From Exact Matching to Semantic Matching: Using neural models for ranking

Zhenghao Liu

liu-zh16@mails.tsinghua.edu.cn

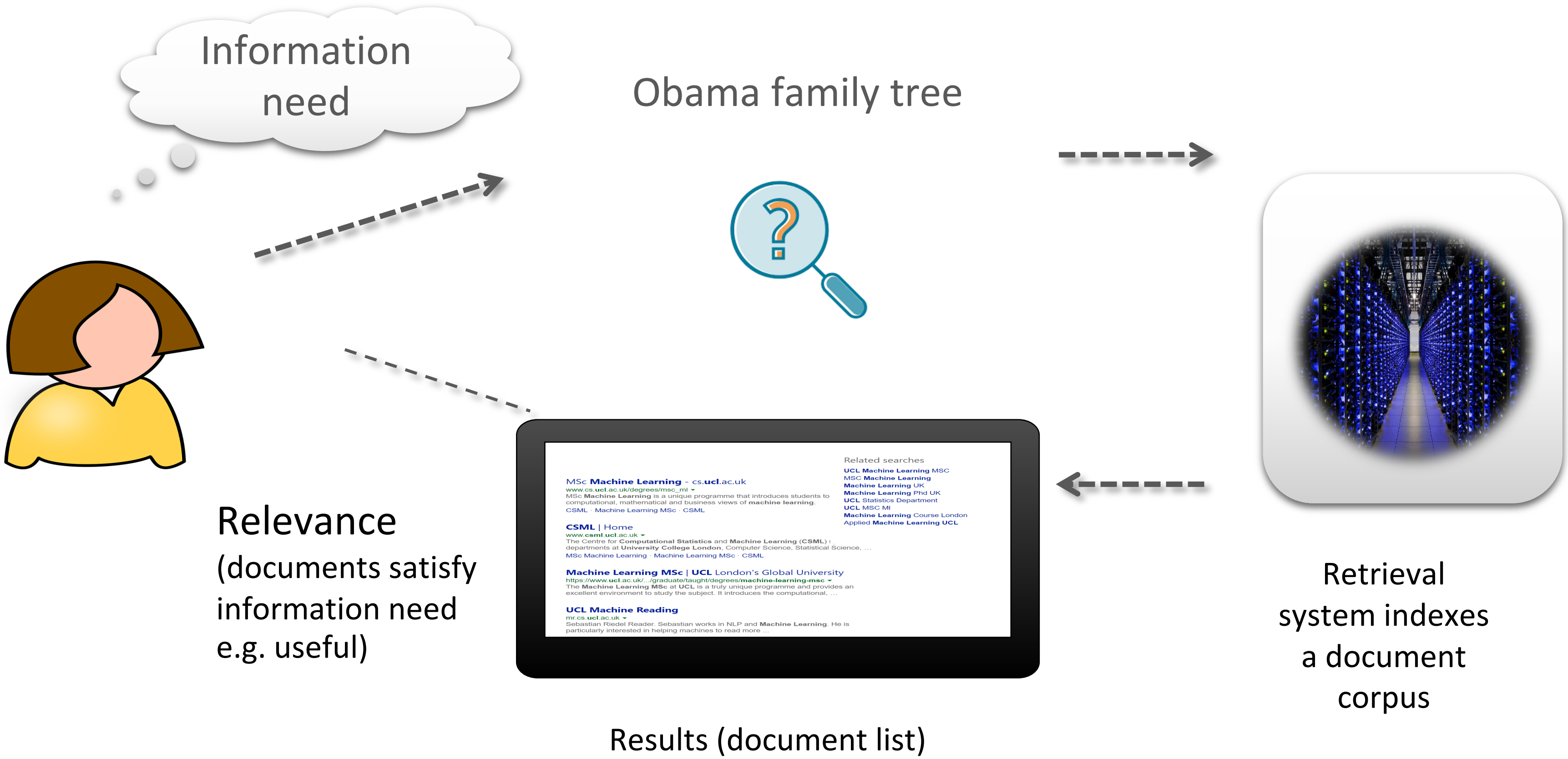
THUNLP

Outline

- Introduction to IR
- Traditional IR Models
- Neural IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

Information Retrieval

- What is Information Retrieval (IR)



