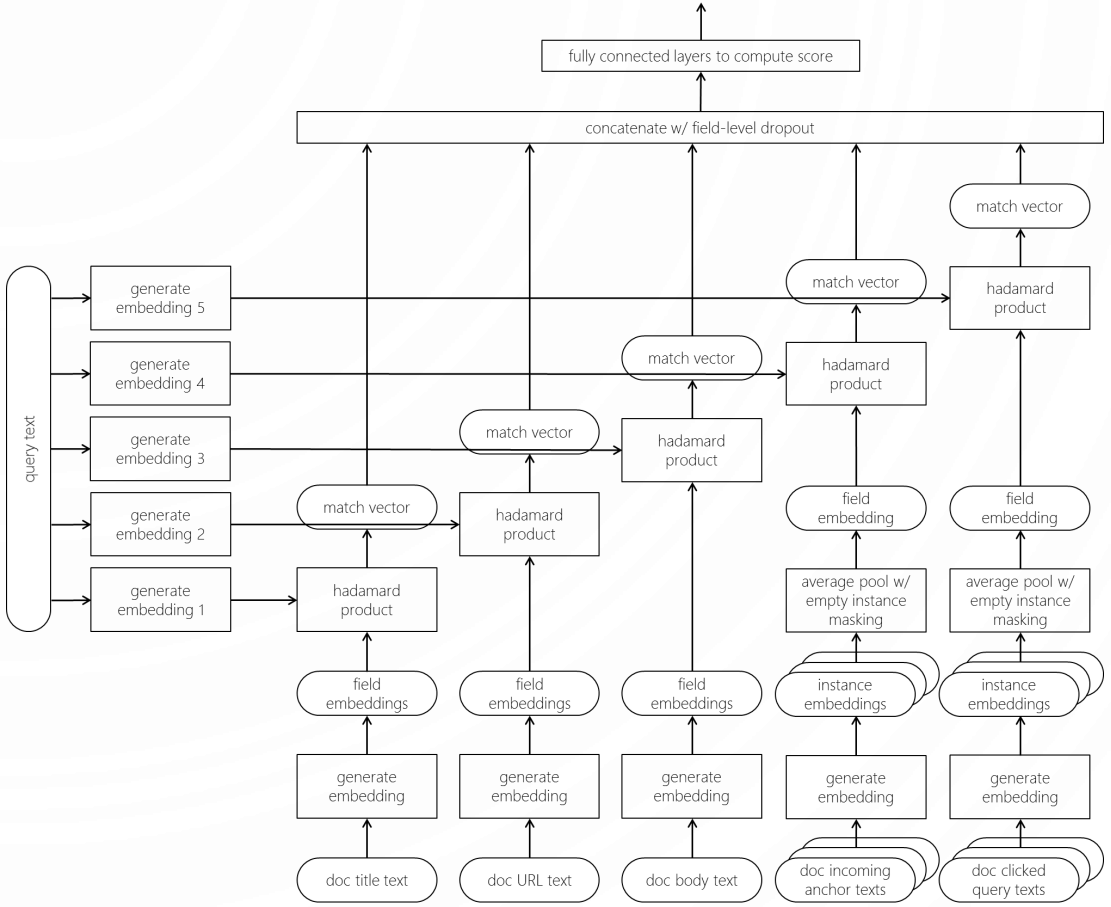
Document consists of multiple text fields (e.g., title, URL, body, incoming anchors, and clicked queries)

Fields, such as incoming anchors and clicked queries, contains variable number of short text instances

Body field has long text, whereas clicked queries are typically only few words in length

