Machine Learning for IR

Claudia Hauff

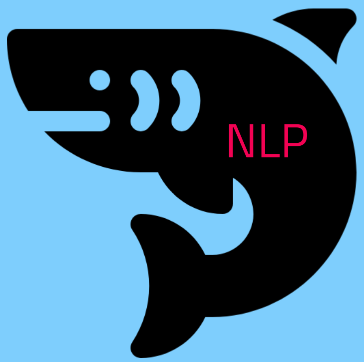
SIKS, October 7, 2019

TU Delft

Machine learning adapted to IR problems

Learning to Rank

Neural IR



MS MARCO

http://www.msmarco.org

Given a query (1M total) and a corpus of 9M passages, rank the passages by relevance. Use either the full corpus or start with BM25's top 1K passages.

```
symptoms of an enlarged heart in dogs
how long can i freeze pork
lewisville texas is in what county
...
```

Passage Retrieval(10/26/2018-Present)

Rank	Model	Ranking Style	Submission Date	MRR@10 On Eval
1	**Enriched BERT base + AOA index V1** Ming Yan of Alibaba Damo NLP	Full Ranking	May 13th, 2019	0.383
2	**BERTter pretraining** (1)Rodrigo Nogueira, (2)Wei Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4)	Full Ranking	May 21st, 2019	0.383
3	**Enriched BERT base + AOA index V2** Ming Yan of Alibaba Damo NLP	Full Ranking	May 13th, 2019	0.380
4	**BM25 + monoBERT + duoBERT + TCP** Anonymous	Full Ranking	June 26th, 2019	0.379
5	**BERT^2** (1)Rodrigo Nogueira, (2)Wei Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4)	Full Ranking	May 13th, 2019	0.375
6	**Enriched BERT base + AOA index** Ming Yan of Alibaba Damo NLP	Full Ranking	May 6th, 2019	0.373
7	**BM25 + monoBERT + duoBERT** Anonymous	Full Ranking	June 26th, 2019	0.370
8	**BERTter Indexing** (1)Rodrigo Nogueira, (2)Wei Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4) [Nogueira et al. '19] and [Code]	Full Ranking	April 8th, 2019	0.368
9	**Enriched BERT base + AOA index** Ming Yan of Alibaba Damo NLP	ReRanking	May 6th, 2019	0.368
10	**BM25 + monoBERT** Anonymous	Full Ranking	June 26th, 2019	0.365
11	**SAN + BERT base** Yu Wang, Xiaodong Liu, Jianfeng Gao - Deep Learning Group, Microsoft Research AI [Xiaodong, et al. '18]	ReRanking	January 22th, 2019	0.359
12	**BERT + Small Training** Rodrigo Nogueira(1) and Kyunghyun Cho(2) - New York University(1,2), Facebook AI Research(2) [Nogueira, et al. '19]	ReRanking	January 7th, 2019	0.359

Given a query (1M total) and a corpus of 9M passages, rank the passages by relevance. Use either the full corpus or start with BM25's top 1K passages.

```
symptoms of an enlarged heart in dogs
how long can i freeze pork
lewisville texas is in what county
...
```

40	**Feature-based LeToR: simple-feature based RankSVM**(1)Yifan Qiao, (2)Chenyan Xiong, (3)Zhenghao Liu, (4)Zhiyuan Liu-Tsinghua University(1, 3, 4); Microsoft Research AI(2)	ReRanking	December 10th, 2018	0.191
41	**BM25 (Lucene8, tuned)** Anonymous	Full Ranking	June 26th, 2019	0.190
42	**BM25 (Anserini)** (1)Rodrigo Nogueira, (2)Wei Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4)[Nogueira et al. '19] and [Code]	Full Ranking	April 10th, 2019	0.186
43	**Unnamed** Hongyin Zhu	ReRanking	June 26th, 2019	0.174
44	**[Official Baseline]BM25** Stephen E. Robertson; Steve Walker; Susan Jones; Micheline Hancock-Beaulieu & Mike Gatford (Implemented by MSMARCO Team) [Robertson et al. '94]	Full Ranking	Novmeber 1st, 2018	0.165

Passage Retrieval(10/26/2018-Present)

0.383

Rank	Model	Ranking Style	Submission Date	
1	**Enriched BERT base + AOA index V1** Ming Yan of Alibaba Damo NLP	Full Ranking	May 13th, 2019	
2	**BERTter pretraining** (1)Rodrigo Nogueira, (2)Wei Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4)	Full Ranking	May 21st, 2019	0.383
3	**Enriched BERT base + AOA index V2** Ming Yan of Alibaba Damo NLP	Full Ranking	May 13th, 2019	0.380
4	**BM25 + monoBERT + duoBERT + TCP** Anonymous	Full Ranking	June 26th, 2019	0.379
5	**BERT^2** (1)Rodrigo Nogueira, (2)Wei Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4)	Full Ranking	May 13th, 2019	0.375
6	**Enriched BERT base + AOA index** Ming Yan of Alibaba Damo NLP	Full Ranking	May 6th, 2019	0.373
7	**BM25 + monoBERT + duoBERT** Anonymous	Full Ranking	June 26th, 2019	0.370
8	**BERTter Indexing** (1)Rodrigo Nogueira, (2)Wei Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4) [Nogueira et al. '19] and [Code]	Full Ranking	April 8th, 2019	0.368
9	**Enriched BERT base + AOA index** Ming Yan of Alibaba Damo NLP	ReRanking	May 6th, 2019	0.368
10	**BM25 + monoBERT** Anonymous	Full Ranking	June 26th, 2019	0.365
11	**SAN + BERT base** Yu Wang, Xiaodong Liu, Jianfeng Gao - Deep Learning Group, Microsoft Research AI [Xiaodong, et al. '18]	ReRanking	January 22th, 2019	0.359
12	**BERT + Small Training** Rodrigo Nogueira(1) and Kyunghyun Cho(2) - New York University(1,2), Facebook AI Research(2) [Nogueira, et al. '19]	ReRanking	January 7th, 2019	0.359

Given a query (1M total) and a corpus of 9M passages, rank the passages by relevance. Use either the full corpus or start with BM25's top 1K passages.

```
symptoms of an enlarged heart in dogs
how long can i freeze pork
lewisville texas is in what county
...
```

0.190

40	**Feature-based LeToR: simple-feature based RankSVM**(1)Yifan Qiao, (2)Chenyan Xiong, (3)Zhenghao Liu, (4)Zhiyuan Liu-Tsinghua University(1, 3, 4); Microsoft Research AI(2)	ReRanking	Decer 10th, 2	
41	**BM25 (Lucene8, tuned)** Anonymous	Full Ranking	June 26th, 2019	0.190
42	**BM25 (Anserini)** (1)Rodrigo Nogueira, (2)Wei Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4)[Nogueira et al. '19] and [Code]	Full Ranking	April 10th, 2019	0.186
43	**Unnamed** Hongyin Zhu	ReRanking	June 26th, 2019	0.174
44	**[Official Baseline]BM25** Stephen E. Robertson; Steve Walker; Susan Jones; Micheline Hancock-Beaulieu & Mike Gatford (Implemented by MSMARCO Team) [Robertson et al. '94]	Full Ranking	Novmeber 1st, 2018	0.165

BERT: Pre-training of Deep Bidirectional Transformers for ...

https://arxiv.org › cs ▾

by J Devlin - 2018 - Cited by 1694 - Related articles

Oct 11, 2018 - Unlike recent language representation models, **BERT** is designed to ... As a result, the pre-trained **BERT** model can be fine-tuned with just one ...

XLNet: Generalized Autoregressive Pretraining for Language ...

https://arxiv.org › cs ▾

by Z Yang - 2019 - Cited by 73 - Related articles

Jun 19, 2019 - In light of these pros and cons, we propose **XLNet**, a generalized autoregressi pretraining method that (1) enables learning bidirectional ...

8	Yang, (3)Jimmy Lin, (4)Kyunghyun Cho - New York University(1,4), University of Waterloo(2,3), Facebook AI Research(4) [Nogueira et al. '19] and [Code]	Full Ranking	201
9	**Enriched BERT base + AOA index** Ming Yan of Alibaba Damo NLP	ReRanking	Ma 201
10	**BM25 + monoBERT** Anonymous	Full Ranking	Jun 201
11	**SAN + BERT base** Yu Wang, Xiaodong Liu, Jianfeng Gao - Deep Learning Group, Microsoft Research AI [Xiaodong, et al. '18]	ReRanking	Jan 22t
12	**BERT + Small Training** Rodrigo Nogueira(1) and Kyunghyun Cho(2) - New York University(1,2), Facebook AI Research(2) [Nogueira, et al. '19]	ReRanking	Jan 201

Kyunghyun Cho
@kchonyc

re "Why not consider other models? such as XLNet": I agree with the reviewer on the importance of time travel research, but it's slightly out of the scope of this paper.

1:44 AM · Jul 13, 2019 · Twitter for Android

74 Retweets **562** Likes

Learning to Rank

[1] Liu, Tie-Yan. "Learning to rank for information retrieval." *Foundations and Trends® in Information Retrieval* Vol. 3, No. 3 (2009): 225-331.

Conventional ranking models in IR

Query-dependent models

Vector space model

Boolean model

BM25

Language modeling

...

Query-independent models

PageRank

TrustRank

Spaminess

Readability

...

How can we **combine** a large number of models to obtain an even better model?

Overview

Learning-to-rank

<u>in the broad sense</u> are all methods that use machine learning to solve the problem of ranking

(e.g. relevance feedback, hyperparameter tuning of BM25 ...)

Learning-to-rank

<u>in the narrow sense</u> are all methods that learn the optimal way to combine **features** extracted from query-document pairs through **discriminative** training

Learns the conditional probability distribution $P(y|x)$. Generative training learns $P(x,y)$ instead.

Given your machine learning background, how would you go about using machine learning to **learn a ranking function**?

Ignore deep learning for now ....

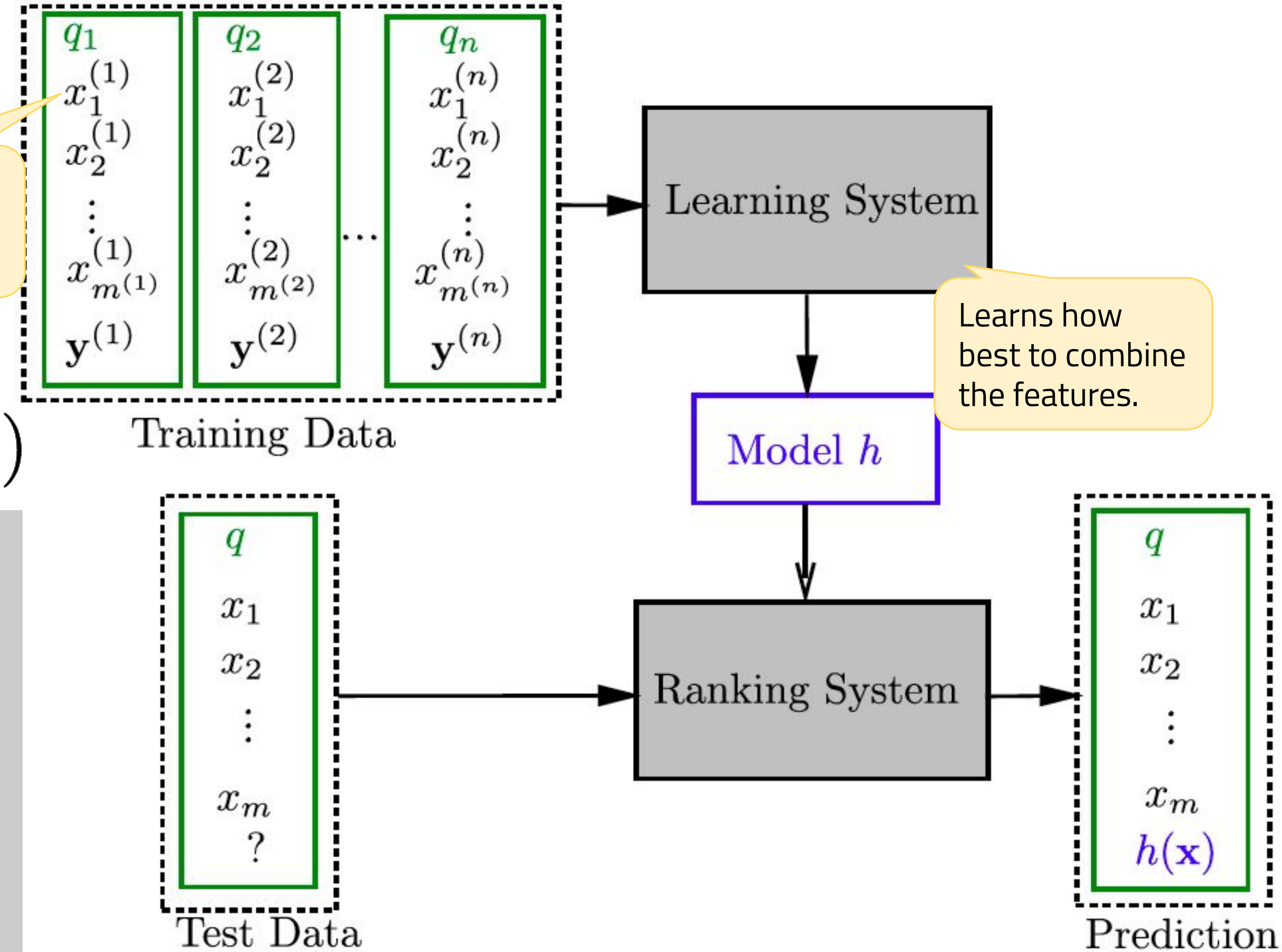
What is the input?

What is the output?

What is your success metric?

L2R setup

m documents associated with query *i*



Learns how best to combine the features.

$$x = \Phi(d, q)$$

Documents are represented by feature vectors.

Φ is a feature extractor.

BM25, PageRank, etc. can be a feature.

Image source: [1]

Document judgment strategies

with respect to a query (topic)

- Specifying whether a document is relevant (binary) or specifying a degree of relevance

- Specifying whether a document is *more* relevant than another one (relative preference)

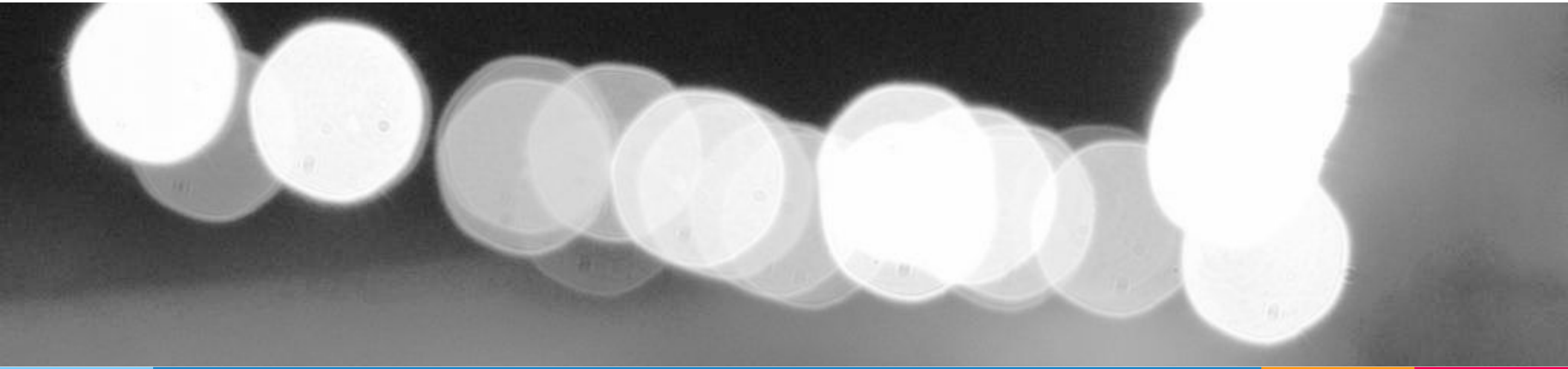
- Specifying the partial or total order of documents (a set of permutations)

Explicit vs. implicit

Learning to rank, BM25, LM ...

- Need **training data** to effectively learn the models' (hyper)parameters
- Often explicit relevance judgments are used

Explicit qrels are extremely **expensive** to accumulate (can become outdated quickly for dynamic collections)



TREC-8 numbers
(ran in 1999)

50

Topics (ad hoc task)

86,830

Pooled documents (k=100)

129

Systems

723

Assessor hours

At $20 an hour, that amounts to $14,460.

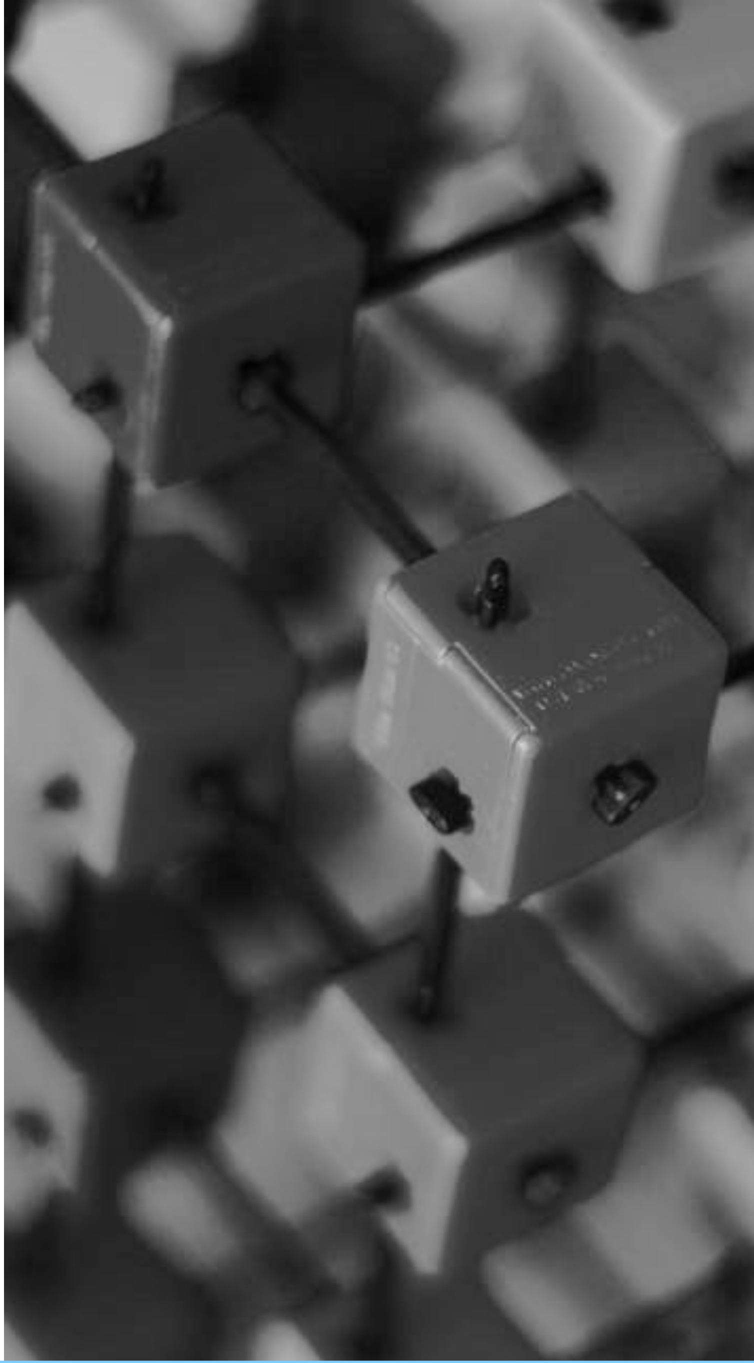
And thus, We are still using the TREC-8 corpus to this day for experiments!