Relevance between text queries and documents

Information Retrieval

- Applications of Information Retrieval
  - Document Ranking
    - Query: Obama family tree
    - Document:
      - **Family of Barack Obama** - Wikipedia
      - Barack **Obama Family Tree** along with family connections to other famous kin. Genealogy charts for Barack Obama may include up to 30 generations of ...
  - Question Answering
    - Query: Who is Barack Obama's sister?
    - Answer:

Maya Soetoro-Ng

Auma Obama

Information Retrieval

- Applications of Information Retrieval
  - The applications of IR can be divided into two categories:
    - Document Ranking and Question Answering

	Document Ranking	Question Answering
Query	Keywords	Natural language question
Document	Web page, news article	A fact and supporting passage
Research solution	▪ Traditional IR ▪ Neural IR	▪ Open Domain QA ▪ Generative QA ▪ Reading Comprehension ▪ Fact Verification
In products	▪ Document rankers at: Google, Bing, Baidu…	▪ Microsoft Xiaoice ▪ Watson@Jeopardy

Outline

- Introduction to IR
- Traditional IR Models
- Neural IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

Traditional IR Methods

- Language modeling approach of IR
  - Given a query q and document d:

$$p(d|q) \approx p(q|d)p(d)$$

    - $\mathrm{p}(d)$ can be assumed uniform across docs
    - $\mathrm{p}(q|d) = \prod_{w \in q} p(w|d)$ depends on how to model the relationship of query word and doc
  - The language modeling approach is quite extensible
    - TF-IDF; BM25 ...

Traditional IR Methods

- TF-IDF
  - Term Frequency (TF)
    - The weight of a term that occurs in a document is simply proportional to the term frequency
    - The number of times that term t occurs in document d:

$$tf(t, D) = \frac{n_t}{n_d}$$

    - Where n_t is the number of times the term t appears in d, and n_d is the word number of the document d

Traditional IR Methods

- TF-IDF
  - Inverse Document Frequency (IDF)
    - The specificity of a term can be quantified as an inverse function of the number of documents in which term t appears
    - IDF is a measure to evaluate if term t is common or rare across the document collection D

$$\mathrm{IDF}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

    - Where N is the total number of documents in the corpus, and $|\{d \in D : t \in d\}|$ denotes the number of documents where the term t appears

Traditional IR Methods

- TF-IDF
  - A high TF-IDF value of term t requires:
    - High term frequency (TF) in the given document
    - Low document frequency (IDF) of the term in the whole collection of documents

$$\mathrm{TF-IDF}(t, D) = \mathrm{TF}(t, D) \cdot \mathrm{IDF}(\mathrm{t}, \mathrm{D})$$

Traditional IR Methods

- BM25
  - BM25 is a bag-of-word retrieval model
    - Given a query Q, which contains n words q_1, ...$q_{n,}$ the BM25 score of a document D is:

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D) \cdot (k+1)}{f(q_i,D) + k \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

      - Where $f(q_i, D)$ is the term frequency of q_i in the document D, $|D|$ is the length of D, and $avgdl$ is the average document length in the document collection
    - BM25 aims to normalize term frequency according to document length

Traditional IR Methods

- Sequential Dependence Model (SDM):
  - Models term dependence for IR
  - Provides a good balance between retrieval effectiveness and efficiency
  - The SDM score is calculated with:
    - Unigram term frequency f_T
    - Bigram term frequency f_O (with order) and f_U (unorder)

$$p(q|d) = \lambda_T \sum_{t_q^i \in q} f_T(t_q^i|d)$$
$$+ \lambda_O \sum_{t_q^i, t_q^{i+1} \in q} f_O(t_q^i, t_q^{i+1}|d)$$
$$+ \lambda_U \sum_{t_q^i, t_q^{i+1} \in q} f_U(t_q^i, t_q^{i+1}|d)$$

    - Where $\lambda_T + \lambda_O + \lambda_U = 1$

D. Metzler and W. B. Croft, A Markov Random Field model for term dependencies. SIGIR 2015

Traditional IR Methods

- Traditional IR methods
  - Pros
    - Have ability to deal with large scale data
    - Do not need annotated labels
  - Cons
    - Have vocabulary mismatch problem
    - Perform shallow understanding for queries and documents

Traditional IR Methods

- Traditional IR methods
  - Vocabulary mismatch
    - Q: How many people live in Sydney?