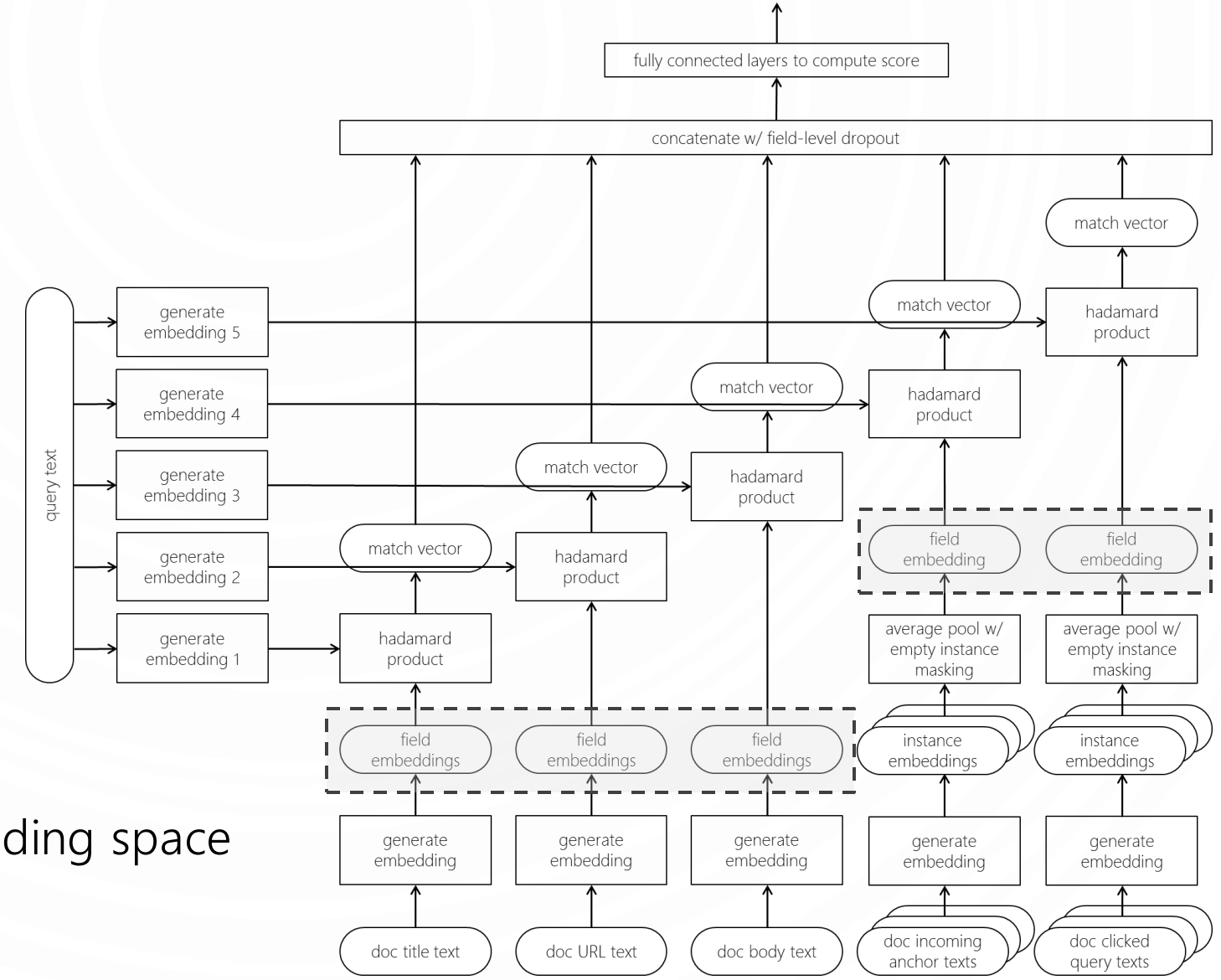
URL field contains non-natural language text