LETOR features

LEarning **TO R**ank for Information Retrieval

TF(Term frequency) of body
TF of anchor
TF of title
TF of URL
TF of whole document
IDF(Inverse document frequency) of body
IDF of anchor
IDF of title
IDF of URL
IDF of whole document
TF*IDF of body
TF*IDF of anchor
TF*IDF of title
TF*IDF of URL
TF*IDF of whole document
DL(Document length) of body
DL of anchor
DL of title
DL of URL
DL of whole document
BM25 of body
BM25 of anchor
BM25 of title
BM25 of URL

BM25 of whole document
LMIR.ABS of body
LMIR.ABS of anchor
LMIR.ABS of title
LMIR.ABS of URL
LMIR.ABS of whole document
LMIR.DIR of body
LMIR.DIR of anchor
LMIR.DIR of title
LMIR.DIR of URL
LMIR.DIR of whole document
LMIR.JM of body
LMIR.JM of anchor
LMIR.JM of title
LMIR.JM of URL
LMIR.JM of whole document
PageRank
Inlink number
Outlink number
Number of slash in URL
Length of URL
Number of child page



[2] Qin, Tao, and Tie-Yan Liu. "Introducing LETOR 4.0 datasets." arXiv preprint arXiv:1306.2597 (2013).

Approaches

Input space (feature vectors)
Output space (learning target)
Hypothesis space
Loss function (prediction vs. ground truth)

Pointwise approach

Each document for itself

Input space: feature vector of each doc

Output space: relevance degree of each document

Hypothesis space: functions that take a doc. feature vector as input and output a relevance degree

Regression or classification loss

Pairwise approach

Each doc. pair for itself

Input space: feature vectors of a pair of docs

Output space: pairwise preferences

Hypothesis space: functions that take a document pair as input and output their relative order

Loss function considers the relative order between the two docs

Listwise approach

Designed for ranking

Input space:

$$\mathbf{x} = \{x_j\}_{j=1}^{m}$$

Output space: (1) relevance degrees of all documents, (2) ranked list of docs

Hypothesis space: functions that take $\boldsymbol{x}$ as input and yield (1) or (2)

Loss function considers (1) or (2)

Input space (feature vectors)
Output space (learning target)
Hypothesis space
Loss function (prediction vs. ground truth)

Approaches

Pointwise approach

Each document for itself.

Input space: feature vector of each doc.

Output space: relevance degree of each document

Hypothesis space: functions th... doc. feature... input and ou... relevance degree

Regression or classification loss.

Pairwise approach

Each doc. pair for itself.

Input space: feature vectors of a pair of docs

Output space: pairwise preferences

Hypothesis space: functions that take a ...t pair as input ...uts their ...rder

Loss function considers the relative order between the two docs.

Different loss functions but one and the same evaluation metric (e.g. MAP)

Listwise approach

Designed for ranking.

Input space:

$$\mathbf{x} = \{x_j\}_{j=1}^{m}$$

Output space: (1) relevance degrees of all documents, (2) ranked list of documents

Hypothesis space: functions that take $\boldsymbol{x}$ as input and produce (1) or (2)

Loss function considers (1) or (2)

How to rank a whole set of documents? Another step is needed.

Document ranking is easy.

L2R categorization

	SVM	Boosting	Neural net	Others
Pointwise		McRank		PRank
Pairwise	RankSVM	RankBoost, LambdaMART, GBRank	RankNet, LambdaRank, FRank	
Listwise	SVM MAP	AdaRank	ListNet	SoftRank, SmoothRank

Tax et al. (2015): "*ListNet, SmoothRank, FenchelRank, FSMRank, LRUF and LARF are Pareto optimal learning to rank methods*"

[3] Tax, Niek, Sander Bockting, and Djoerd Hiemstra. "A cross-benchmark comparison of **87 learning to rank methods.**" Information processing & management 51.6 (2015): 757-772.

Pointwise approach

Direct application of standard supervised ML.

Pointwise categories

- Regression based algorithms

  Real-valued relevance scores

- Classification based algorithms

  Non-ordered categories

- Ordinal regression based algorithms

  Variables with a natural categorical ordering

Main issue of regression?

Qualitative judgment is encoded quantitatively

Polynomial regression function

$$Given\ q\ \ and\ associated\ \mathbf{x} = \{x_j\}_{j=1}^m,$$

$$let\ the\ ground\ truth\ label\ be:$$

doc judged non-relevant

doc judged relevant

$$\textbf{binary:}\ \ \vec{y_j} = (0,1)\ \ or\ \vec{y_j} = (1,0)$$

$$\textbf{ordered categories:}\ \ \vec{y_j} = (0,0,..,1,..0)$$

doc judged belonging to a category (e.g. 'Fair' or 'Good')

Predictor of kth element in the ground truth vec.

$$Scoring\ function:\ \vec{f} = (f_1, f_2, ..., \boxed{f_k})\ \ with$$

Tth feature in feature vector j

$$f_k(x_j) = w_{k,0} + w_{k,1} \times x_{j,1} + ... + w_{k,T} \times \boxed{x_{j,T}}$$
$$+ \boxed{w_{k,T+1}} \times x_{j,1}^2 + w_{k,T+2} \times x_{j,1} \times x_{j,2} + ...$$

Combination coefficient

[4] Fuhr, Norbert. "Optimum polynomial retrieval functions based on the probability ranking principle." ACM Transactions on Information Systems (TOIS) 7.3 (1989): 183-204.

Pairwise approach

Pairwise

Focus: **relative ordering** of pairs of documents

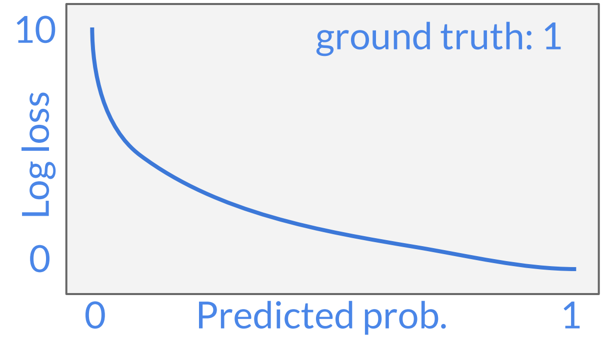
Ranking problem **reduced** to a classification problem
(goal: minimize #misclassified pairs)

IR issue: **target cost** vs. **optimization cost** (tractable opt.,
good approximation of target cost)

Training data:

$$\{(x_1, x_2, +1), (x_1, x_3, -1), ..., (x_i, x_j, +1), ...\}$$

RankNet



$$Given\ q\ and\ two\ documents\ x_u\ and\ x_v,$$

$$\textbf{modeled prob:}\ \boxed{P_{u,v}(f)} = \frac{\exp(f(x_u) - f(x_v))}{1 + \exp(f(x_u) - f(x_v))}$$

Based on the diff. between the two documents' scores

Shallow neural network learns scoring function f; gradient descent as optimization alg

$$\boxed{L(f; x_u, x_v, y_{u,v})} = -\boxed{\bar{P}_{u,v}} \log P_{u,v}(f) - (1 - \boxed{\bar{P}_{u,v}}) \log(1 - P_{u,v}(f))$$

Cross-entropy loss

Target probability:

$$\bar{P}_{u,v} = 1,\ \ if\ y_{u,v} = 1;$$

$$\bar{P}_{u,v} = 0\ \ otherwise$$

[5] Burges, Christopher JC. "From ranknet to lambdarank to lambdamart: An overview." Learning 11.23-581 (2010): 81.

RankNet - loss function issue

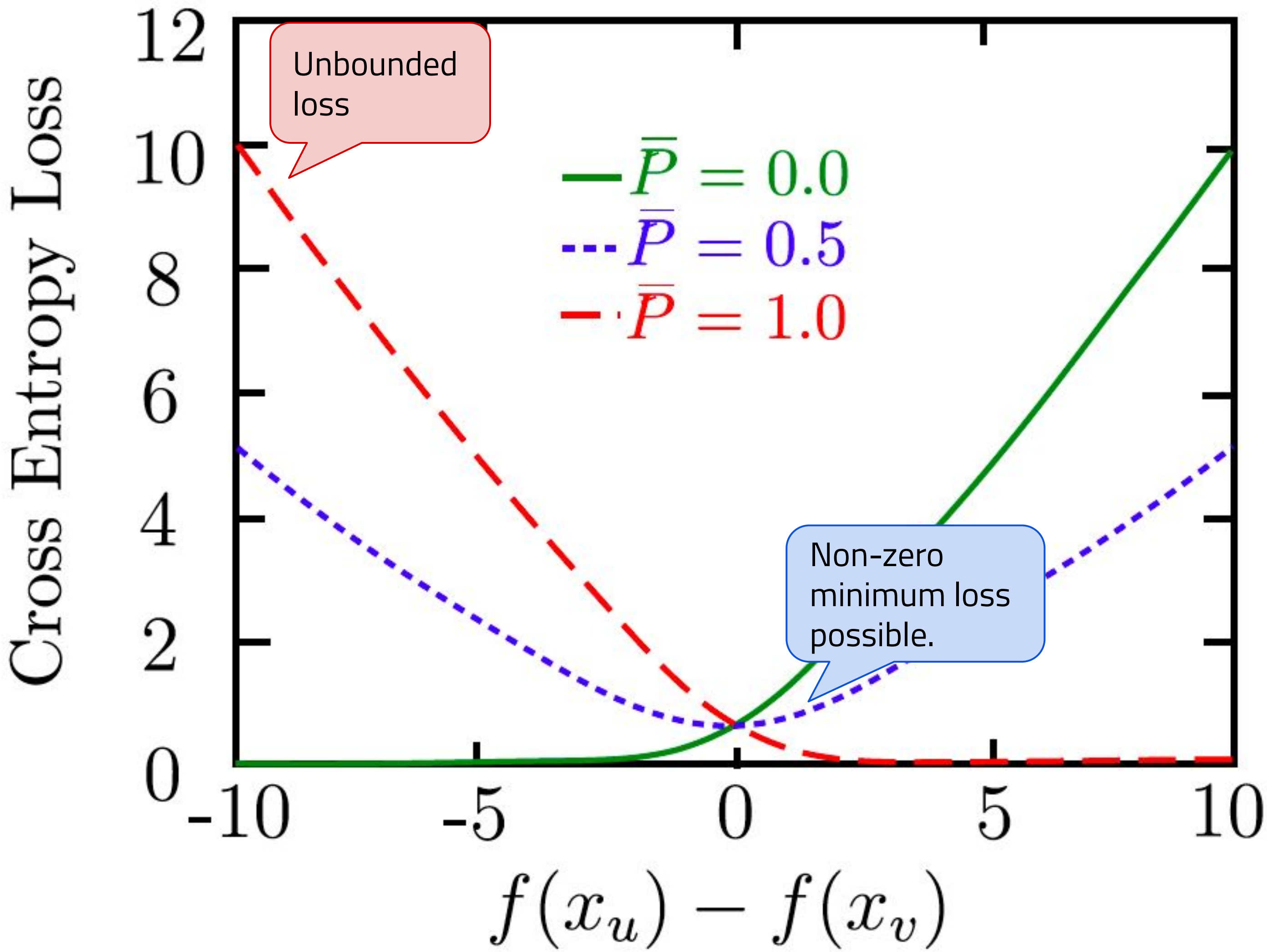
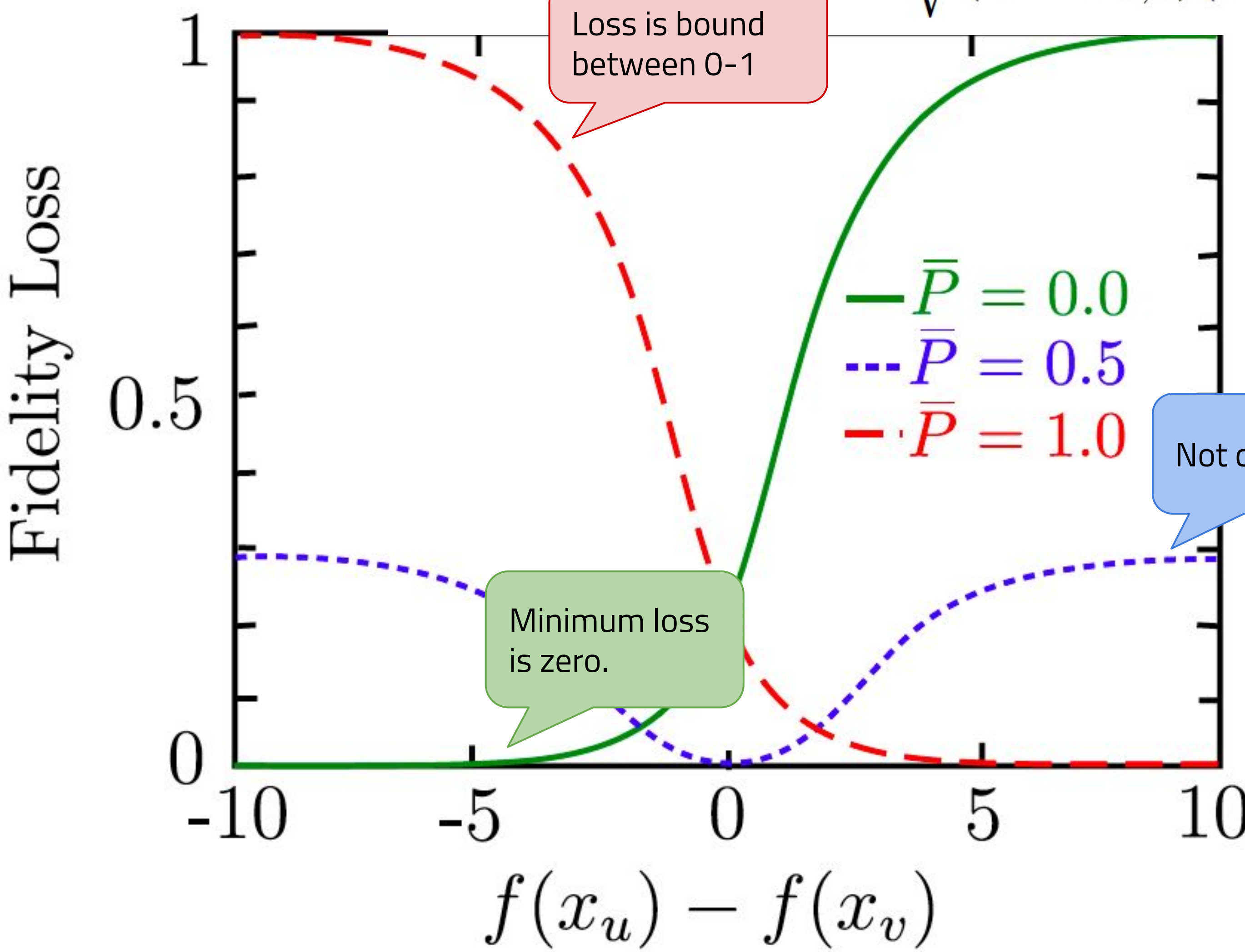


Image source: [1]

$$L(f; x_u, x_v, y_{u,v}) = 1 - \sqrt{\bar{P}_{u,v} P_{u,v}(f)} - \sqrt{(1 - \bar{P}_{u,v})(1 - P_{u,v}(f))}$$

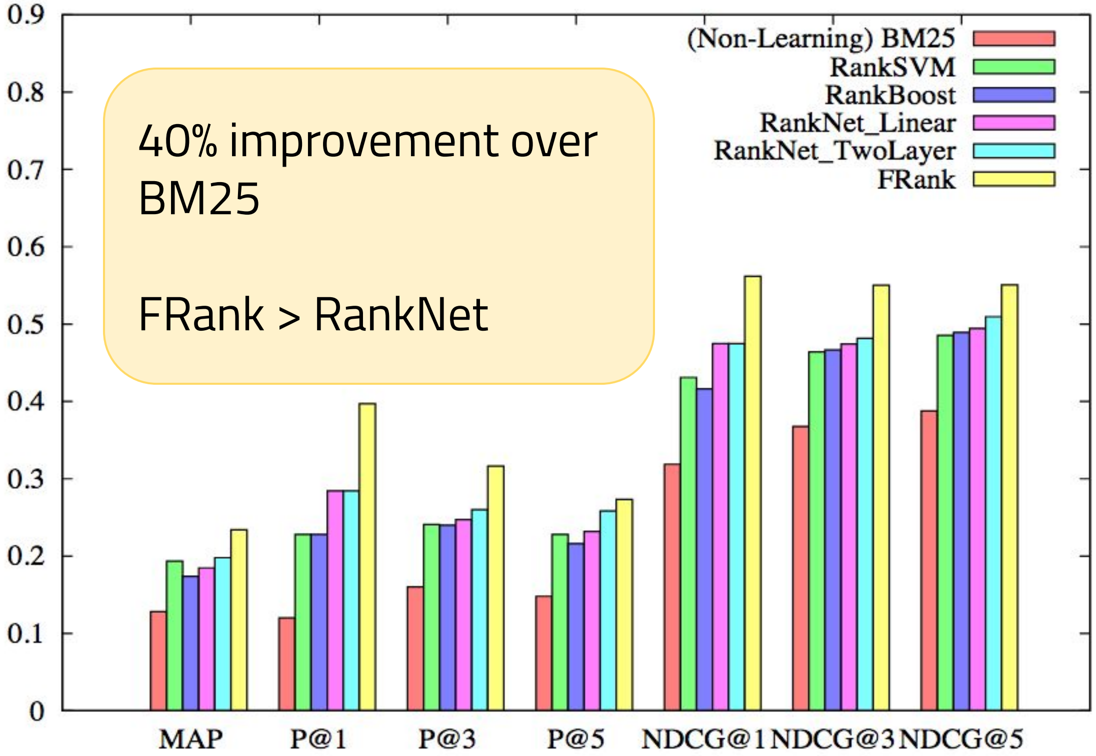
FRank: Fidelity loss



[6] Tsai, Ming-Feng, et al. "FRank: a ranking method with fidelity loss." SIGIR. ACM, 2007.

Image source: [1]

Experimentally: RankNet vs. FRank



TREC topic distillation aims at finding key resources which are high-quality pages for certain topics.

Corpus: 1M `.gov` pages, 14 features per document

50 topics (between 1 and 86 relevant docs per topic)

4-fold cross validation

Is training of the pairwise approach slower/faster than the pointwise one?