      ➢ Sydney's population is 4.9 million
      [relevant, but missing 'people' and 'live']

      ➢ Hundreds of people queueing for live music in Sydney
      [irrelevant, and matching 'people' and 'live']
  - Perform shallow understanding for queries and documents
    - Query: Albuquerque

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.

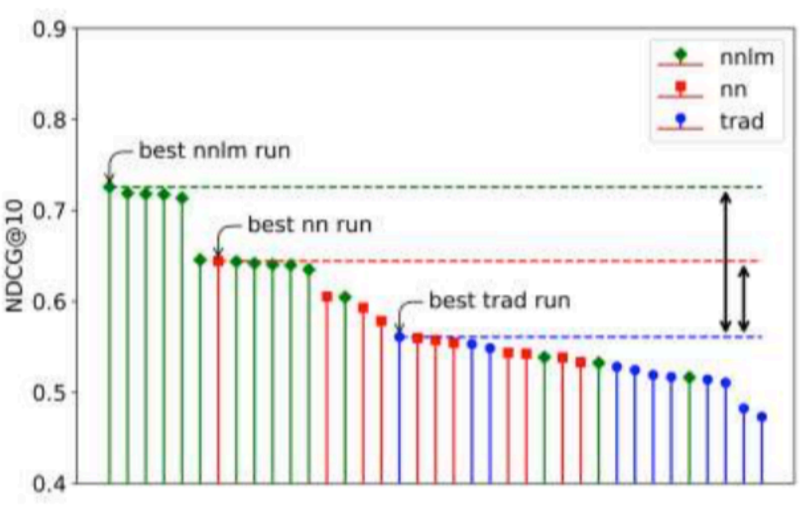
Passage not about Albuquerque

Outline

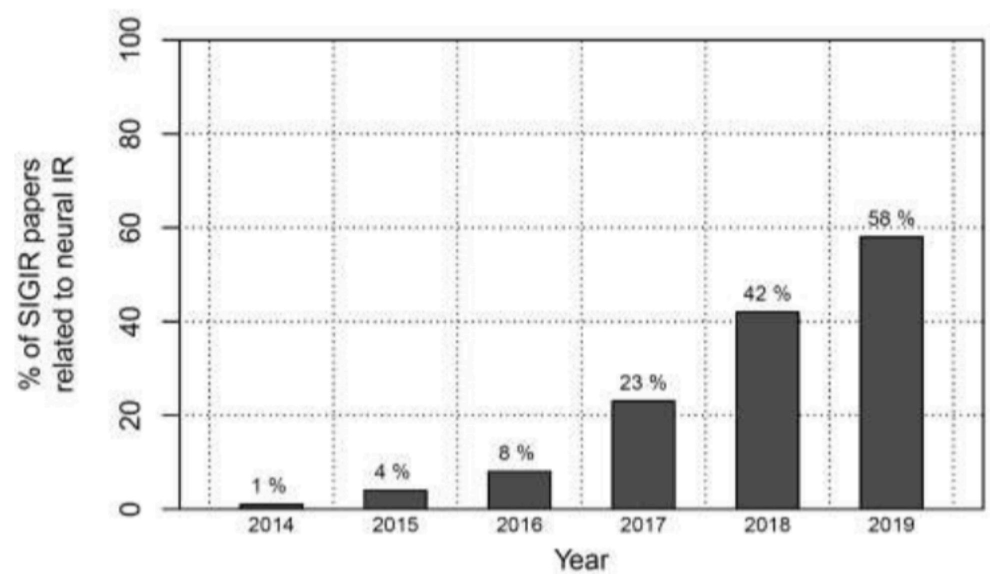
- Introduction to IR
- Traditional IR Models
- Neural IR Models
  - Neural Models for IR
  - Representation-based IR Models
  - Interaction-based IR Models
  - Summary of Neu-IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

Neural Models for IR

- Why choose neural models
  - Neural models outperform traditional IR models significantly
  - Being neural has become a tendency for IR