(Zamani et al., 2018)
Pre-print coming soon…



Microsoft

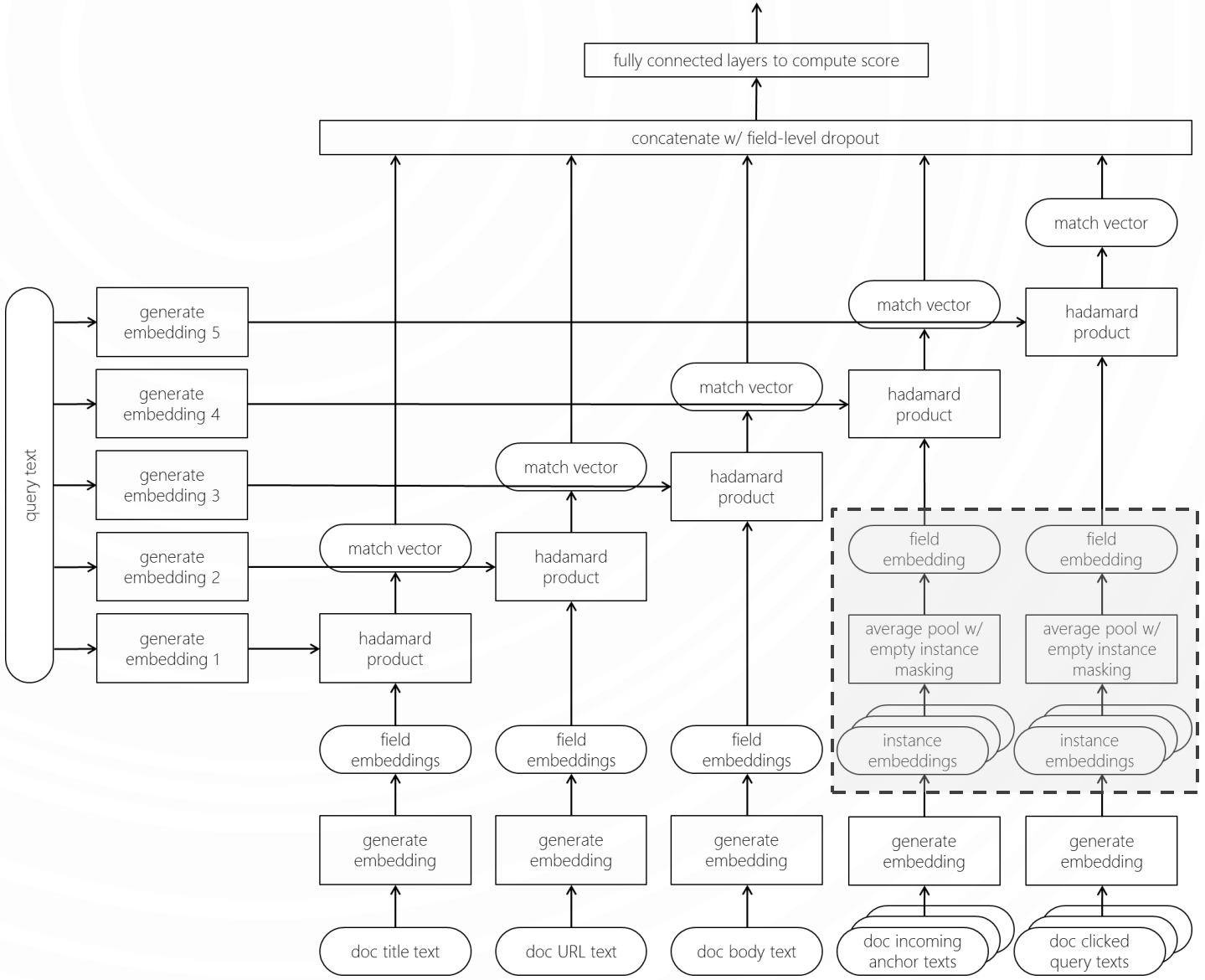
UCL



Learn a different embedding space for each document field

For multiple-instance fields, average pool the instance level embeddings

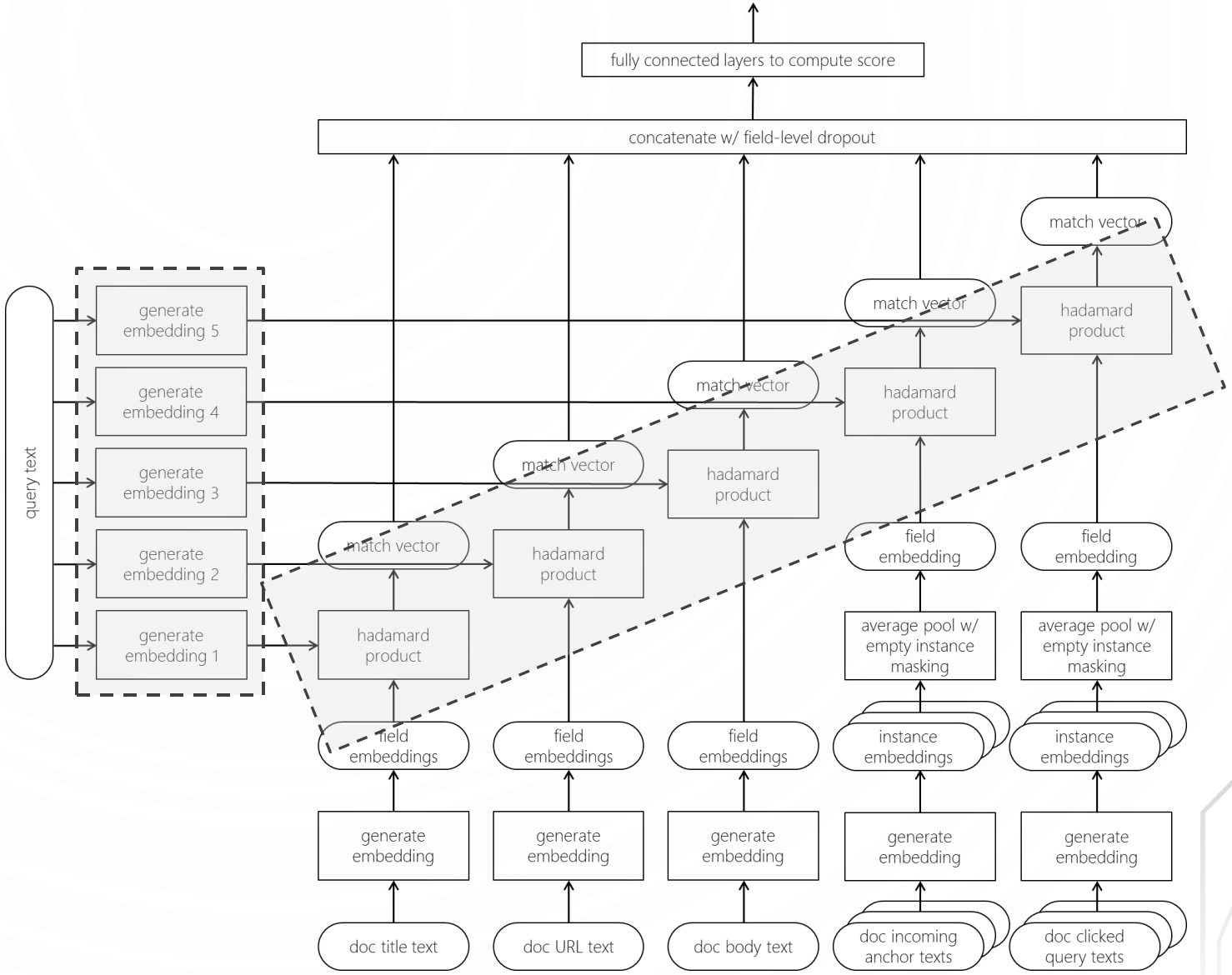
Mask empty text instances, and average only among non-empty