Image source: [6]

Document pair issue

Document pairs only make it into the training set if their relevance degrees differ

Queries differ widely in the number of pairs they generate

Loss will be dominated by queries with many pairs

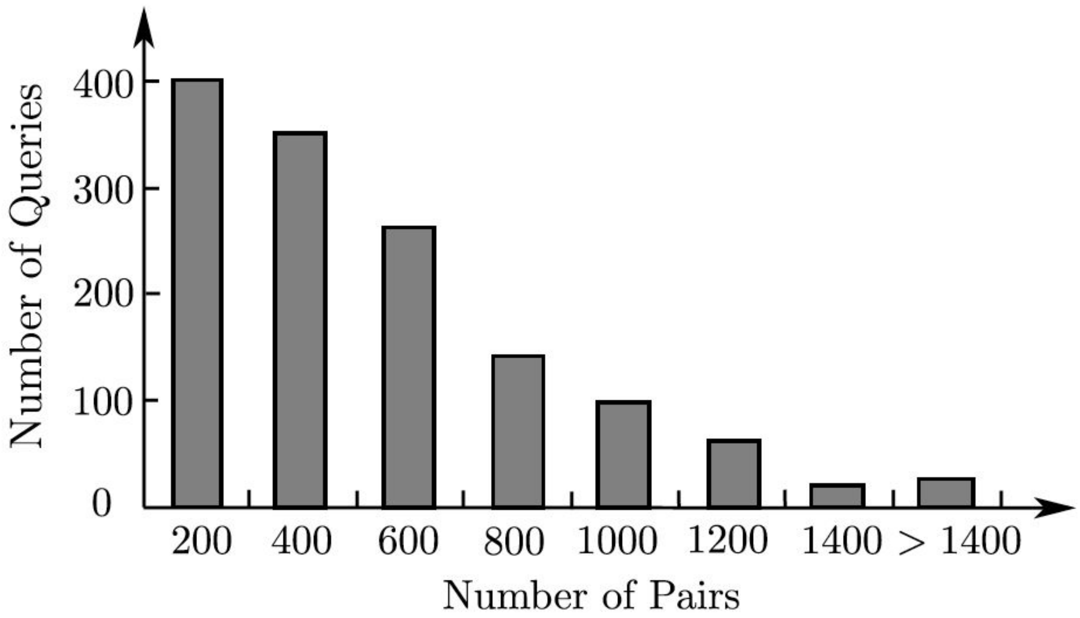


Image source: [1]

Beyond RankNet ... [6]

>> LambdaRank

- Insight: neural net training requires only the *gradients* of the cost function
- Heuristic rules on how the cost changes if document rankings are swapped

>> LambdaMART

- Gradient boosted decision trees

Burges, Christopher JC. "From ranknet to lambdarank to lambdamart: An overview." Learning 11.23-581 (2010): 81.

Listwise approach

Listwise

1) Optimize a continuous & differentiable approximation of an IR metric.
2) Optimize a continuous & differentiable bound of an IR metric.
3) Choose optimization approach that can handle complex objectives.

Direct optimization

Output space contains relevance degrees of all docs associated with q.

Loss function **optimizes an IR metric.** Not easy as MAP, NDCG, … are non-continuous & non-differentiable.

Most optimization techniques are designed for continuous and differentiable functions.

Permutation-based

The output space contains the permutation of the documents associated with q.

The loss function measures the difference between the permutation given by the hypothesis and the ground truth permutation.

Example: ListNet

ListNet

Required: a loss function that considers the document list

Idea: define two probability distributions, one on the hypothesized and one on the reference ranking. Use a metric that **compares the two probability distributions** as loss function.

Which metric?

Given scoring function f

and document relevance scores $\mathbf{s} = \{s_j\}_{j=1}^{m}$, *where* $s_j = f(x_j)$,

define a prob. for each possible permutation π *of the documents:*

$$P(\pi|\mathbf{s}) = \prod_{j=1}^{m} \underbrace{\frac{\varphi(s_{\pi^{-1}(j)})}{\sum_{u=j}^{m} \varphi(s_{\pi^{-1}(u)})}}_{\text{conditional probability}}$$

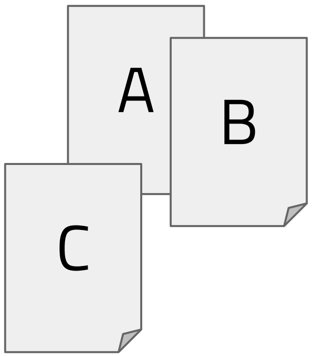
permutation probability

doc. ranked at jth position in the permutation

transformation function (e.g. exponential)

[7] Cao, Zhe, et al. "Learning to rank: from pairwise approach to listwise approach." ICML. ACM, 2007.

ListNet permutation example



Given 3 documents for query q, what is the probability of the ranking permutation A-B-C?

$$P_\pi = P_1 \times P_2 \times P_3$$

$$P_1 = \frac{\varphi(s_A)}{\varphi(s_A) + \varphi(s_B) + \varphi(s_C)}$$

Probability of doc A being ranked at the top. Determined by comparing A's score to B's and C's scores.

$$P_2 = \frac{\varphi(s_B)}{\varphi(s_B) + \varphi(s_C)}$$

Probability of B being ranked at position 2, given that A has been ranked already.

$$P_3 = 1$$

Only C is left.

ListNet

ListNet defines the permutation probability distribution based on the scores given by the scoring function.

The **reference permutation probability distribution** is based on the ground truth labels.

KL divergence between both distributions to define the listwise ranking loss:

$$L(f;\mathbf{x},\pi_y) = D(P(\pi\,|\,\varphi(f(w,\mathbf{x})))\|P_y(\pi))$$

Shallow neural network learns scoring function f; gradient descent as optimization alg

Practical issue: training over all possible permutations of a list is impractical ($m!$ permutations of size m)

Benchmarks & practice

	Queries	Doc.	Rel.	Feat.	Year
LETOR 3.0 – Gov	575	568 k	2	64	2008
LETOR 3.0 – Ohsumed	106	16 k	3	45	2008
LETOR 4.0	2,476	85 k	3	46	2009
Yandex	20,267	213 k	5	245	2009
Yahoo!	36,251	883 k	5	700	2010
Microsoft	31,531	3,771 k	5	136	2010

Burges et al. (2011) used a linear combination of 12 ranking models, 8 of which were LambdaMART (Burges, 2010) boosted tree models, 2 of which were LambdaRank neural nets, and 2 of which were logistic regression models. While LambdaRank was originally instantiated using neural nets, LambdaMART implements the same ideas using the boosted-tree style MART algorithm, which itself may be viewed as a gradient descent algorithm. Four of the LambdaMART rankers (and one of the nets) were trained using the ERR measure, and four (and the other net) were trained using NDCG. Extended training sets were also generated by randomly deleting feature vectors for each query.

[8] Chapelle, Olivier, and Yi Chang. "Yahoo! learning to rank challenge overview." Proceedings of the learning to rank challenge. 2011.

Image source: [8]

Implicit feedback example

[9] Agichtein, Eugene, Eric Brill, and Susan Dumais. "Improving web search ranking by incorporating user behavior information." SIGIR. ACM, 2006.

Potential source: clickthrough data

RQ: How effective is **implicit feedback** in a large-scale web environment?

- Web search engines use hundreds of features and are heavily tuned (a 2006 paper)
- Tuning is a continuous process
- Long-tail issues (no feedback for rare queries)

RQ: How can implicit feedback be **combined** with the existing ranking produced by the search system?

- Reranking vs. RankNet

RQ: Can we move beyond clicks?

Clickthrough data in L2R

1) Derive a set of features from implicit feedback
2) At runtime, the search engine needs to fetch the implicit feedback features associated with each `(query, URL)` pair

L2R needs to be robust to missing values: *long tail* issue

Here: **RankNet**

- Neural net based tuning algorithm that optimizes feature weights to best match explicitly provided **pairwise** user preferences
- Has both train- and run-time efficiency
- Aggregate `(query, URL)` pair features across all instances in the session logs

How does a search engine get this data?

Features

Different types of user action features

Directly observed vs. derived features (derivations)

Browsing behaviour after the result has been clicked

Snippet based features are included as users often determine relevance based on snippet information

Clickthrough features	
Position	Position of the URL in Current ranking
ClickFrequency	Number of clicks for this query, URL pair
ClickProbability	Probability of a click for this query and URL
ClickDeviation	Deviation from expected click probability
IsNextClicked	1 if clicked on next position, 0 otherwise
IsPreviousClicked	1 if clicked on previous position, 0 otherwise
IsClickAbove	1 if there is a click above, 0 otherwise
IsClickBelow	1 if there is click below, 0 otherwise
Browsing features	
TimeOnPage	Page dwell time
CumulativeTimeOnPage	Cumulative time for all subsequent pages after search
TimeOnDomain	Cumulative dwell time for this domain
TimeOnShortUrl	Cumulative time on URL prefix, no parameters
IsFollowedLink	1 if followed link to result, 0 otherwise
IsExactUrlMatch	0 if aggressive normalization used, 1 otherwise
IsRedirected	1 if initial URL same as final URL, 0 otherwise
IsPathFromSearch	1 if only followed links after query, 0 otherwise
ClicksFromSearch	Number of hops to reach page from query
AverageDwellTime	Average time on page for this query
DwellTimeDeviation	Deviation from average dwell time on page
CumulativeDeviation	Deviation from average cumulative dwell time
DomainDeviation	Deviation from average dwell time on domain
Query-text features	
TitleOverlap	Words shared between query and title
SummaryOverlap	Words shared between query and snippet
QueryURLOverlap	Words shared between query and URL
QueryDomainOverlap	Words shared between query and URL domain
QueryLength	Number of tokens in query
QueryNextOverlap	Fraction of words shared with next query

Image source: [9]

Evaluation

Random sample of queries from a Microsoft query log with associated results and traces of user actions

8 weeks of user interactions with 1.2M unique queries (sufficient interactions for 50% of queries) and 12M interactions

On average, 30 results judged per query by human assessors on a **six point scale** (83K results judged)

	MAP
BM25F (content + link-based info)	0.184
RankNet	0.215
ReRanking (independent evidence)	
BM25F + Click-through statistics only	0.215
BM25F + Implicit feedback	0.222
Integrated as features	
RankNet + Implicit feedback	0.248

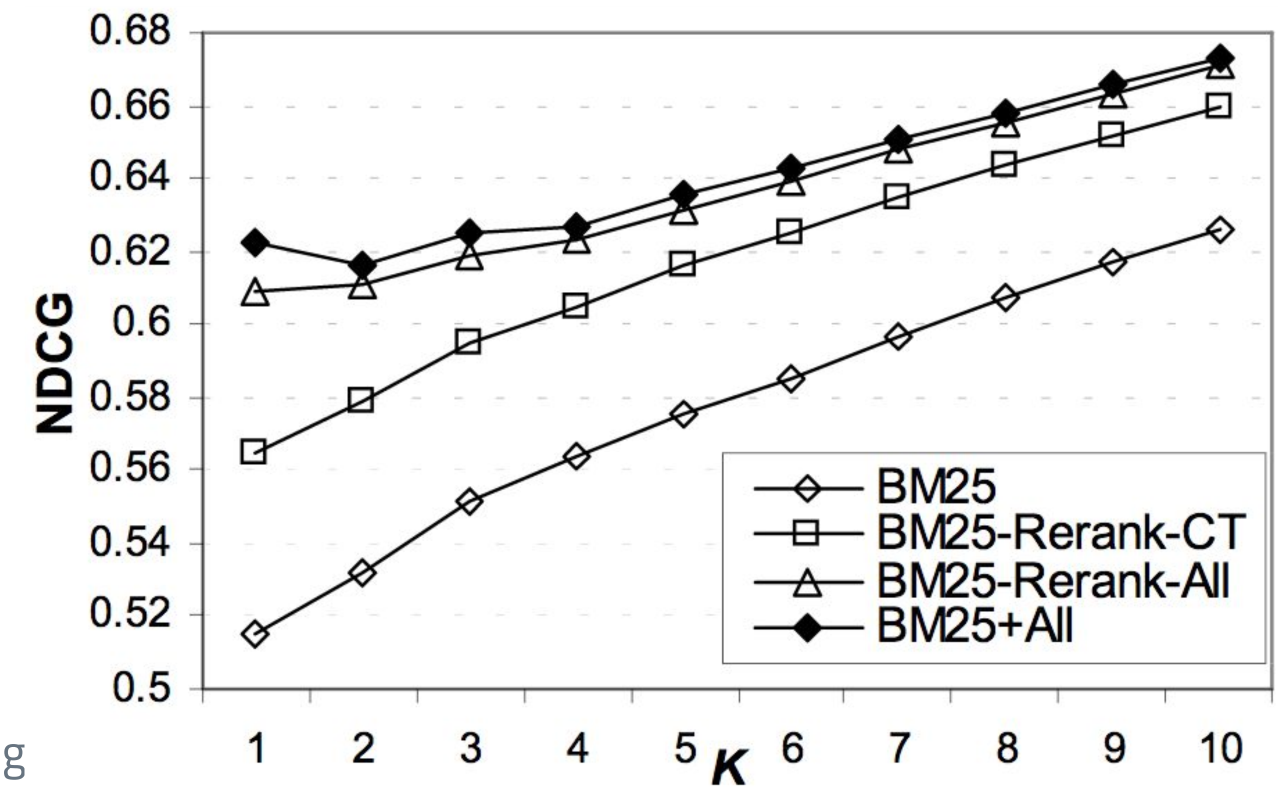
Results

`BM25F`: content-based (fields) and query-independent link-based information (PageRank, URL depth, etc.); does *not* make use of implicit/explicit feedback

`BM25F-RerankCT`: reranking based on clickthrough statistics (weight $w=1000$)

`BM25F-RerankAll`: RankNet-based reranking with all behavioural features

`BM25F+All`: RankNet-based ranking on BM25F features+IF



https://dl.acm.org/citation.cfm?id=1148177

Implicit feedback can replace hundreds of features

`RankNet (RN)`: hundreds of features of a Web search engine; based on explicit judgments

`RankNet+All`: including IF features

`BM25F`: content-based (fields) and query-independent link-based information (PageRank, URL depth, etc.)

`BM25F+All`: train RankNet over the feature set of BM25F and IF

	MAP
BM25F	0.184
BM25F-Rerank-CT	0.215
BM25F-RerankImplicit	0.218
BM25F+Implicit	**0.222**
RN	0.215
RN+All	**0.248**

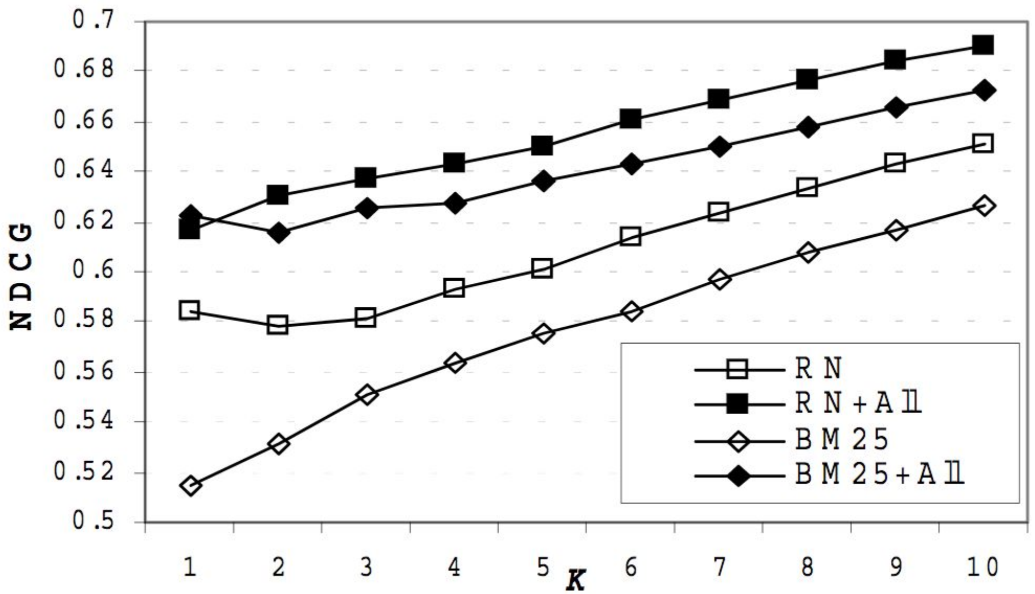


Image source: [8]

Software (actively maintained)

RankLib

sourceforge.net/p/lemur/wiki/RankLib

TF-Ranking

github.com/tensorflow/ranking

(ICTIR/SIGIR 2019 tutorials on the TensorFlow-based toolkit)

LIBLINEAR

www.csie.ntu.edu.tw/~cjlin/liblinear/

XGBoost

https://github.com/dmlc/xgboost

Summary L2R

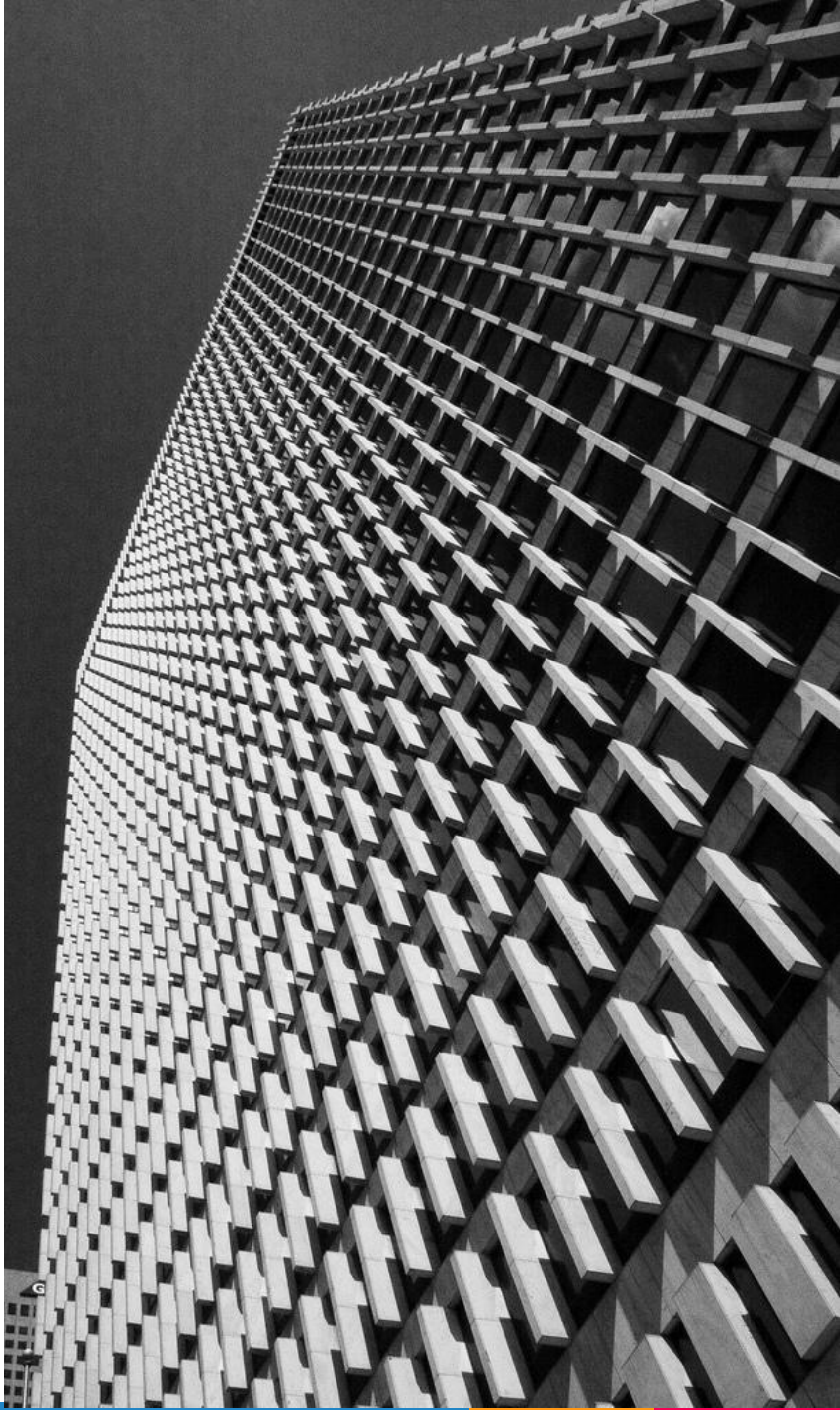
It is not sufficient to deploy standard ML algs; need to be adapted towards ranking (pairwise, listwise)

Listwise vs. pairwise: the former makes most theoretical sense, the latter is empirically robust and efficient

L2R is often neural-network based (shallow, not deep)

Best results achieved in ensembles

Current research: unbiased L2R, efficient & scalable L2R, direct metric optimization



Neural IR

[10] Mitra, Bhaskar, and Nick Craswell. "An introduction to neural information retrieval." Foundations and Trends® in Information Retrieval 13.1 (2018): 1-126.