(a) Document retrieval task

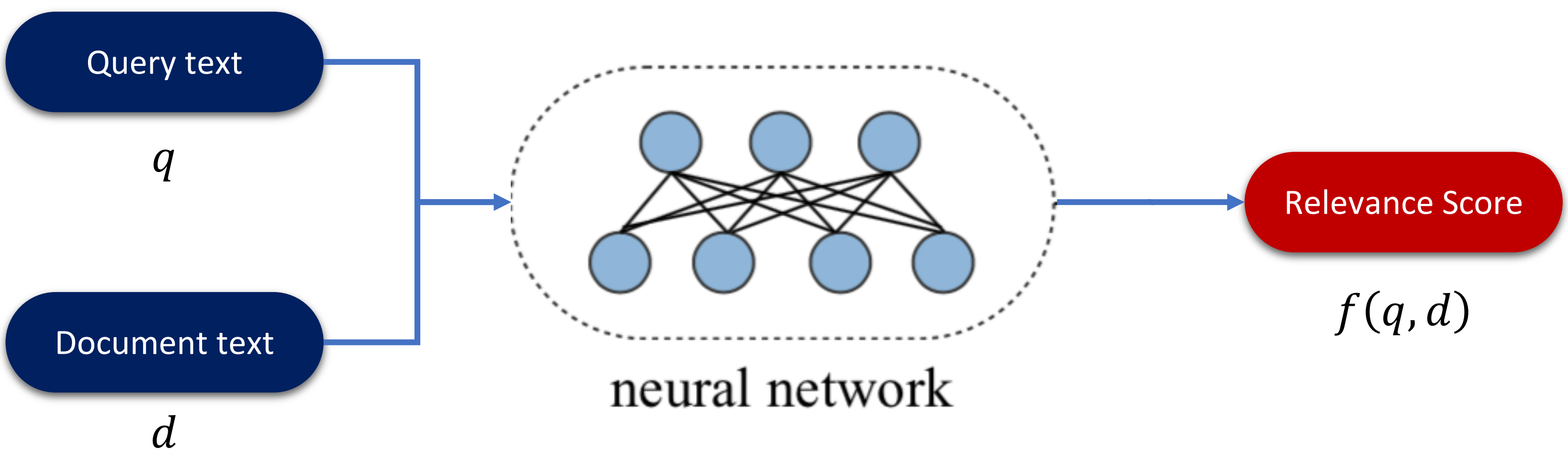


Neural Models for IR

- Why choose neural models
  - Deeper model has stronger ability to fit data

Neural Models for IR

- Given a query q and a document d
  - We can use a neural network to get relevance score $f(q, d)$
  - Then train and optimize the neural model
    - Pairwise training
    - Pointwise training



Neural Models for IR

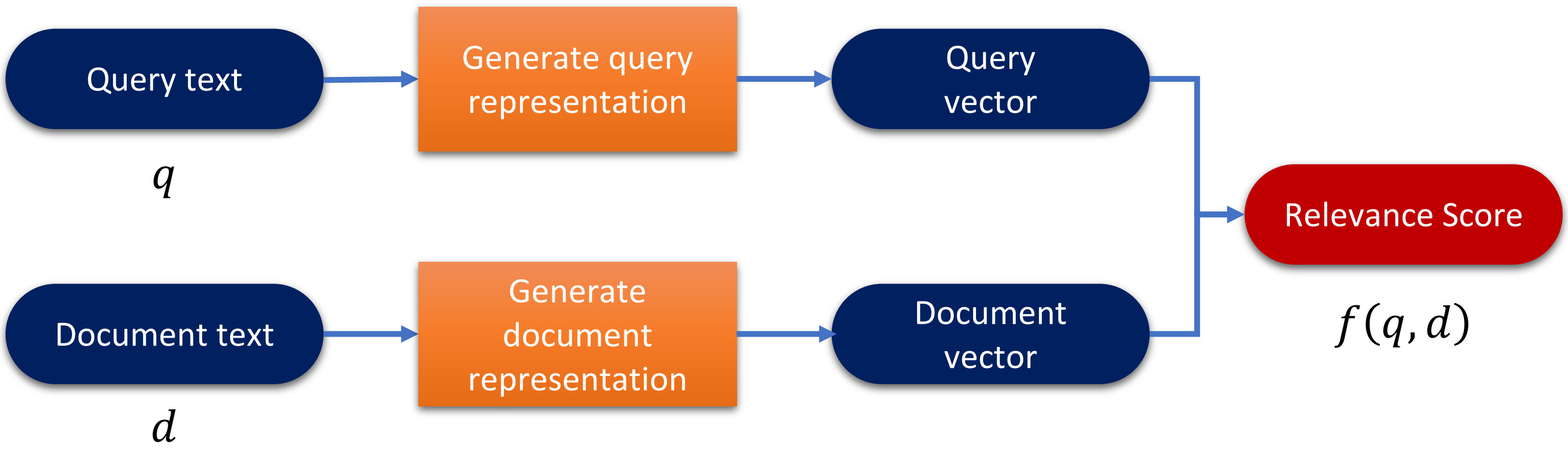
- Given a query q and a document d
  - We can use a neural network to get relevance score $f(q, d)$
  - Then train and optimize the neural model
    - Pointwise training
      - $L = \|y - f(q, d)\|^2$
      - $L = CrossEntropy(f(q, d), y)$
    - Pairwise training
      - $L = \phi(f(q, d_+) - f(q, d_-))$
        - Hinge function $\phi(z) = max(0, 1 - z)$
        - Exponential function $\phi(z) = e^{-z}$
        - Logistic function $\phi(z) = \log(1 + e^{-z})$
      - $L = -\log(\frac{e^{f(q,d_+)}}{e^{f(q,d_+)} + e^{f(q,d_-)}})$