instances to avoid preferring documents with more instances



Learn different query embeddings for matching against different fields

Different fields may match different aspects of the query

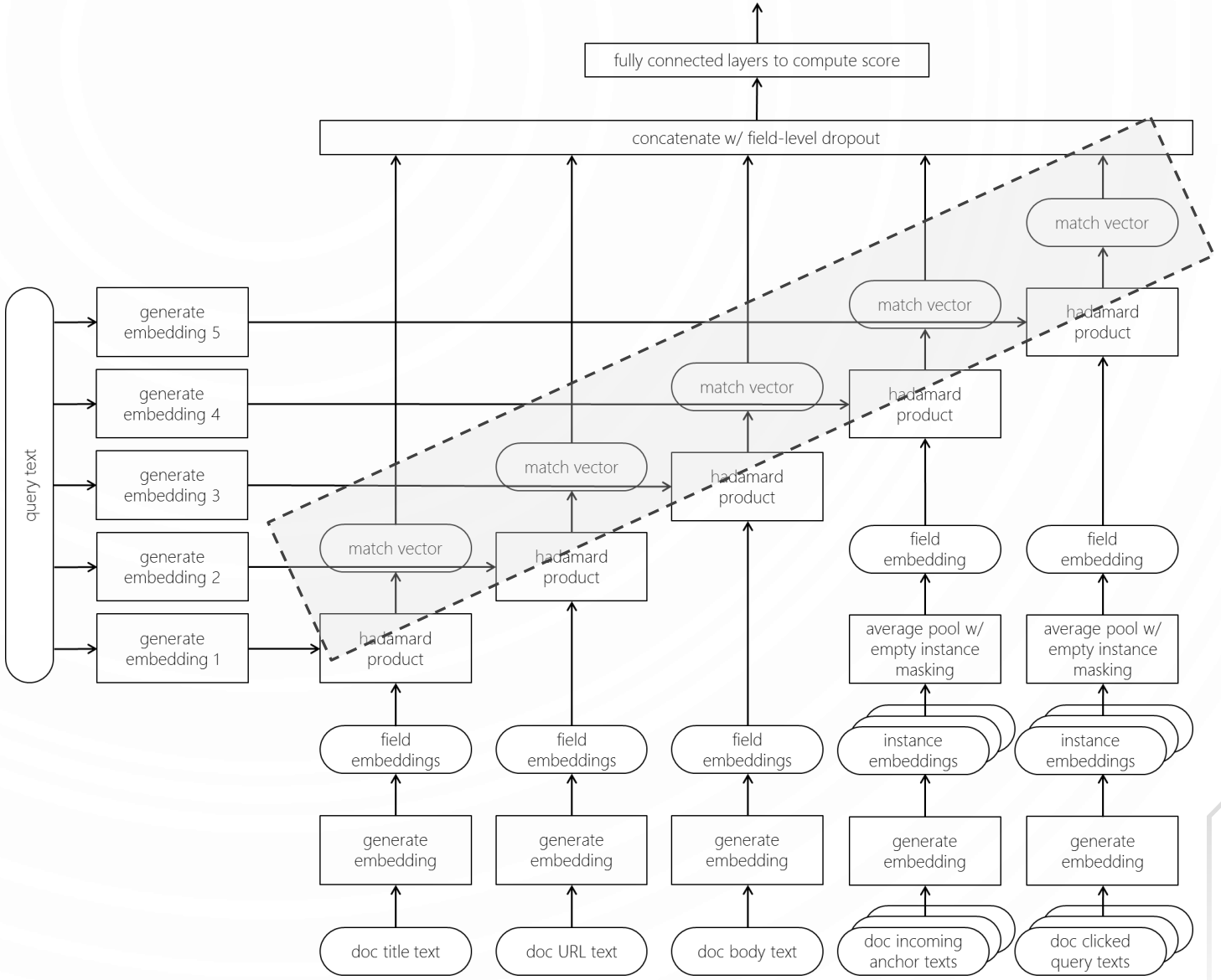
Ideal query representation for matching against URL likely to be different from for matching with title