Ian Goodfellow @goodfellow_ian · 26 Mar 2018

Reviewers should be very suspicious of anyone who has implemented their own baseline. There are a lot of subtle ways to screw up deep learning algorithms and authors have an incentive not to check their own baseline very carefully.

1 6 18

Ian Goodfellow
@goodfellow_ian

Following

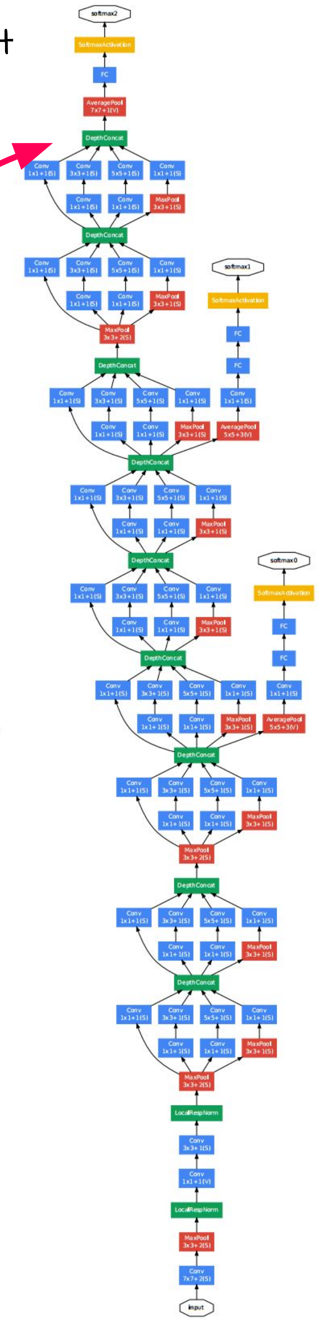
Usually, at least one of the baselines should be a result published in another paper, where the authors of that other paper had some incentive to get a good result. This way the evaluations are at least incentive-compatible.

8:34 PM - 26 Mar 2018

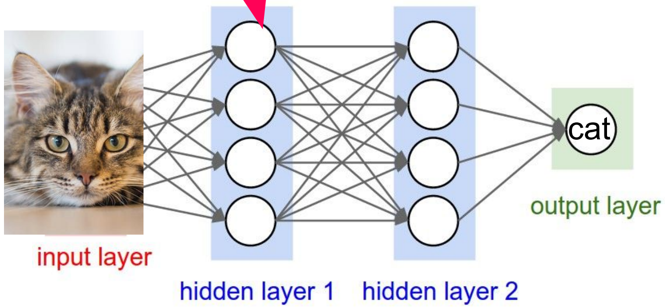
Different fields have different takes on reproducibility …

Overview

GoogLeNet
(2014)



Neural IR is the application of shallow or deep neural networks to IR tasks.



#params
LM, BM25

Neural IR models contain thousands/millions/billions of params.
(large-scale training data needed)

Instead of hand-crafting **features** (classic ML), we now handcraft NN **architectures** and search

[11] Szegedy, Christian, et al. "Going deeper with convolutions." CVPR. 2015.
[12] `http://cs231n.github.io/neural-networks-1/`

Image sources: [11,12]

Neural net basics

Feed-forward neural net with one hidden layer.

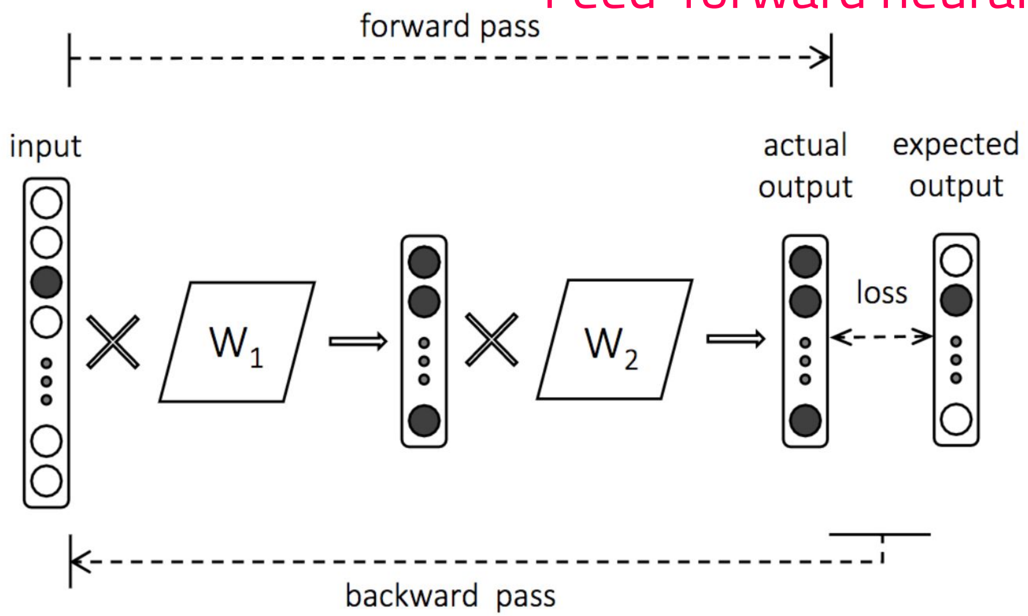
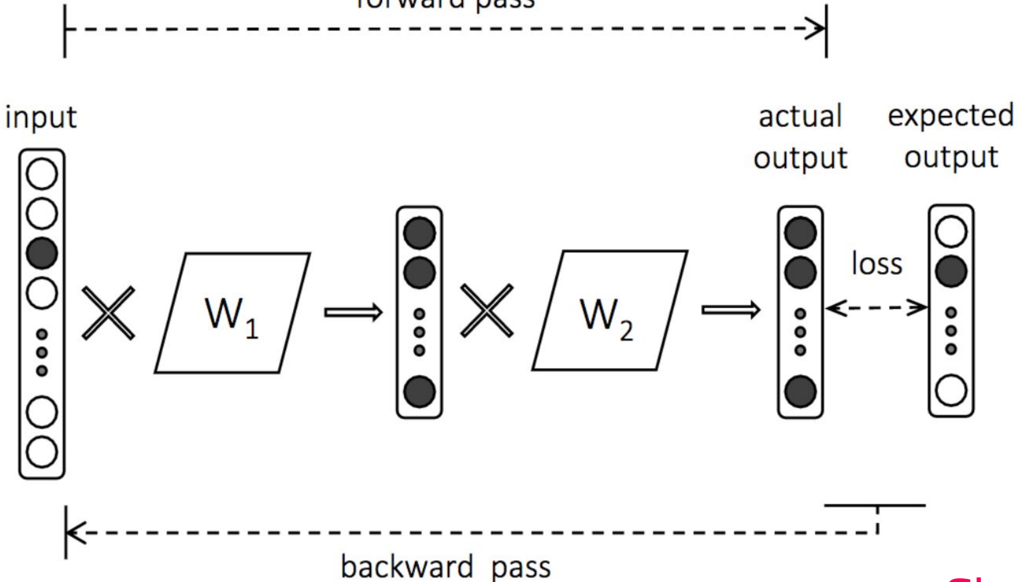


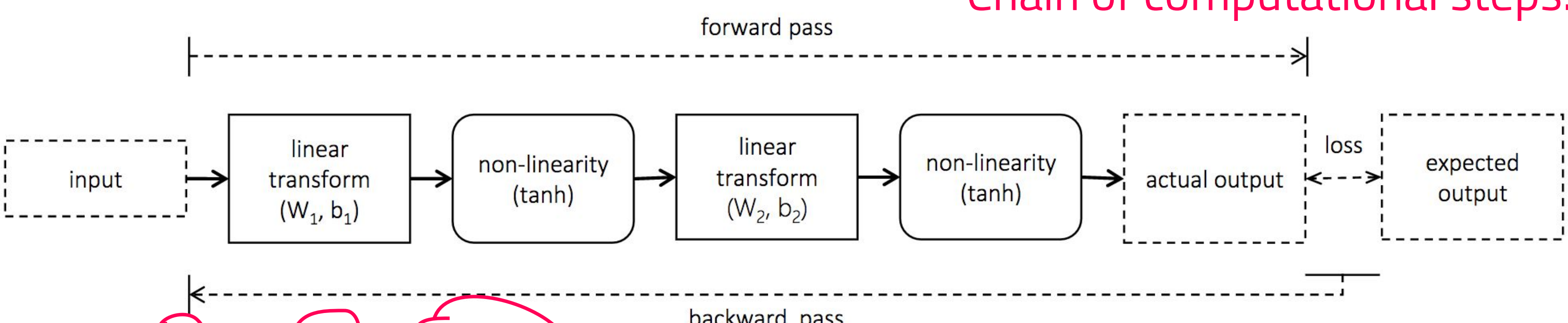
Image source: [10]

Neural net basics

Feed-forward neural net with one hidden layer.



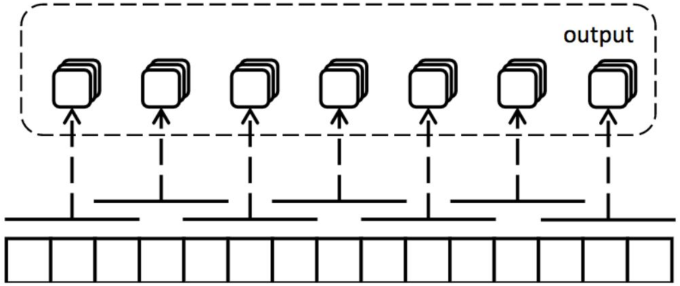
Chain of computational steps.

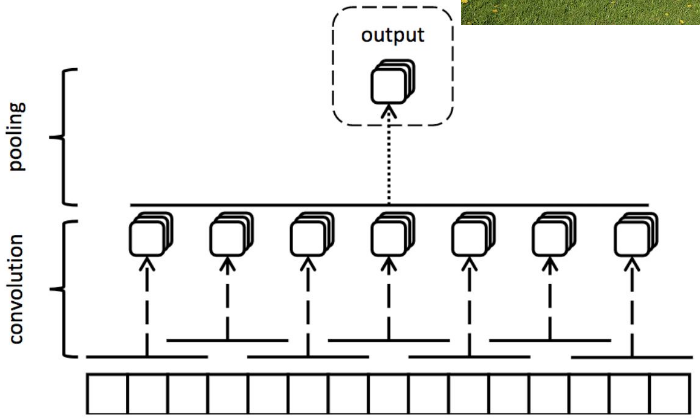


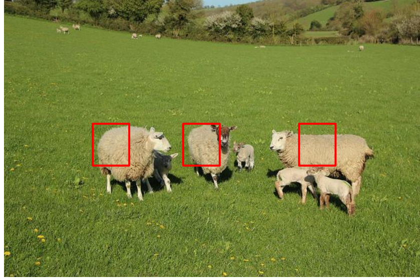
$$\vec{y} = tanh(W_2 \cdot tanh(W_1 \cdot \vec{x} + \vec{b}_1) + \vec{b}_2)$$

gradient error with respect to its parameters

CNNs and RNNs



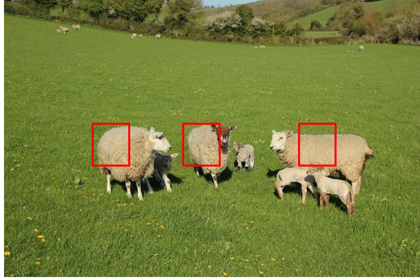


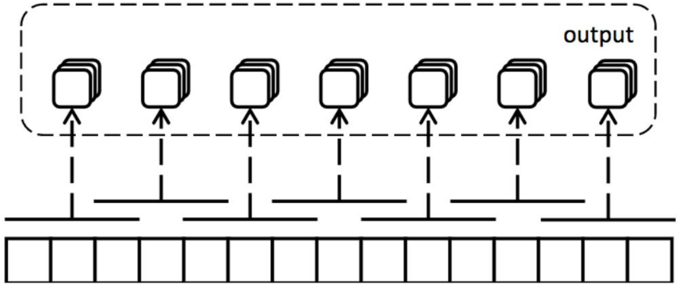


CNN: convolutional neural network.
Most often found in computer vision setups.

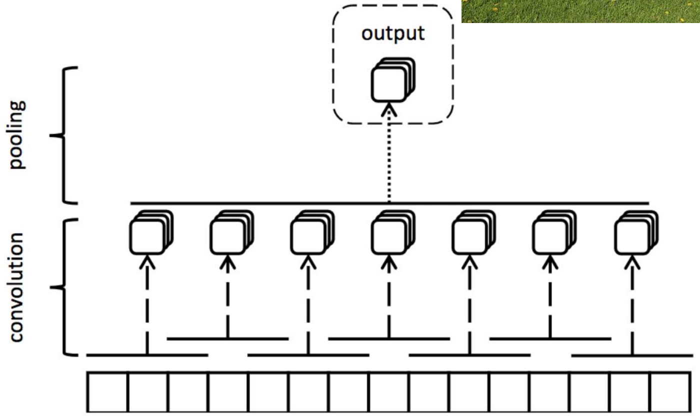
Image source: [10]

CNNs and RNNs

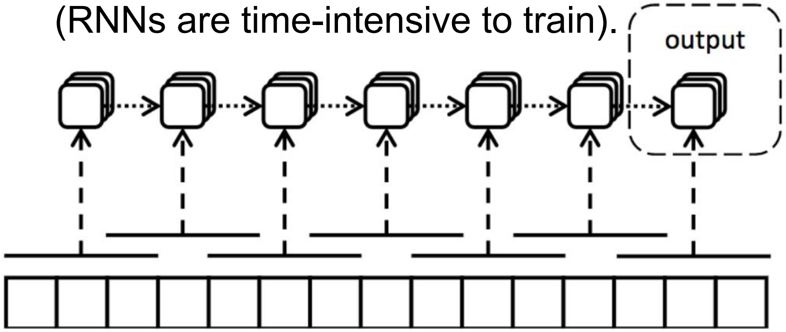




CNN: convolutional neural network.
Standard in computer vision setups.



RNN: recurrent neural network.
Often found in NLP setups, but alternatives are continuously explored (RNNs are time-intensive to train).



Obama entered the White House. He met there with Trump.

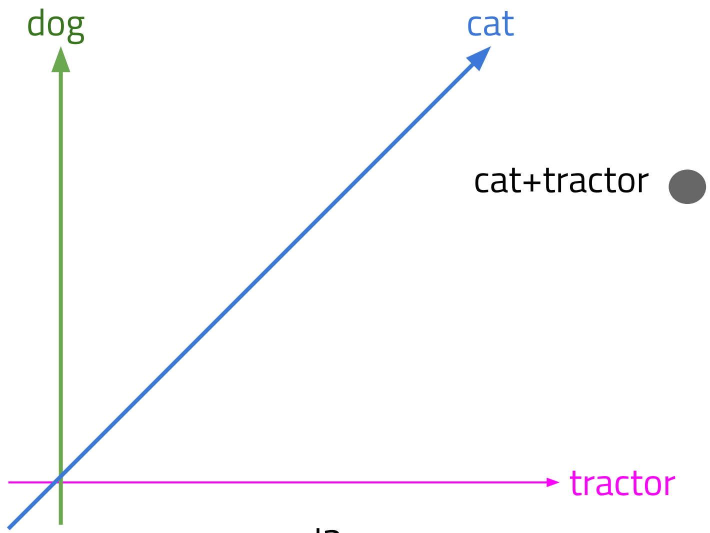
All shift-invariant neural operations, including convolutional and recurrent layers move a fixed size window over the input space with fixed stride.

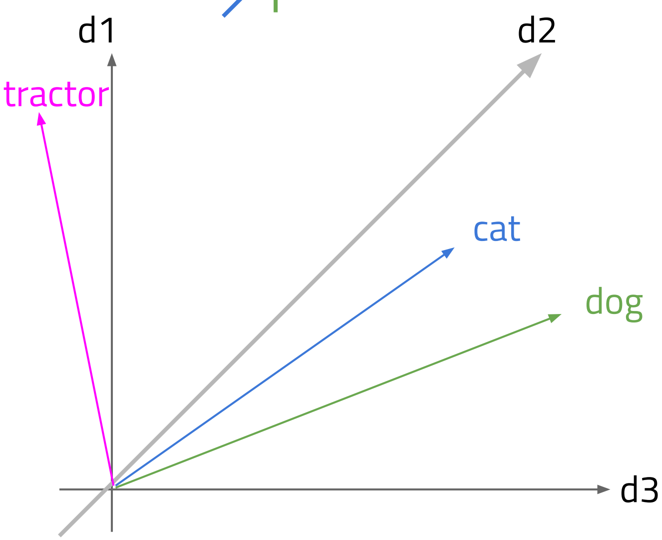
Each window is projected by a parameterized neural operation, often followed by an aggregation step such as (max) pooling.

Image source: [10]

Text representations

Text representations





In DL, commonly known as "one-hot" encoding/repr.

Local vector representation (sparse, high-dimensional)

$$cat = (0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ..., 0)$$

$$tractor = (\quad 0)$$

Individual dimensions are atomic and interpretable.

$$dog = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ..., 0)$$

$$feline = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, ..., 0)$$

Semantic gap: the similarity between any of those two terms is zero.