Outline

- Introduction to IR
- Traditional IR Models
- Neural IR Models
  - Neural Models for IR
  - Representation-based IR Models
  - Interaction-based IR Models
  - Summary of Neu-IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

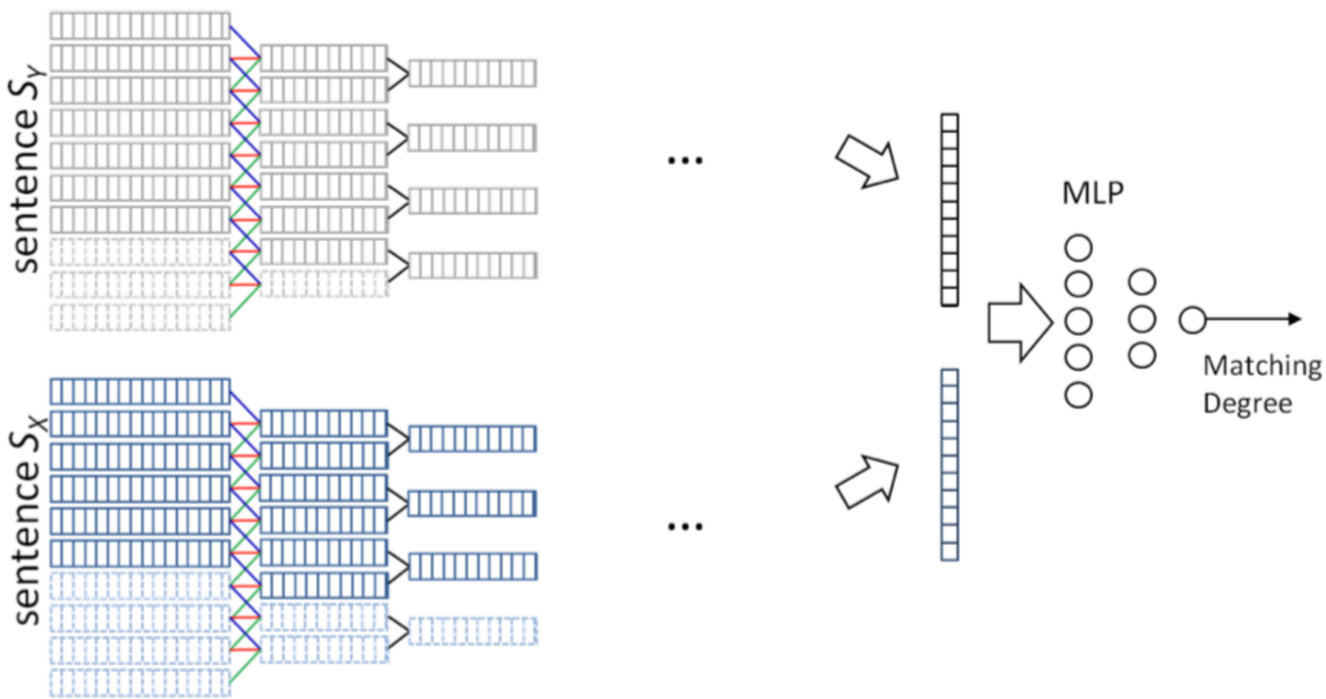
Representation-based IR Models

- Representation-based IR models
  - Use neural networks to generate query and document representations
  - Then estimate the relevance of the query and document