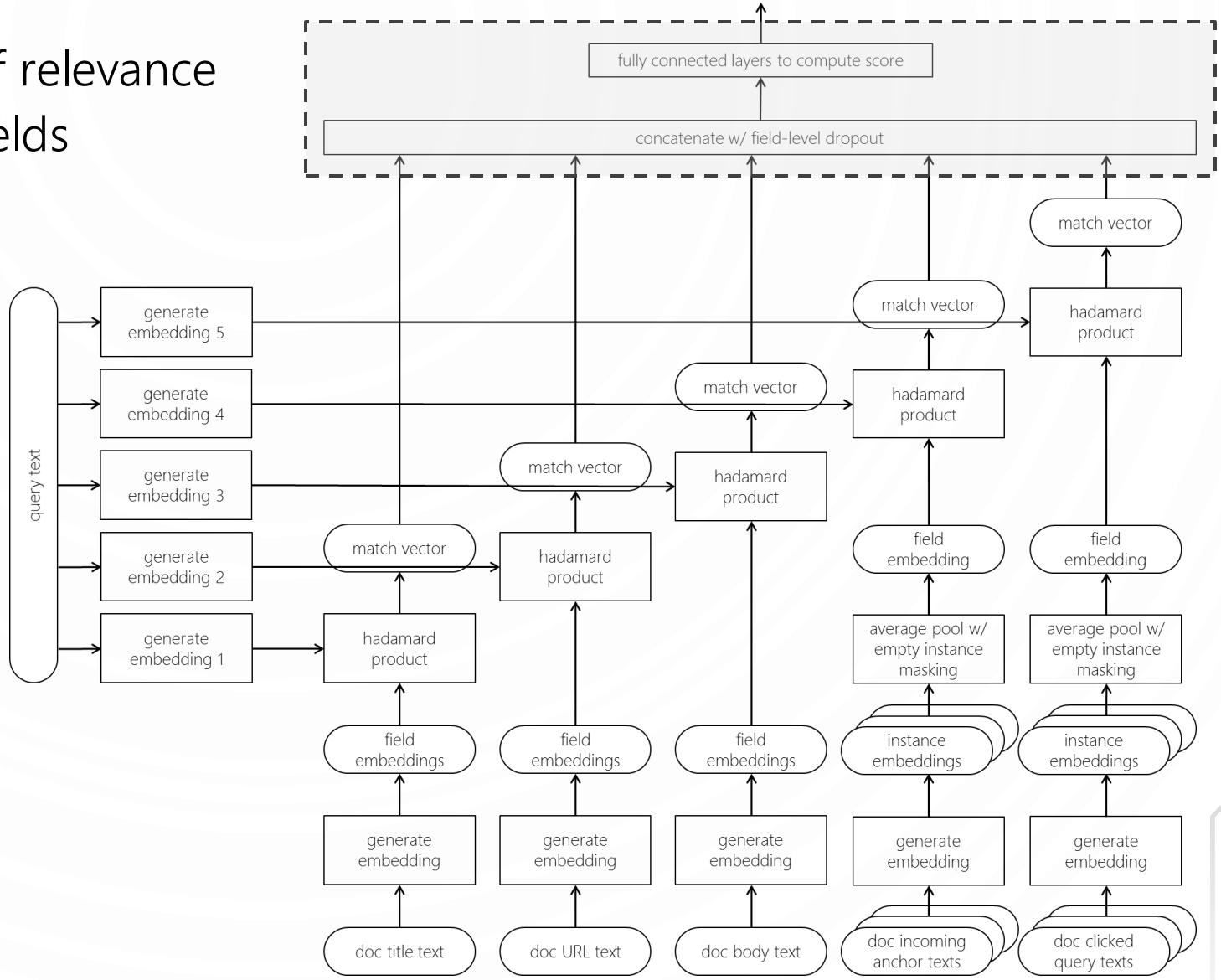
Represent per field match by a vector, not a score