Handcrafted or learnt. Individual dimensions no longer interpretable.

distributed vector representation (dense, often low-dimensional)

$$cat = (0, 1, 1, 0, 1, 0, 0, 0, 0, 0)$$

$$tractor = (0, 0, 0, 1, 0, 0, 0, 1, 1, 0)$$

$$dog = (1, 1, 0, 0, 0, 0, 0, 0, 0, 0)$$

$$feline = (0, 1, 1, 1, 1, 0, 0, 0, 0, 1)$$

The similarity between some of those terms is no longer zero.

Text representations

Text representations can be learnt in a **supervised** or **unsupervised** fashion.

In IR, supervised approaches use query-document pairs.

In IR, the unsupervised approach uses just queries or just documents.

"**Similarity**" is not an absolute concept, it depends on the task & context at hand. In IR, it is related to relevance.

In DL, commonly known as "one-hot" encoding/repr.

Local vector representation (sparse, high-dimensional)

$$cat = (0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ..., 0)$$

$$tractor = (\quad 0)$$

Individual dimensions are atomic and interpretable.

$$dog = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, ..., 0)$$

$$feline = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, ..., 0)$$

Semantic gap: the similarity between any of those two terms is zero.

Handcrafted or learnt. Individual dimensions no longer interpretable.

distributed vector representation (dense, often low-dimensional)

$$cat = (0, 1, 1, 0, 1, 0, 0, 0, 0, 0)$$

$$tractor = (0, 0, 0, 1, 0, 0, 0, 1, 1, 0)$$

$$dog = (1, 1, 0, 0, 0, 0, 0, 0, 0, 0)$$

$$feline = (0, 1, 1, 1, 1, 0, 0, 0, 0, 1)$$

The similarity between some of those terms is no longer zero.

Feature-based representation examples

in-document features

$$cat = (0, 0, \underset{doc\ 3}{\underset{\downarrow}{1}}, 0, \underset{doc\ 5}{\underset{\downarrow}{1}}, 0, 0, 0, 1, 0, ..., \underset{doc\ 1678}{\underset{\downarrow}{1}}, 0)$$

neighbouring-word features

$$cat = (0, 0, \underset{runs}{\underset{\downarrow}{1}}, 0, \underset{eats}{\underset{\downarrow}{1}}, 0, 0, 0, \underset{hurt}{\underset{\downarrow}{1}}, 0, ..., \underset{is}{\underset{\downarrow}{1}}, 0)$$

neighbouring-word with distances features

$$cat = (0, 0, \underset{runs^{+1}}{\underset{\downarrow}{1}}, 0, \underset{black^{-1}}{\underset{\downarrow}{1}}, 0, 0, 0, \underset{along^{+2}}{\underset{\downarrow}{1}}, 0, ..., \underset{old^{-1}}{\underset{\downarrow}{1}}, 0)$$

character trigram features

$$kitten = (0, 0, \underset{en\#}{\underset{\downarrow}{1}}, 0, \underset{\#ki}{\underset{\downarrow}{1}}, 0, \underset{tte}{\underset{\downarrow}{1}}, 0, \underset{kit}{\underset{\downarrow}{1}}, 0, \underset{ten}{\underset{\downarrow}{1}}, ..., \underset{itt}{\underset{\downarrow}{1}}, 0)$$

Text input to deep neural nets

term-level input with
bag-of-chars per term

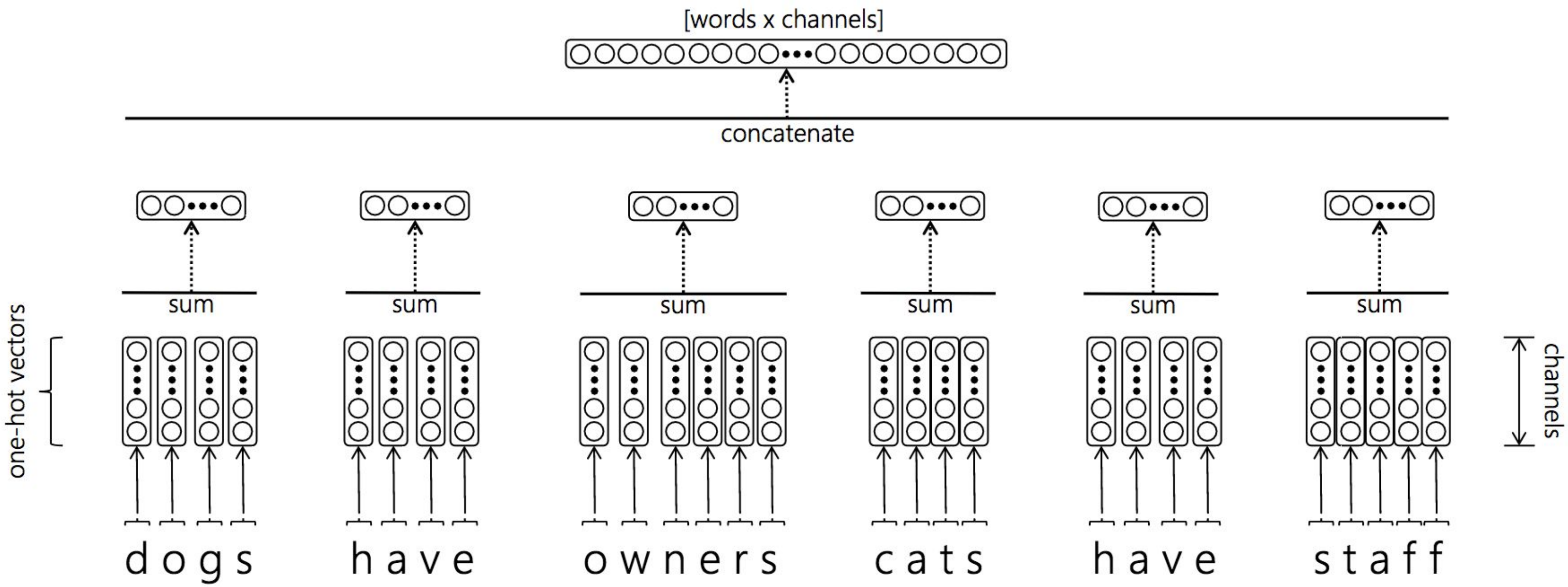


Image source: [10]

Text input to deep neural nets

term-level input with
bag-of-trigrams per term

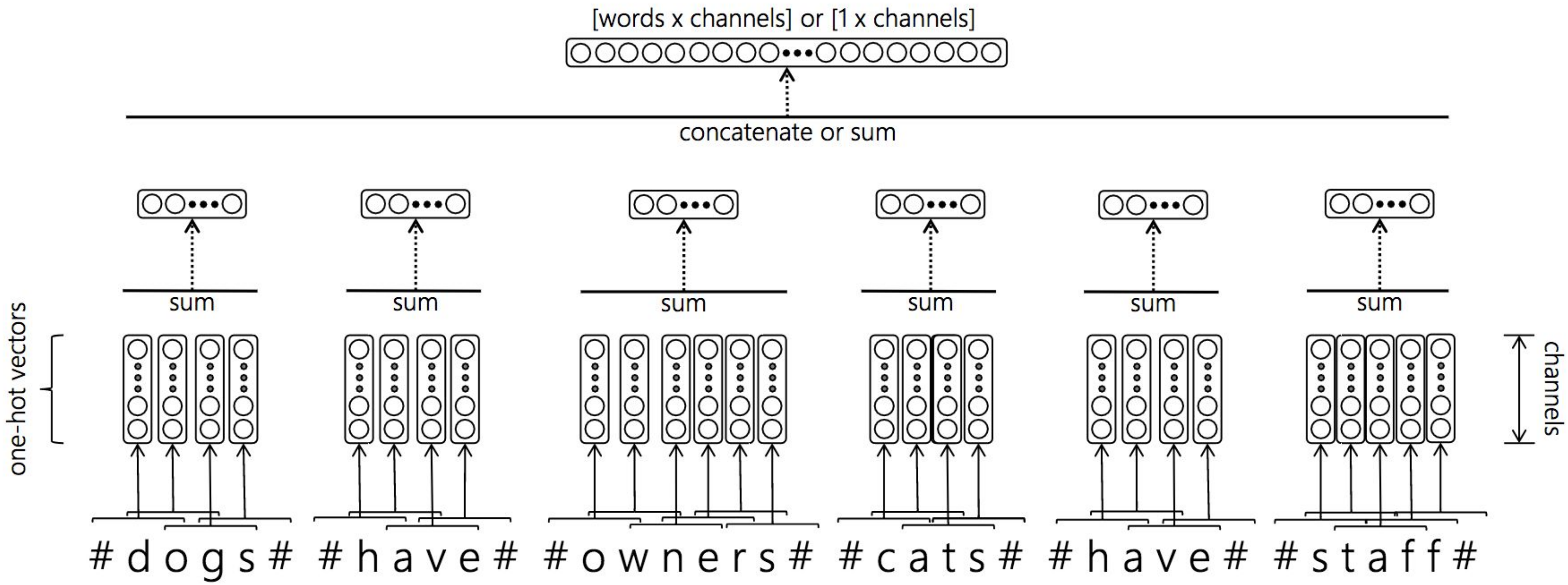
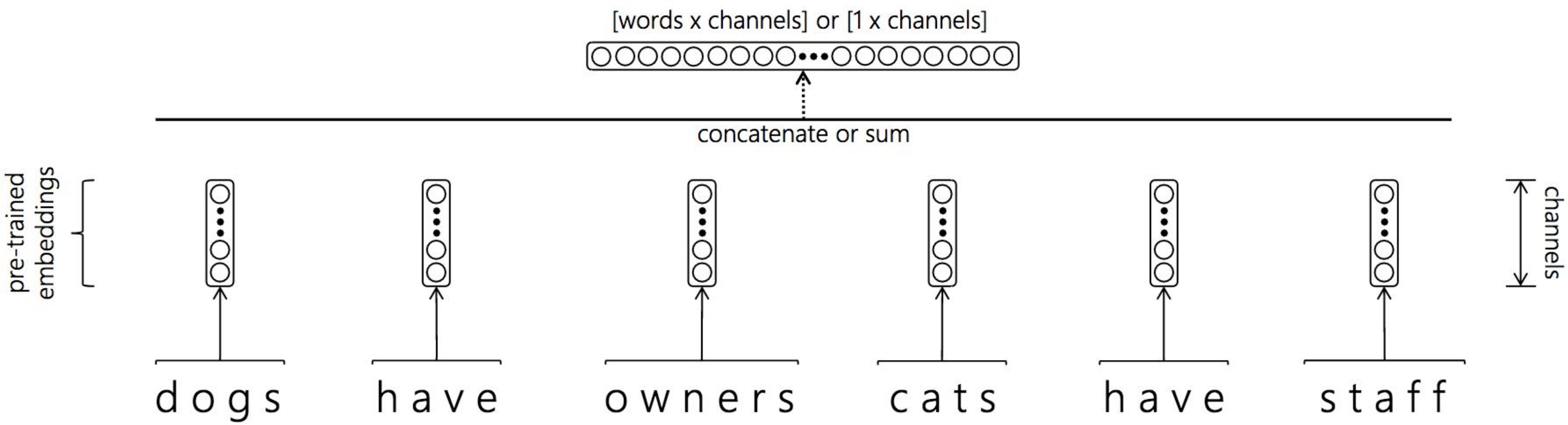


Image source: [10]

Image source: [10]

Text input to deep neural nets

term-level input with
pre-trained word embeddings



- Pre-trained word vectors. This data is made available under the Public Domain Dedication and License v1.0 whose full text can be found at: http://www.opendatacommons.org/licenses/pddl/1.0/.
  - Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): glove.6B.zip
  - Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): glove.42B.300d.zip
  - Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): glove.840B.300d.zip
  - Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, & 200d vectors, 1.42 GB download): glove.twitter.27B.zip

Embeddings

`word2vec/GloVe`: the vector of a word should be similar to the vectors of its neighbouring words

Context-free: one embedding per word

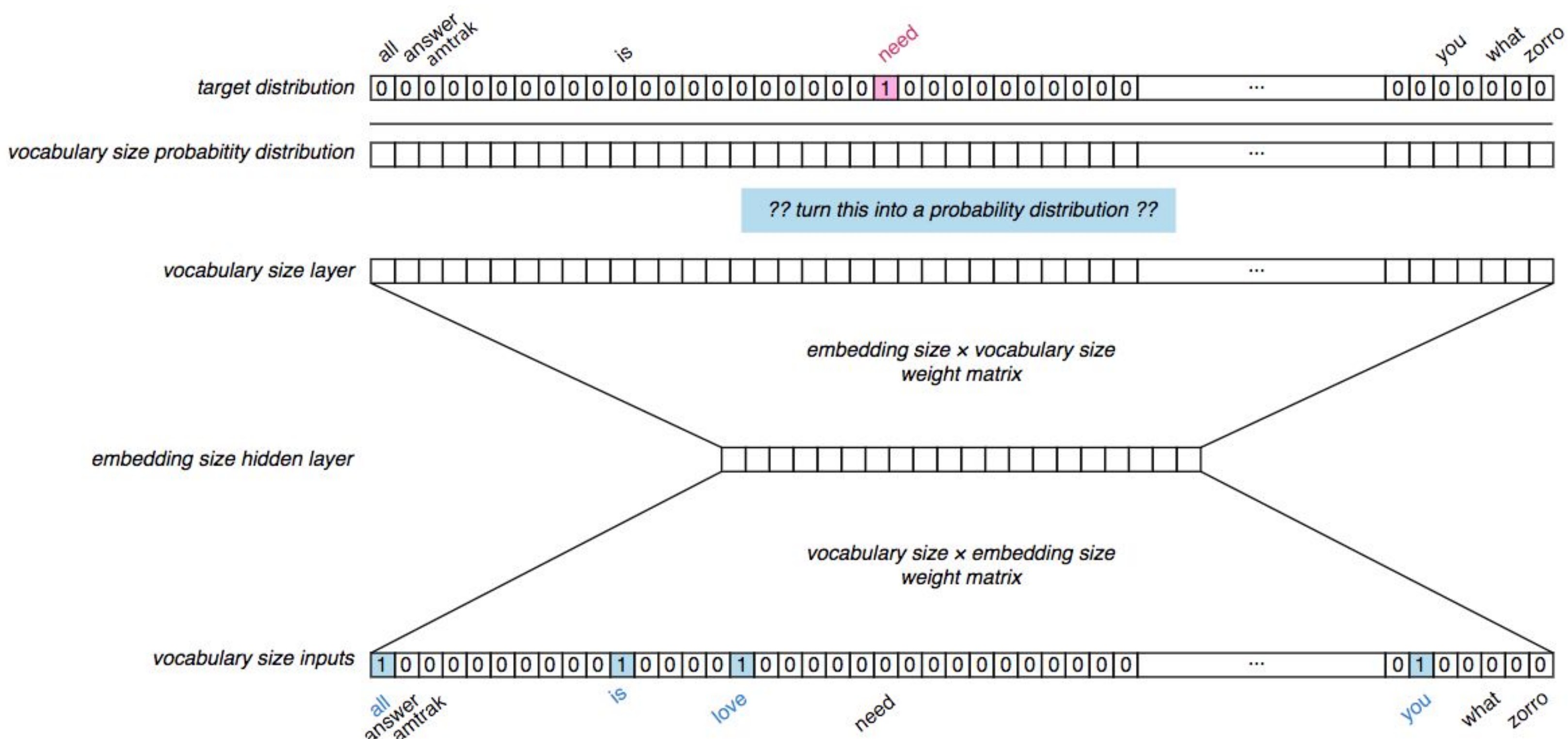


Image source: [10]

… representation in a new space that should preserve the relationships between items of the original representation.

Embeddings

Dense vector representation.

Low-dimensional.

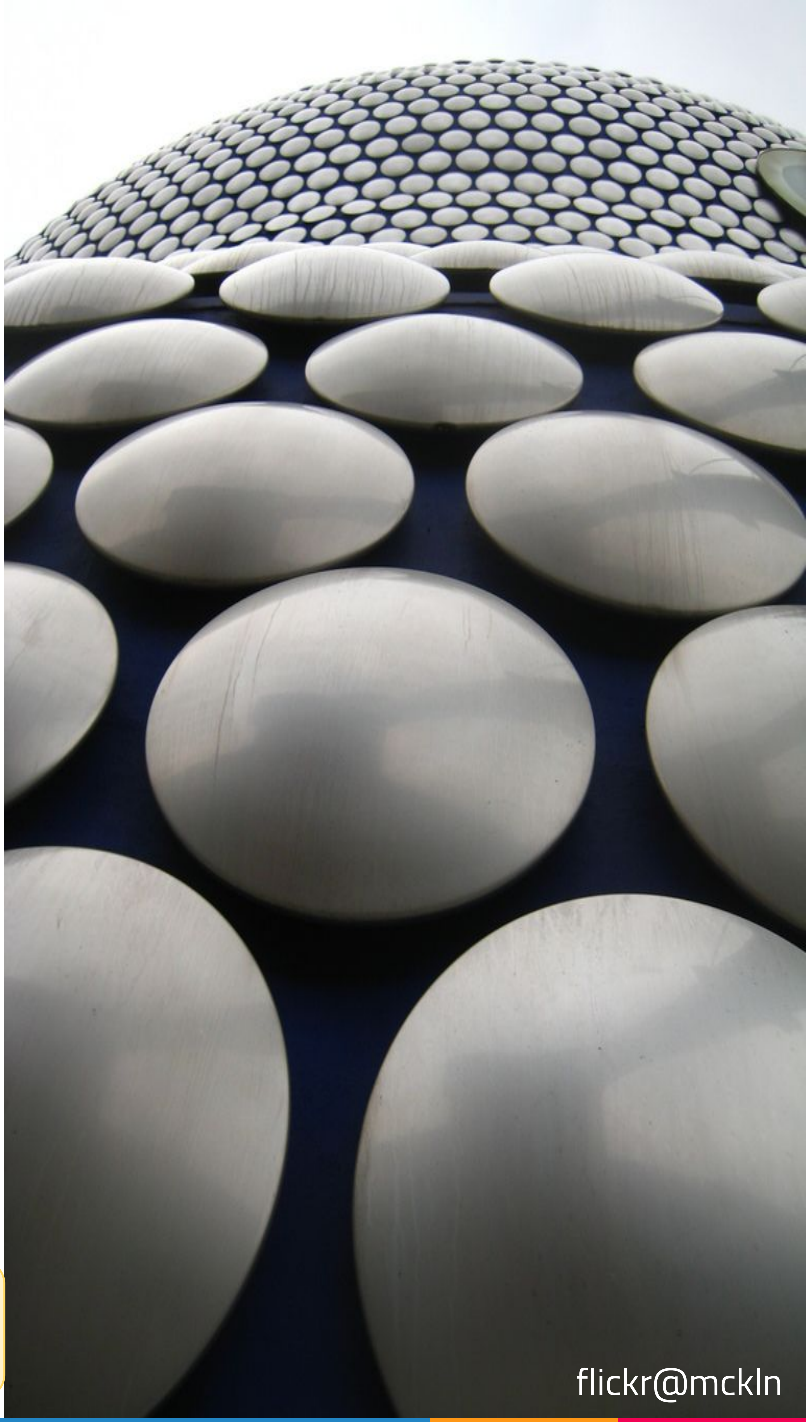
Learnt from data.