Allows the model to validate that across the different fields all aspects of the query intent have been covered

(Similar intuition as BM25F)

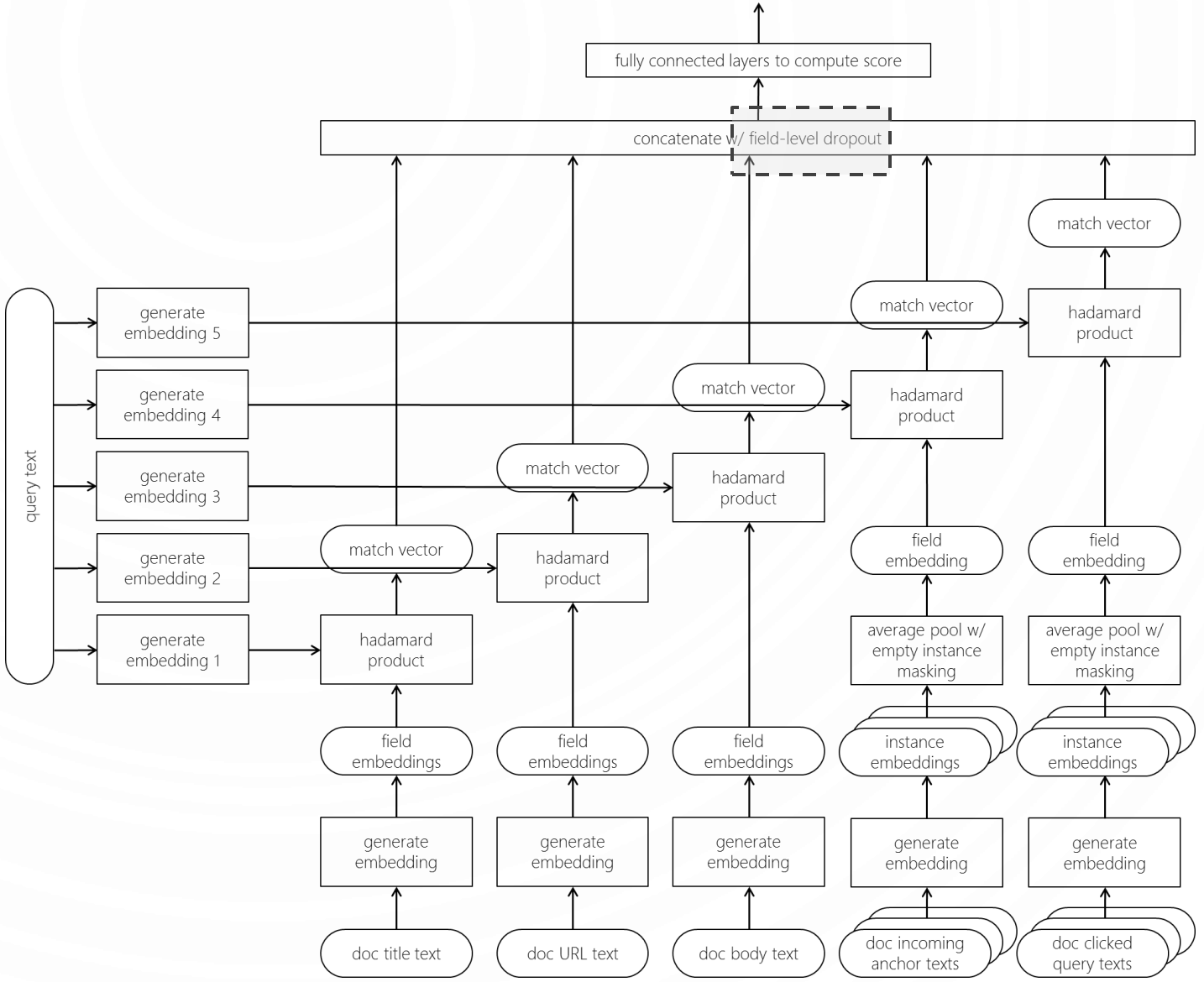


Aggregate evidence of relevance across all document fields



High precision fields, such as clicked queries, can negatively impact the modeling of the other fields

Field level dropout during training can regularize against over-dependency on any individual field



The design of both Duet and NRM-F models were motivated by long-standing IR insights

Deep neural networks have not reduced the importance of core IR research nor the studying and understanding of IR tasks

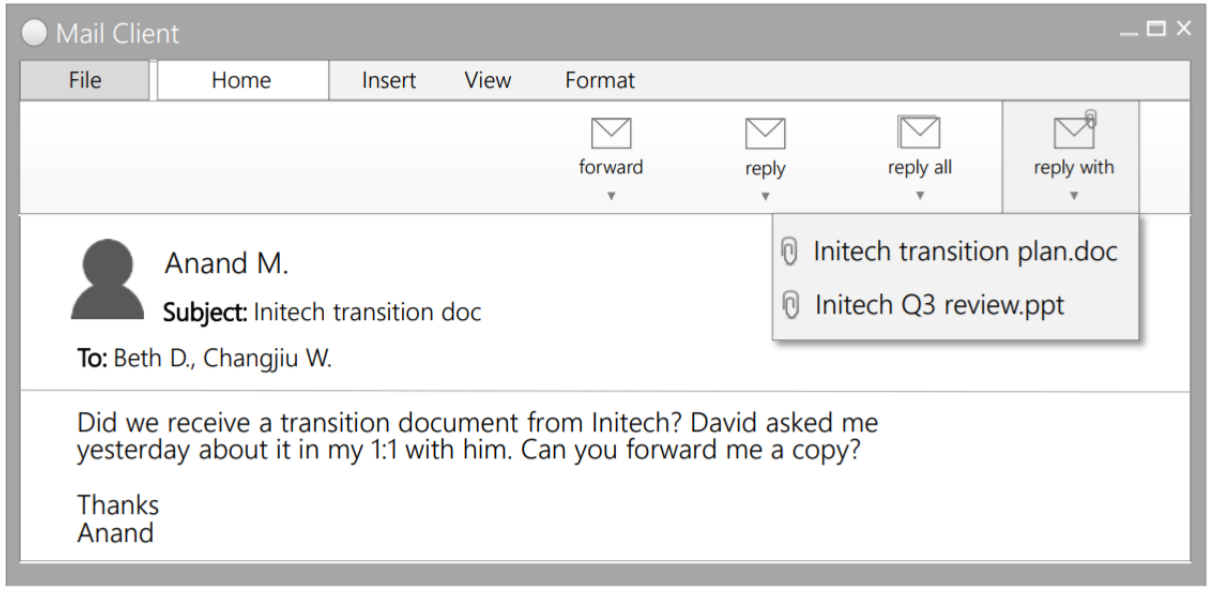
In fact, the same intuitions behind the design of classical IR models were important for feature design in early learning-to-rank models, and now in the architecture design of deep neural networks

Looking forward, there is also exciting opportunities for neural IR on emerging IR tasks, such as proactive retrieval…

For example,

REPLY WITH: PROACTIVE RECOMMENDATION OF EMAIL ATTACHMENTS

Given the context of the current conversation, the neural model formulates a short text query to pro-actively retrieve relevant attachments that the user may want to include in her response



(Van Gysel et al., 2017)
https://arxiv.org/abs/1710.06061

SUMMARY OF PAPERS DISCUSSED

AN INTRODUCTION TO NEURAL INFORMATION RETRIEVAL
Bhaskar Mitra and Nick Craswell, in **Foundations and Trends® in Information Retrieval,** Now Publishers, **2018** (upcoming).
https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/fntir-neuralir-mitra.pdf