Distributed representation most often refers to learnt embeddings.

In today’s NLP literature, the default encoding.

Motivated by the **distributional hypothesis.**

This is an old idea! LSI was one of the first practical ‘implementations’ of it (1988). Did not work well.



Neural IR architectures

Categorized according to query representation, document representation and relevance estimation.

A good retrieval system should …

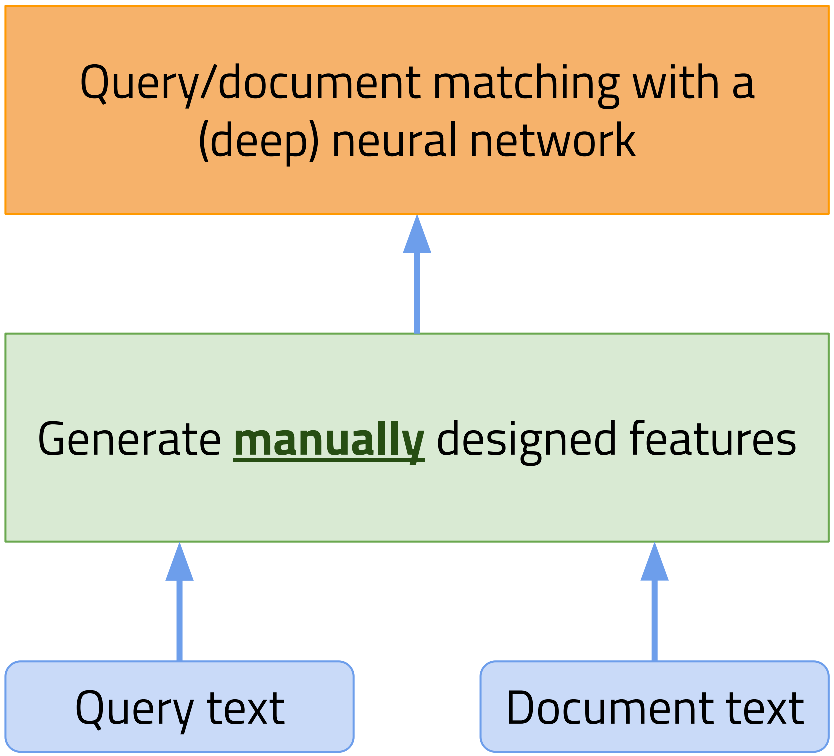
1. Have **semantic understanding** and enable **exact term matching**

2. Be **robust** to rare **inputs**: remember the long tail

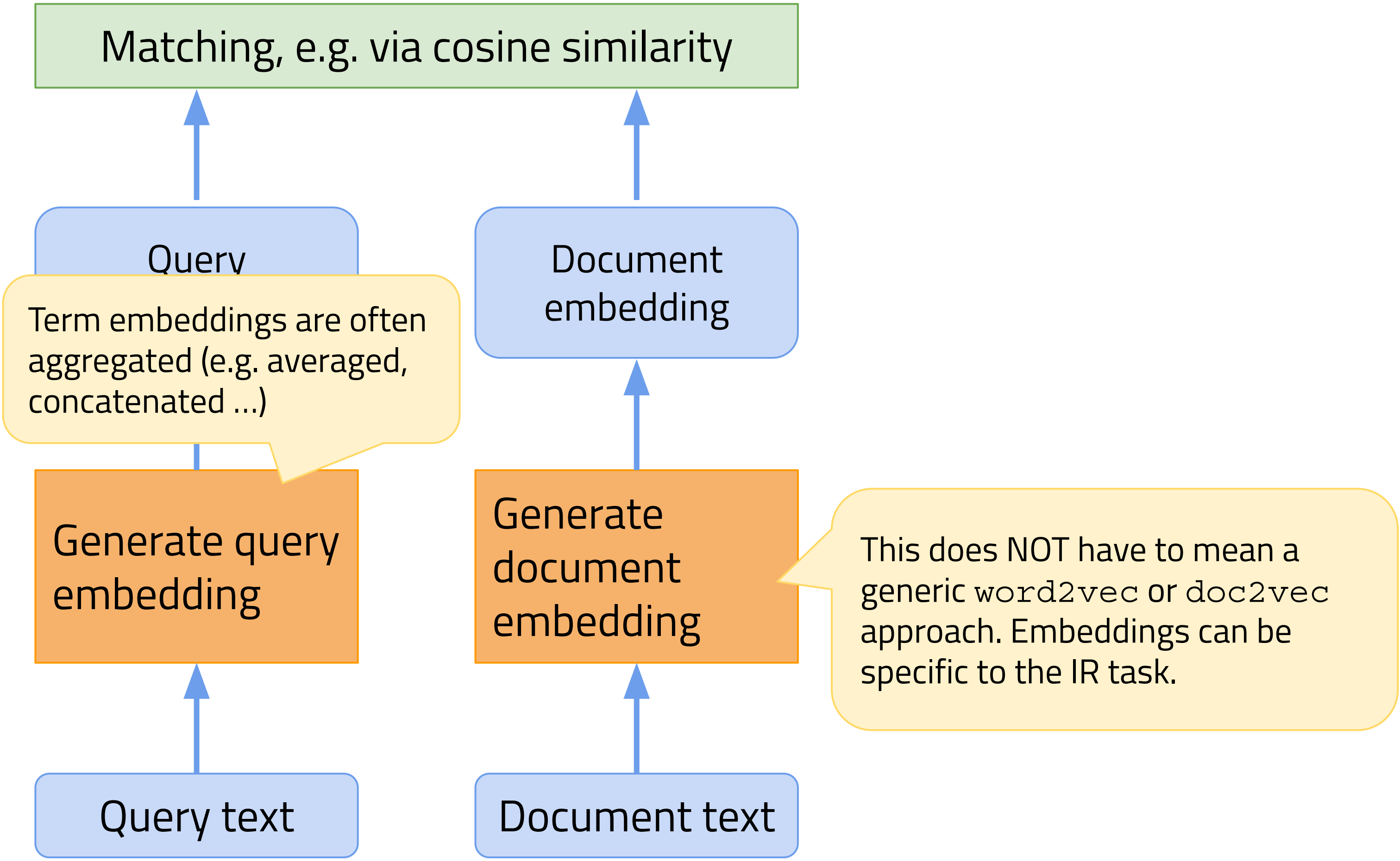
3. Be **robust** to **corpus variance**

4. Be **robust** to **variable length input**

Let's start with L2R's architecture



Representation focused models

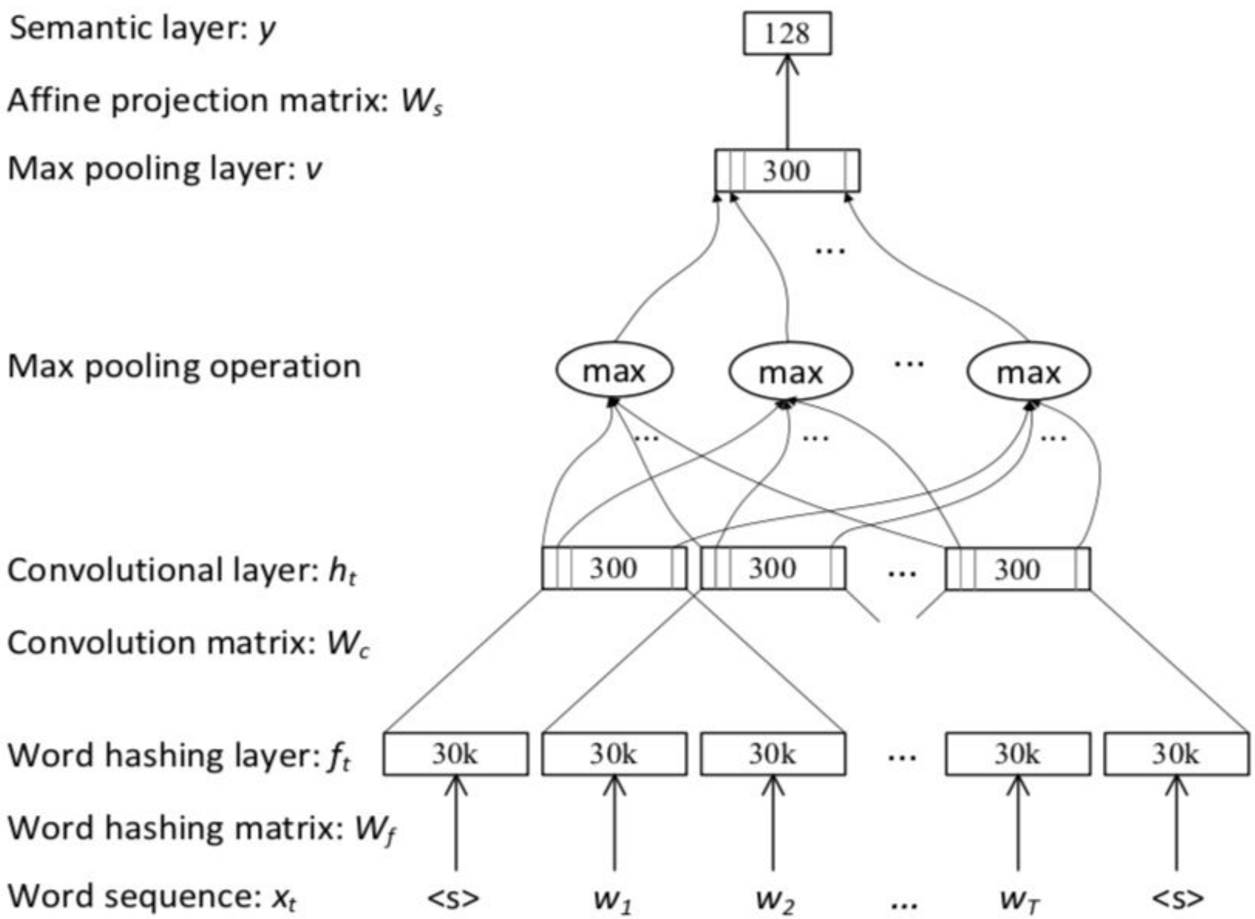


Example: C-DSSM

Convolutional Deep Structured Semantic Model

	Models	NDCG@1
1	BM25	0,305
5	DSSM	0,320
6	**C-DSSM**	**0,342**

(30M query-clicked titles instances for training; document titles only)



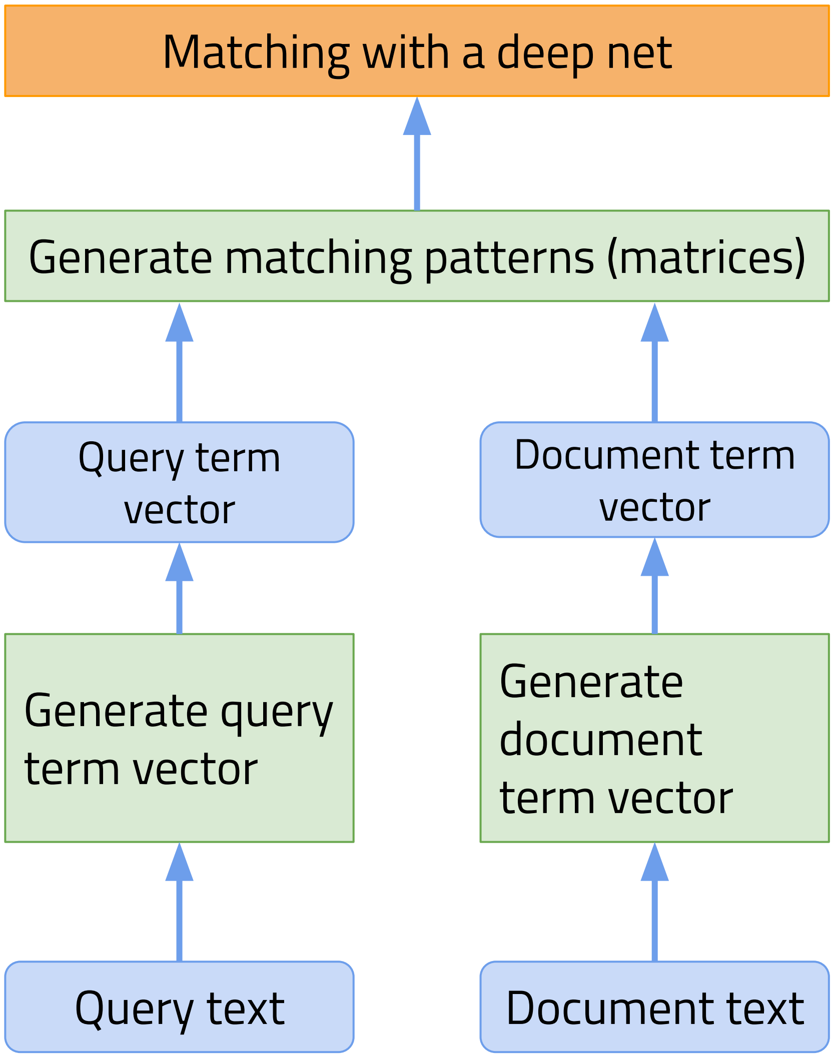
Query-doc similarity based on cosine similarity

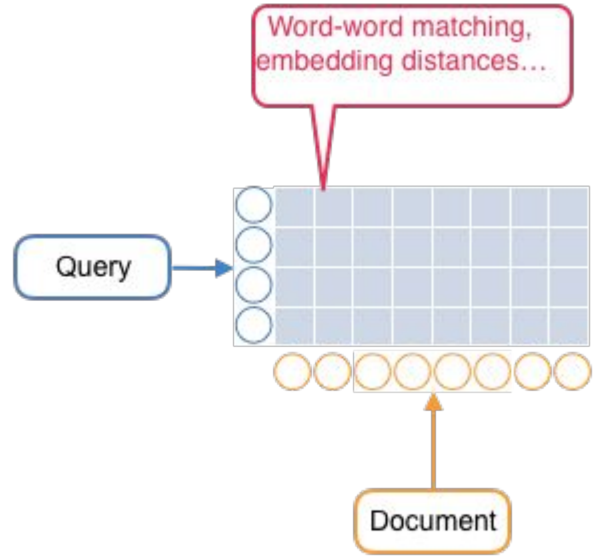
char trigram representation

[13] Shen, Yelong, et al. "Learning semantic representations using convolutional neural networks for web search." WWW. ACM, 2014.

Image source: [13]

Interaction focused models



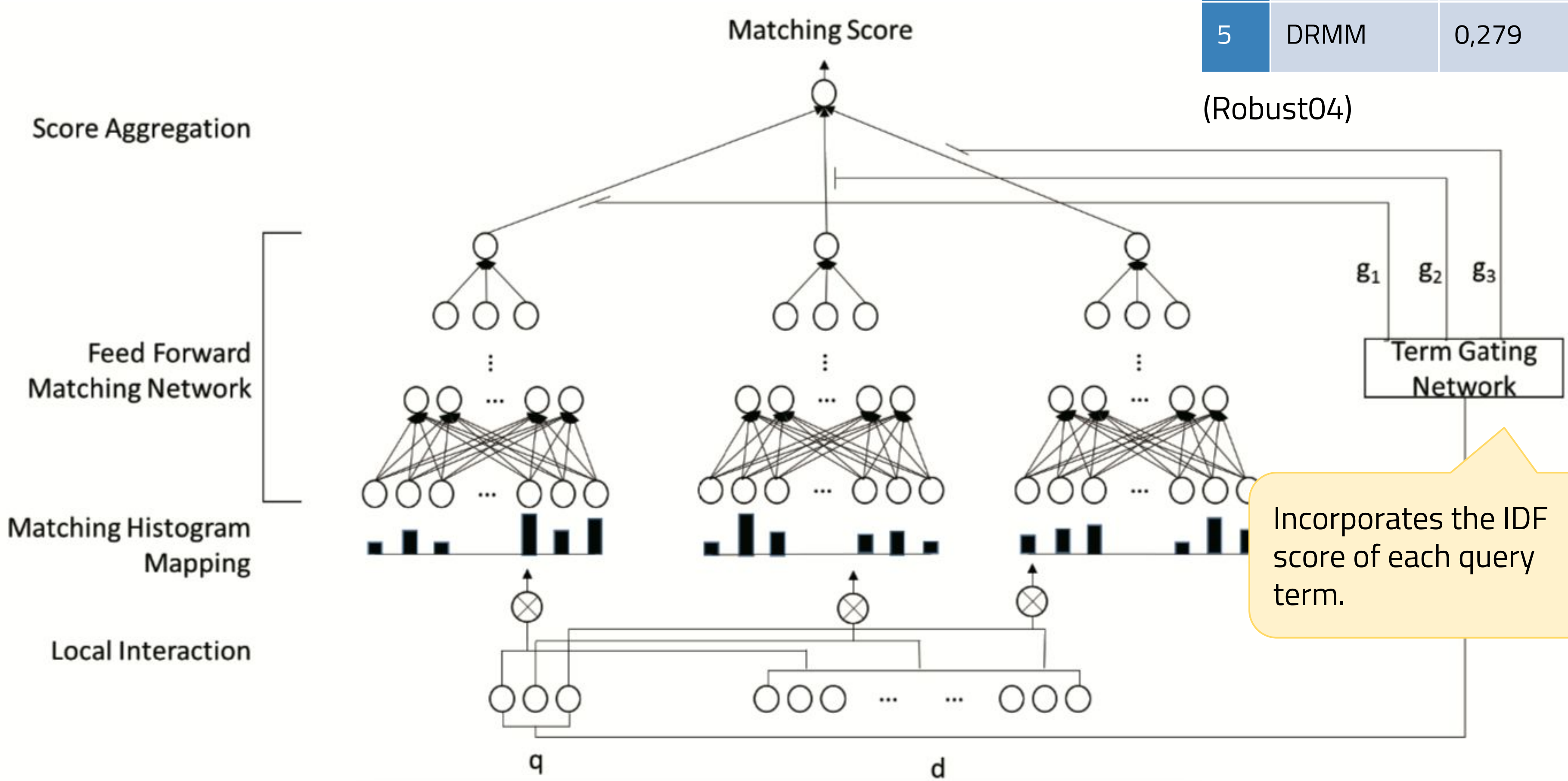


Example: DRMM

Deep Relevance Matching Model

	Models	MAP
1	QL	0,253
2	BM25	0,255
3	CDSSM	0,067
4	ARC-II	0,067
5	DRMM	0,279

(Robust04)



Incorporates the IDF score of each query term.

[14] Guo, Jiafeng, et al. "A deep relevance matching model for ad-hoc retrieval." CIKM. ACM, 2016.

Image source: [14]

joint training

Hybrid example: Duet

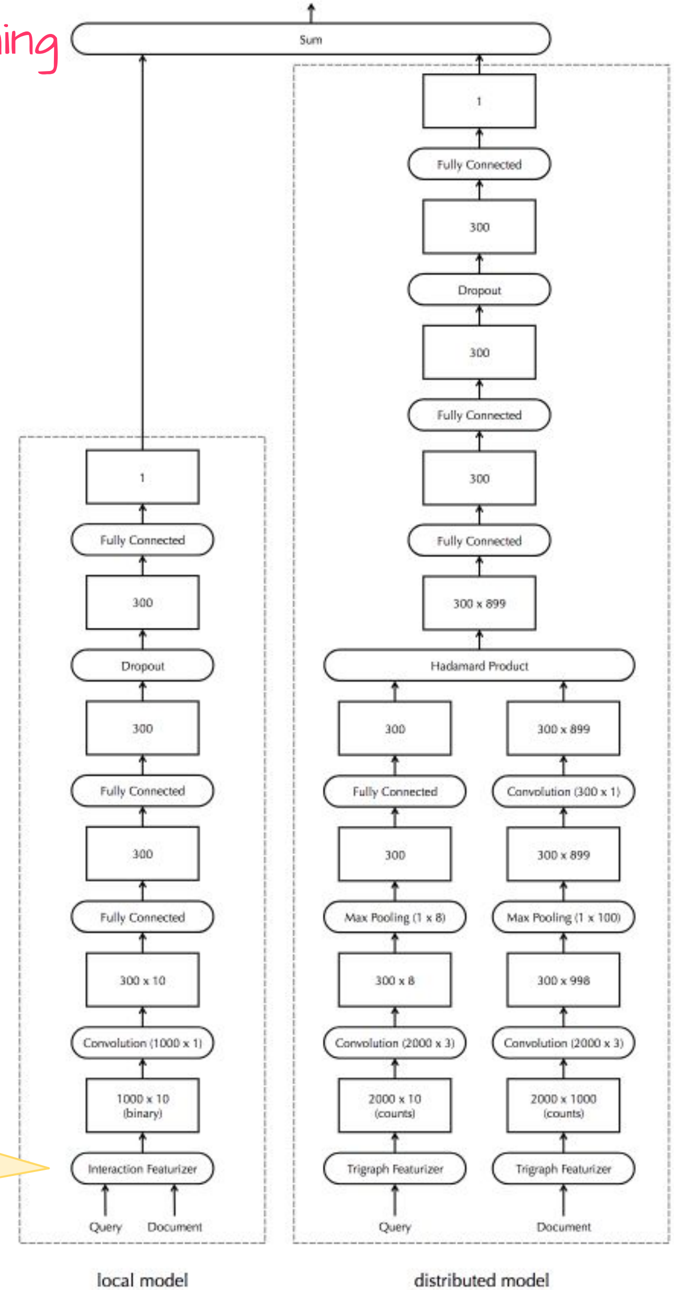
$$f(\mathbf{Q},\mathbf{D}) = f_{local}(\mathbf{Q},\mathbf{D}) + f_{distr}(\mathbf{Q},\mathbf{D})$$

Fixed input length: first 10 terms per query and first 1000 terms per document

Local model: one-hot encoded input

Distributed model: n-grams as input

Patterns of term matches



[15] Mitra, Bhaskar, Fernando Diaz, and Nick Craswell. "Learning to match using local and distributed representations of text for web search." WWW, 2017.

Image source: [15]

	Models	NDCG@1
1	QL	0,246
2	CDSSM	0,273
3	Duet local	0,246
4	Duet distr.	0,286
5	Duet	0,322

Hybrid example: Duet

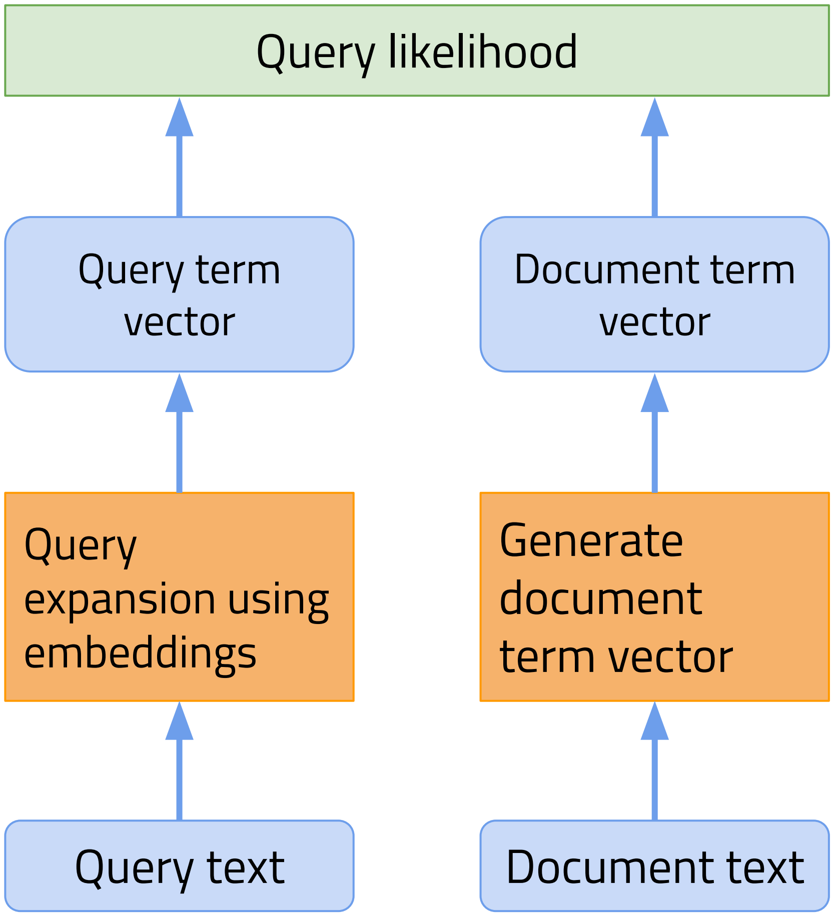
Training data: **200K queries** sampled from Bing's search logs.

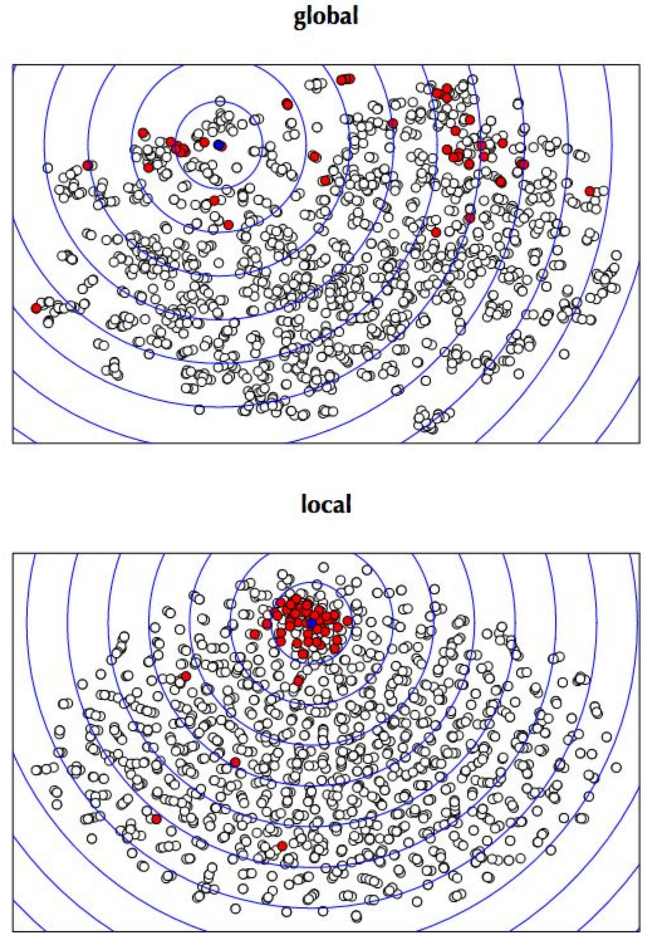
Test data: **8K queries**, with approx. 25 documents per query (**14%** of queries also occurred in the training set)

Four negative documents sampled per relevant document.

Domain knowledge is key: "*For training, we discarded all documents rated as perfect because a large portion of them fall under the navigational intent, which can be better satisfied by historical click based ranking signals.*"

Query expansion





Point: term

Blue: query term

Red: term in the rel. Doc. set

t-SNE projection
(stochastic neighbour embedding)

PRF: learn embeddings on the top retrieved docs. only instead of a large corpus.

[16] Diaz, Fernando, Bhaskar Mitra, and Nick Craswell. "Query expansion with locally-trained word embeddings." ACL. 2016.

Image source: [16]

Insights

Many specific architectures have been proposed, little work on **automatic architecture search**

Interaction-focused techniques tend to outperform representation-based models [17]

Full ranking instead of re-ranking remains challenging (**sparse indices**) .. but BERT!

Relevance matching != semantic matching

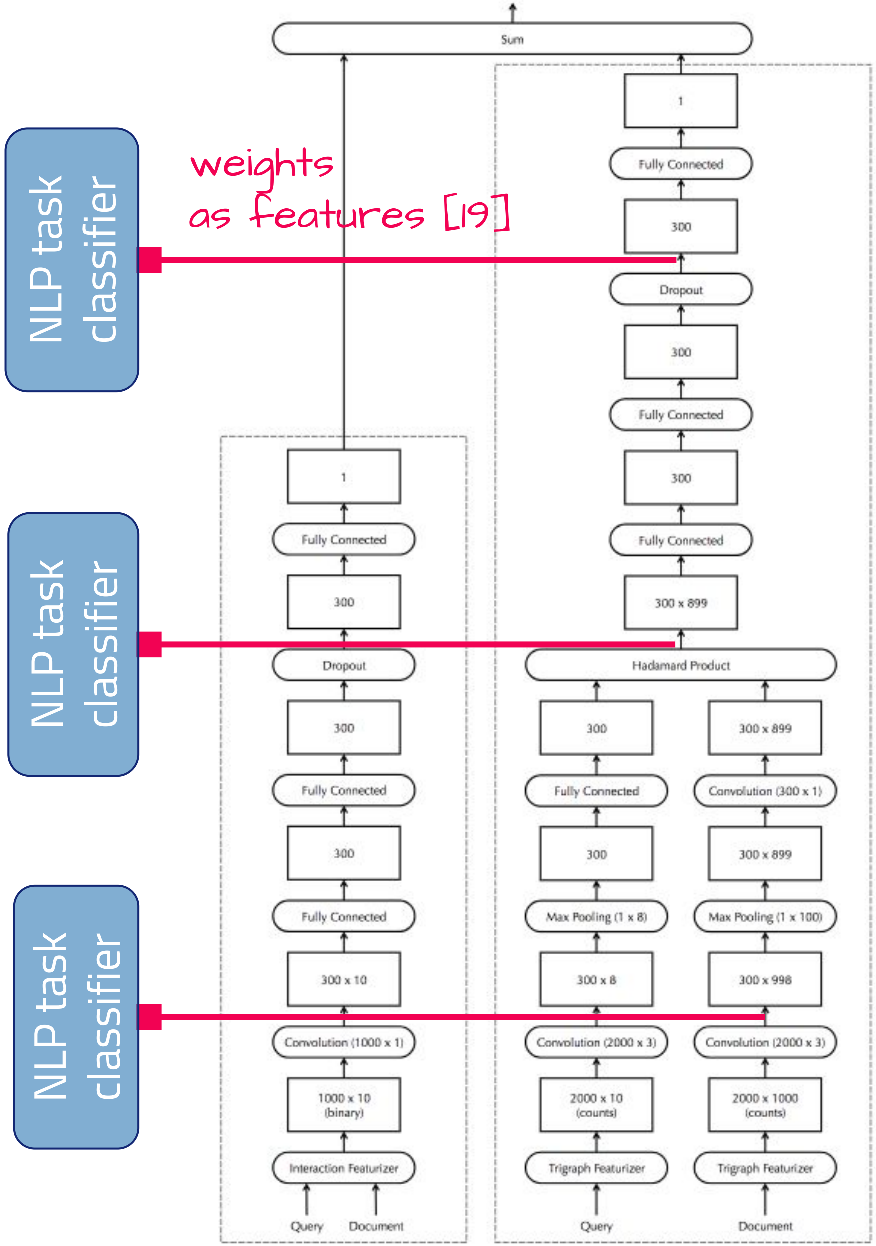
Reproducibility is a big issue (many implementation details to get right)

We need a lot of **training data** - how to get it?

Model analysis is in its infancy

[17] Nie, Yifan, Yanling Li, and Jian-Yun Nie. "Empirical study of multi-level convolution models for ir based on representations and interactions." ICTIR. ACM, 2018.

Model analysis



[18] Rennings, Daniël, Felipe Moraes, and Claudia Hauff. "An Axiomatic Approach to Diagnosing Neural IR Models." ECIR, 2019.
[19] Cohen, Daniel, Brendan O'Connor, and W. Bruce Croft. "Understanding the representational power of neural retrieval models using NLP tasks." ICTIR. ACM, 2018.

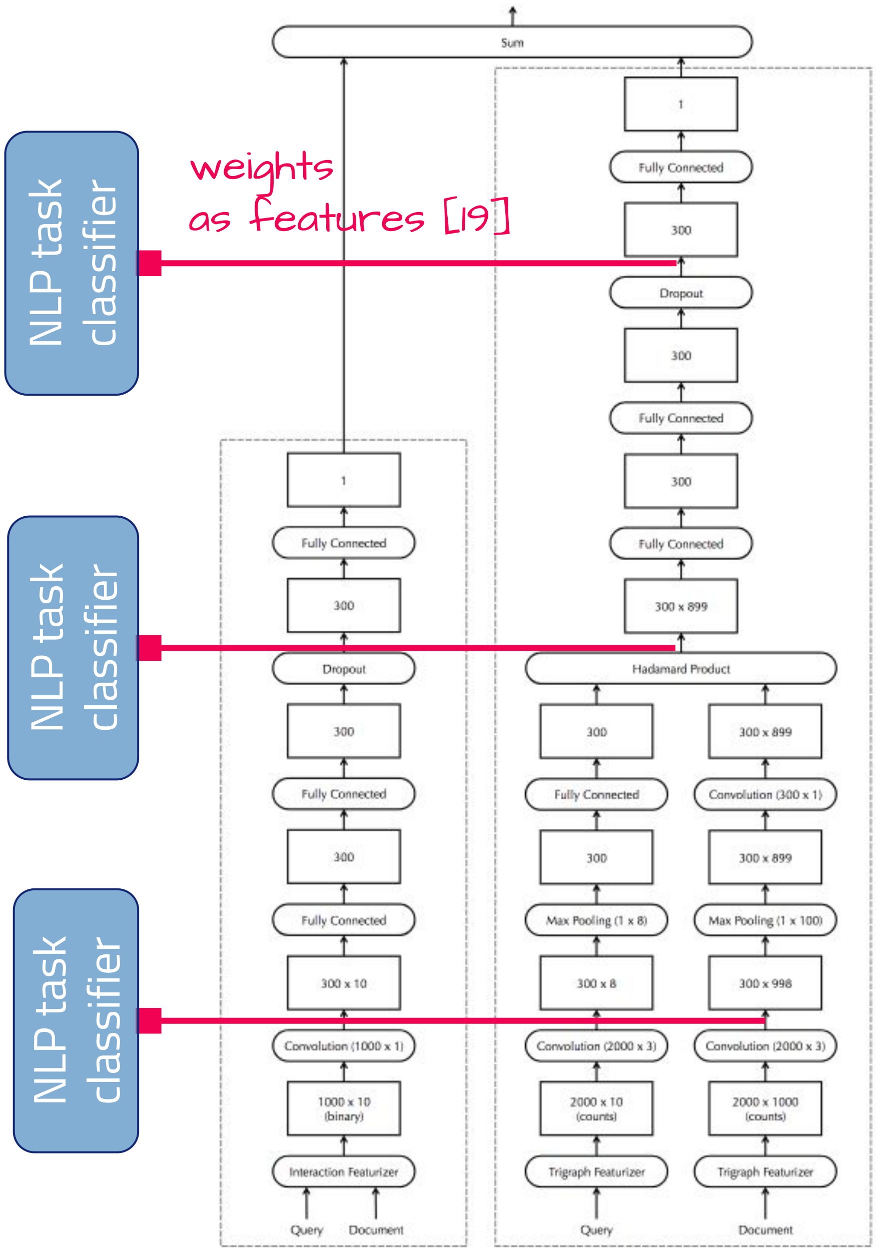
Model analysis

Diagnostic approach [18]:

- Create diagnostic dataset to fulfill an **axiom;** evaluate to what extent neural models realize the patterns

- `TFC1`: given a single-term query w and two equally long documents, the RSV of the document with a higher freq. of w should be higher
- `LNC2`: If a document is replicated k times, its RSV should not be lower than that of its un-replicated variant

- Neural models fare poorly on LNC2



[18] Rennings, Daniël, Felipe Moraes, and Claudia Hauff. "An Axiomatic Approach to Diagnosing Neural IR Models." ECIR, 2019.
[19] Cohen, Daniel, Brendan O'Connor, and W. Bruce Croft. "Understanding the representational power of neural retrieval models using NLP tasks." ICTIR. ACM, 2018.

Issue across IR, not specific to neural models.
Ongoing efforts (e.g. OSIRRC 2019).

Baselines: well-tuned?

Condition	AP	P20
QL	0.2499	0.3556
QL + RM3	0.2865	0.3773
Neural_1	0.2815	0.3752
Neural_2	0.2801	0.3764
Neural_3	0.2856	0.3766
$\text{Neural}_3{}'$	0.2971	0.3948
Anserini: QL	0.2496	0.3543
Anserini: BM25	0.2526	0.3604
Anserini: BM25 + RM3 (independent)	0.2954	0.3885
Anserini: BM25 + RM3 (joint)	0.2973	0.3871

(Robust04)

[20] Lin, Jimmy. "The neural hype and comparisons against weak baselines." ACM SIGIR Forum. Vol. 52. No. 1. ACM, 2019.
OSIRRC @SIGIR 2019: `https://osirrc.github.io/osirrc2019/`

MatchZoo

https://github.com/NTMC-Community/MatchZoo

Used by 3 | Watch 157 | Unstar 2,500 | Fork 680

DRMM | MatchPyramid | ARC-I/-II | aNMM

DSSM | C-DSSM | Duet | ...