NEURAL RANKING MODELS WITH MULTIPLE DOCUMENT FIELDS
Hamed Zamani, **Bhaskar Mitra**, Xia Song, Nick Craswell, and Saurabh Tiwary, in Proc. **WSDM, 2018** (upcoming).
(Pre-print coming soon)

REPLY WITH: PROACTIVE RECOMMENDATION OF EMAIL ATTACHMENTS
Christophe Van Gysel, **Bhaskar Mitra**, Matteo Venanzi, and others, in Proc. **CIKM, 2017** (upcoming).
https://arxiv.org/abs/1710.06061

LEARNING TO MATCH USING LOCAL AND DISTRIBUTED REPRESENTATIONS OF TEXT FOR WEB SEARCH
Bhaskar Mitra, Fernando Diaz, and Nick Craswell, in Proc. **WWW, 2017**.
https://dl.acm.org/citation.cfm?id=3052579

BENCHMARK FOR COMPLEX ANSWER RETRIEVAL
Federico Nanni, **Bhaskar Mitra**, Matt Magnusson, and Laura Dietz, in Proc. **ICTIR, 2017**.
https://dl.acm.org/citation.cfm?id=3121099

QUERY EXPANSION WITH LOCALLY-TRAINED WORD EMBEDDINGS
Fernando Diaz, **Bhaskar Mitra**, and Nick Craswell, in Proc. **ACL, 2016**.
http://anthology.aclweb.org/P/P16/P16-1035.pdf

A DUAL EMBEDDING SPACE MODEL FOR DOCUMENT RANKING
Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana, **arXiv preprint, 2016**.
https://arxiv.org/abs/1602.01137

IMPROVING DOCUMENT RANKING WITH DUAL WORD EMBEDDINGS
Eric Nalisnick, **Bhaskar Mitra**, Nick Craswell, and Rich Caruana, in Proc. **WWW, 2016**.
https://dl.acm.org/citation.cfm?id=2889361

QUERY AUTO-COMPLETION FOR RARE PREFIXES
Bhaskar Mitra and Nick Craswell, in Proc. **CIKM, 2015**.
https://dl.acm.org/citation.cfm?id=2806599

EXPLORING SESSION CONTEXT USING DISTRIBUTED REPRESENTATIONS OF QUERIES AND REFORMULATIONS
Bhaskar Mitra, in Proc. **SIGIR, 2015**.
https://dl.acm.org/citation.cfm?id=2766462.2767702

AN INTRODUCTION TO NEURAL INFORMATION RETRIEVAL

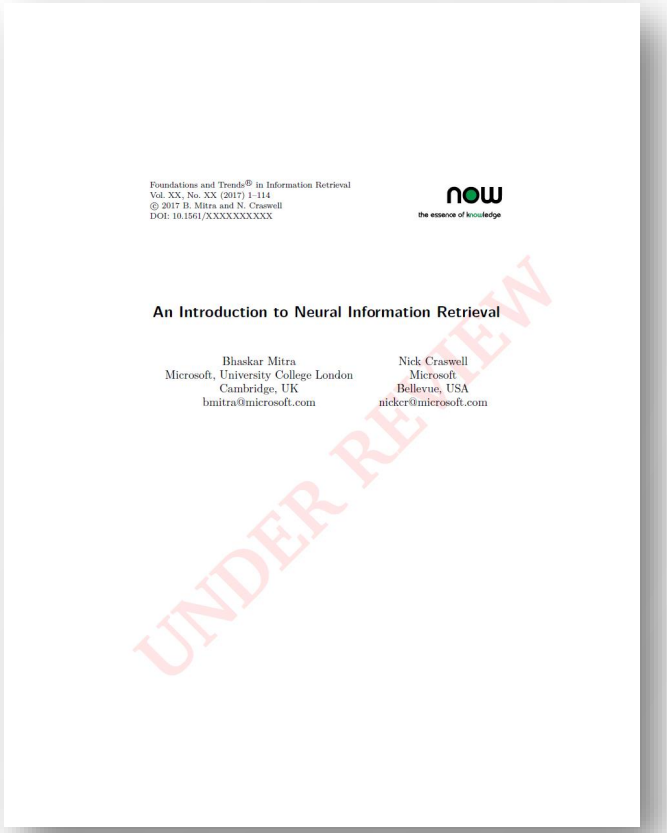
Manuscript under review for

Foundations and Trends® in Information Retrieval

Pre-print is available for free download

http://bit.ly/neuralir-intro

(Final manuscript may contain additional content and changes)



THANK YOU