Thanks **@bwanglzu** for pointing out the differences of the implementations in MatchZoo with the descriptions in some papers. Yes, this is possible. We tried to implement the most important components/novel parts of these neural models. But it is still possible that there are some differences between our current implementation details with some details described in the paper. For some critical differences, we will fix them in the next version of MatchZoo. But for some differences like "dropout", I think it is fine to keep it. You can adjust the dropout rate to control the network as you want. What do you think about it ? **@faneshion @pl8787 @bwanglzu**

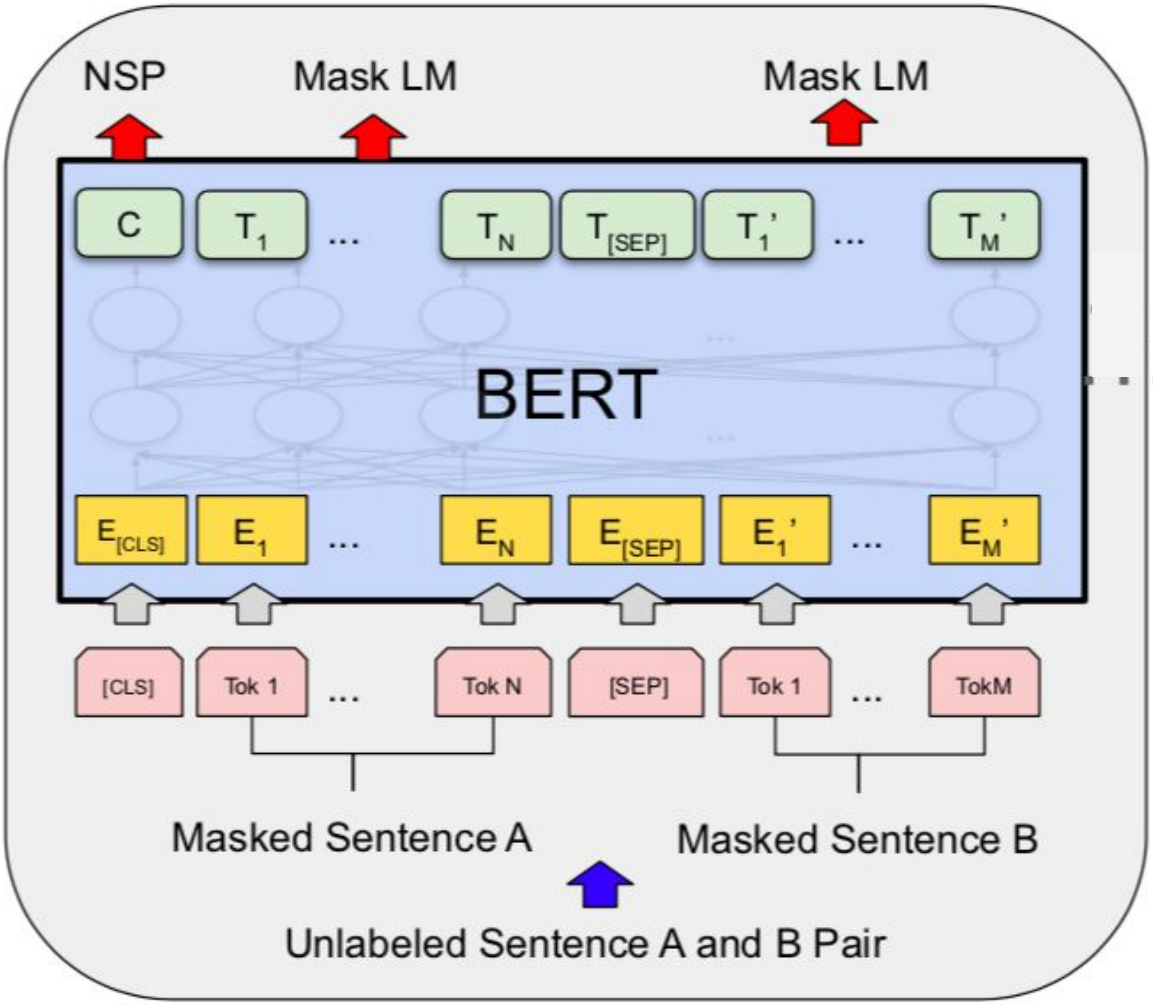
https//github.com/NTMC-Community/MatchZoo/issues/99

Finally ... BERT

Bidirectional (contextual) Encoder Representations from Transformers (a popular attention model)

[21] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT: pre-training



Pre-training based on:

1. Masked language model (15% tokens)
2. Next sentence prediction

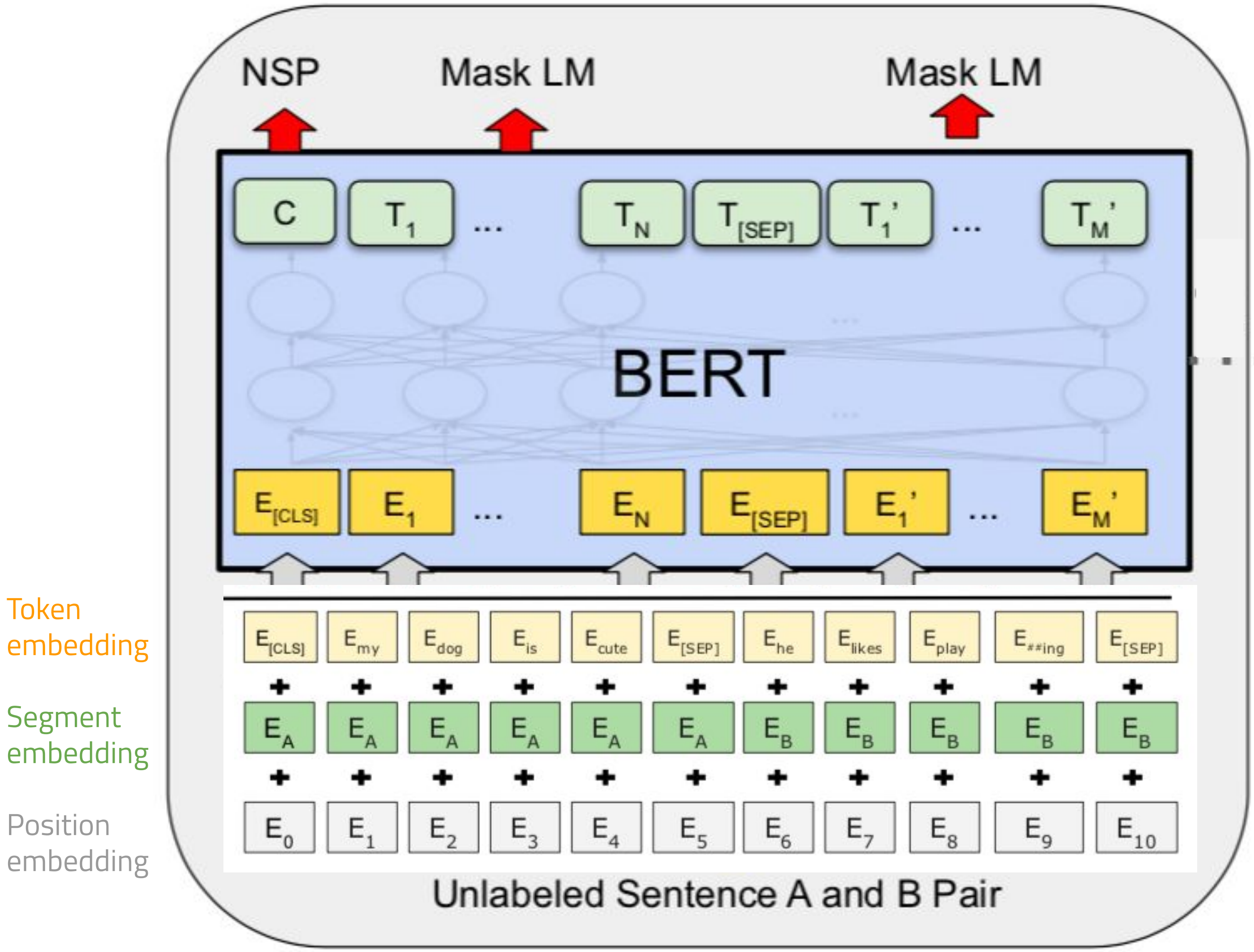
(i.e. *self-supervised*)

340M parameters

Finding: BERT reduces the need for task-specific architectures (fine-tuning of the pre-trained model is enough)

Image source: [21]

BERT: pre-training



Pre-training based on:

1. Masked language model (15% tokens)
2. Next sentence prediction

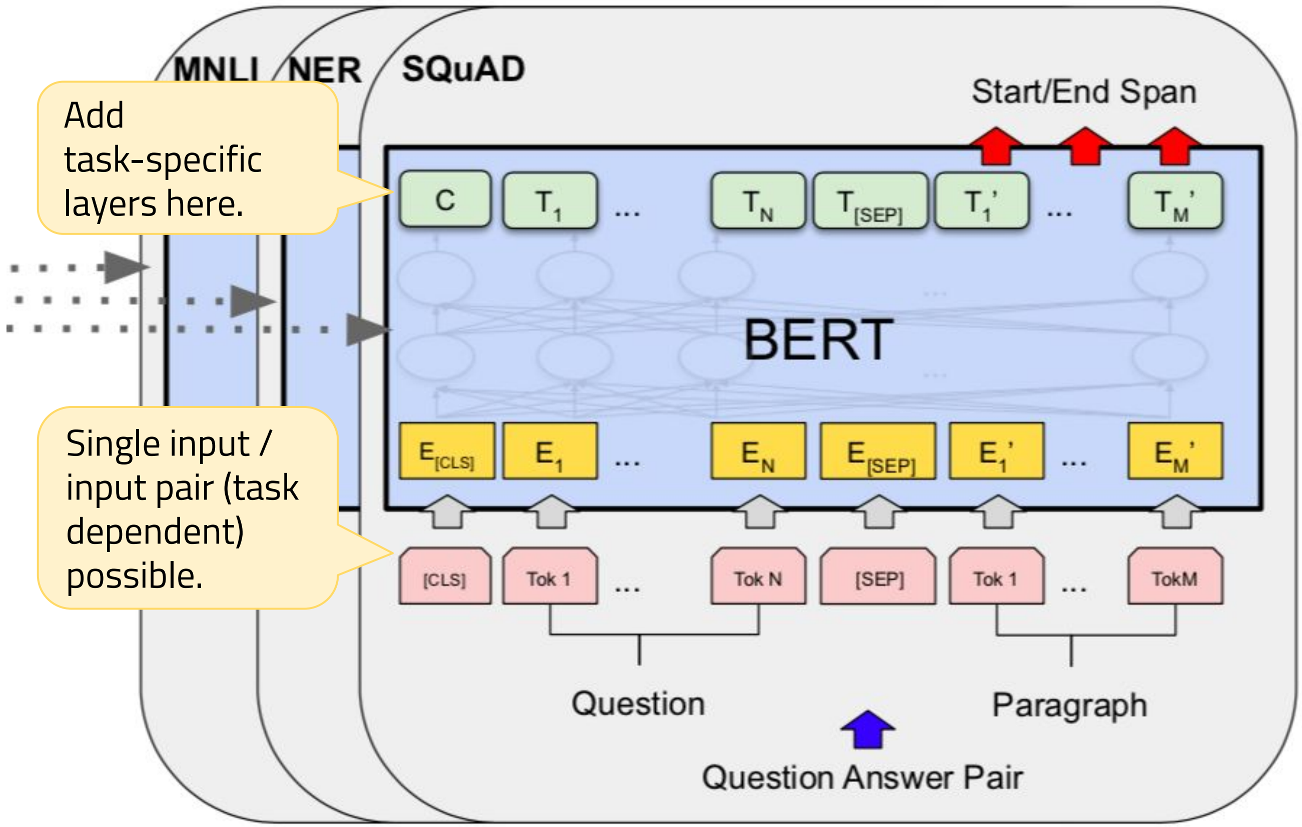
(i.e. *self-supervised*)

340M parameters

Finding: BERT reduces the need for task-specific architectures (fine-tuning of the pre-trained model is enough)

Image source: [21]

BERT: fine-tuning



1. Initialized with pre-trained parameters
2. Adapt the final layer in a task-specific manner.
3. Fine-tune with labelled data specific to your task

Image source: [21]

Passage re-ranking with BERT

MSMarco (1M+ queries):

1. Retrieve top-k passages with BM25 (computationally inexpensive)

2. Rerank top-k passages based on fine-tuned BERT

Fine-tuning:

- Query: “Sentence A” (max. 64 tokens)
- Passage: “Sentence B” (query+passage+[SEP] max. 512 tokens)
- Output: probability of relevance
- Loss function: cross-entropy loss

[22] Nogueira, Rodrigo, and Kyunghyun Cho. "Passage Re-ranking with BERT." arXiv preprint arXiv:1901.04085 (2019).

Passage re-ranking with BERT

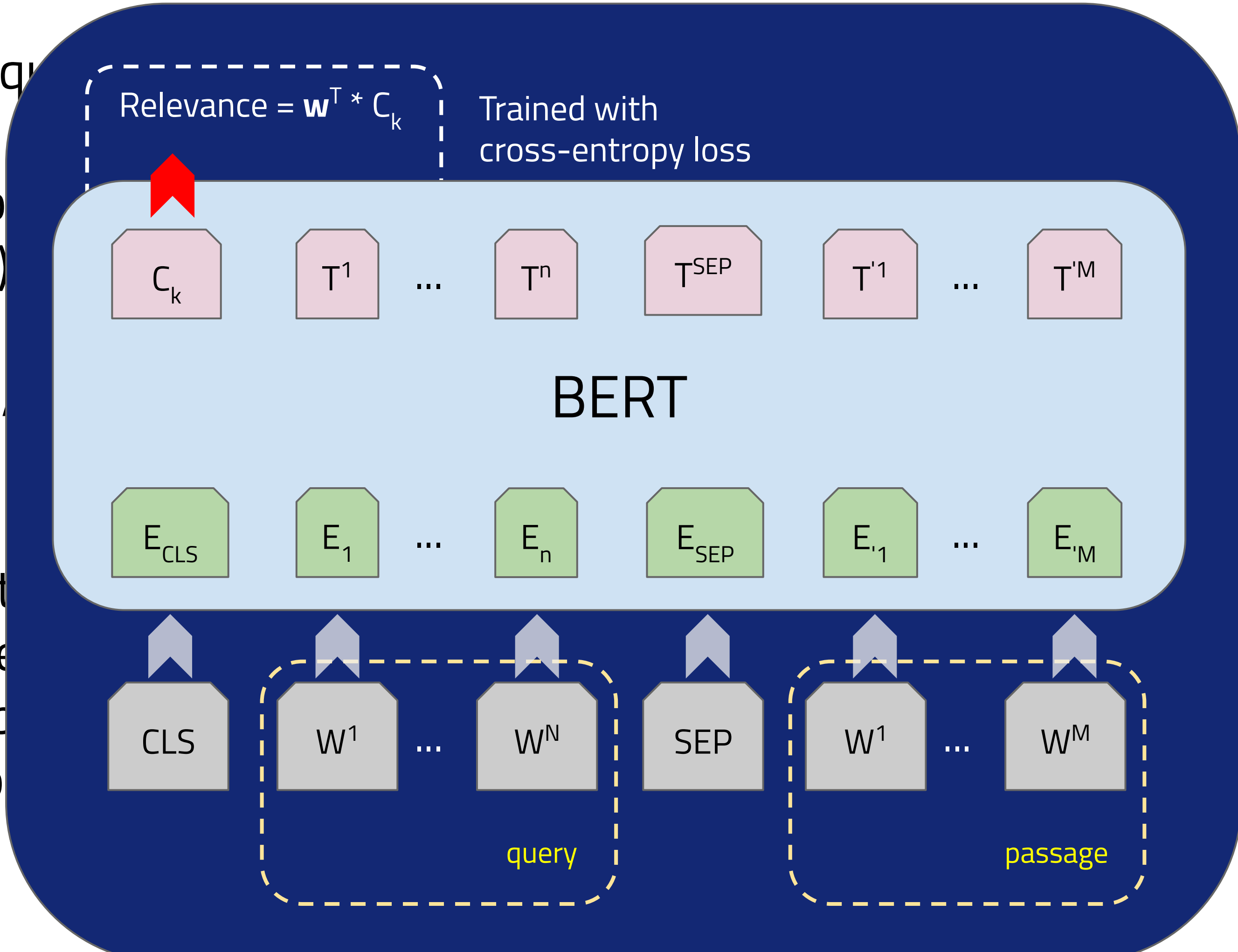
MSMarco (1M+ q

1. Retrieve top
   inexpensive)

2. Rerank top-

Fine-tuning:

- Query: “Sent
- Passage: “Se
- Output: prob
- Loss functio



[22] Nogueira, Rodrigo, and Kyunghyun Cho. "Passage Re-ranking with BERT." arXiv preprint arXiv:1901.04085 (2019).

Passage re-ranking with BERT

MSMarco (1M+ queries):

1. Retrieve top-k passages with BM25 (computationally inexpensive)

2. Rerank top-k passages based on fine-tuned BERT

Fine-tuning:
- Query: “Sentence A” (max. 64 tokens)
- Passage: “Sentence B” (query+passage+[SEP] max. 512 tokens)
- Output: probability of relevance
- Loss function: cross-entropy loss

Result: **27%** relative improvement in MRR over previous SOTA!

[22] Nogueira, Rodrigo, and Kyunghyun Cho. "Passage Re-ranking with BERT." arXiv preprint arXiv:1901.04085 (2019).

BERT for ad-hoc retrieval (newswire)

Insight: document relevance can be approximated by the “best” sentence or paragraph in a document

Idea: combine **document retrieval score** with **sentence-level BERT scores** (**sentences ordered by BERT score**); **few hyperparameters**

$$\text{Score}_d = a \cdot S_{\text{doc}} + (1-a) \cdot \sum_{i=1}^{n} w_i \cdot S_i$$

Fine-tune on datasets with sentence-level qrels, e.g. TREC microblog datasets (instead of the target corpus Robust04)

[23] Yang, Wei, Haotian Zhang, and Jimmy Lin. "Simple applications of bert for ad hoc document retrieval." arXiv preprint arXiv:1903.10972 (2019).

BERT for ad-hoc retrieval (newswire)

Insight: document relevance can be approximated by the "best" sentence or paragraph in a document

Idea: combine

scores (**sente**

Robust04

BM25+RM3	0.3033	0.3974
1S: BERT FT(QA)	0.3102	0.4068
2S: BERT FT(QA)	0.3090	0.4064
3S: BERT FT(QA)	0.3090	0.4064
1S: BERT FT(Microblog)	0.3266	0.4245
2S: BERT FT(Microblog)	**0.3278**	0.4267
3S: BERT FT(Microblog)	**0.3278**	**0.4287**

$$\text{Score}_d =$$

Fine-tune on

datasets (ins

[23] Yang, Wei, Haotian Zhang, and Jimmy Lin. "Simple applications of bert for ad hoc document retrieval." arXiv preprint arXiv:1903.10972 (2019).

BERT is not the last word!

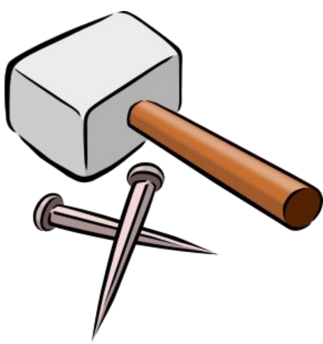
XLNet, RoBERTa, DistilBERT, ...

Very worthwhile tutorial on Writing code for NLP (and IR) research:
http://tiny.cc/pxes9y

However, given the pace at which the area of deep learning is growing [...] we should be wary of the combinatorial explosion of trying every model on every IR task.

We should not disproportionately focus on maximizing quantitative improvements and in the process neglect theoretical understanding and qualitative insights. [...]

Neural models should not be the hammer that we try on every IR task, or we may risk reducing every IR task to a nail.



Mitra, Bhaskar, and Nick Craswell. "An introduction to neural information retrieval." Foundations and Trends® in Information Retrieval 13.1 (2018): 1-126.

flickr@phunk

- ☑ Learning to rank
- ☑ Neural IR

- ☑ Research directions
- ☑ Pitfalls

✉ c.hauff@tudelft.nl

🌐 chauff.github.io