Representation-based IR Models

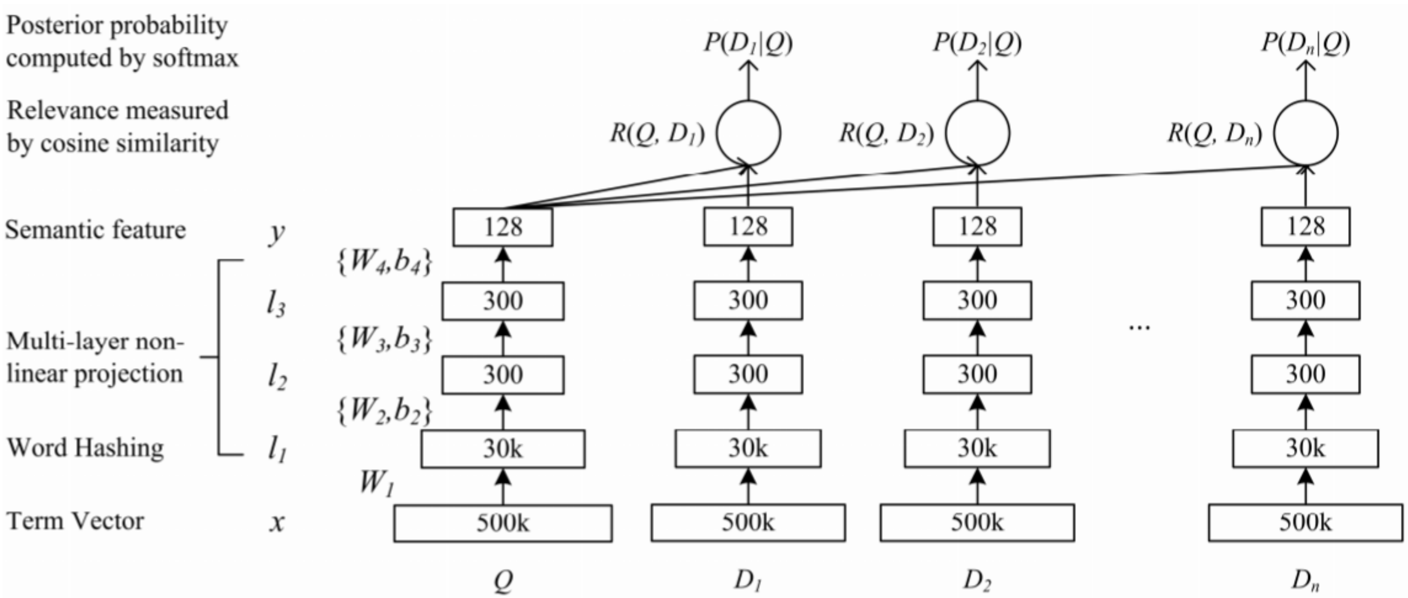
- ARC-I
  - Stacked layers of convolution and max-pooling



Hu et al. Convolutional Neural Network Architectures for Matching Natural Language Sentences. 2014

Representation-based IR Models

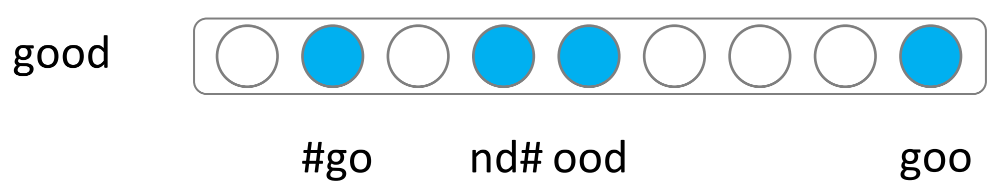
- Deep Semantic Similarity Model (DSSM)
  - Input: Character trigram counts after word hashing
  - Query and document relevance is estimated by the cosine similarity of their representations



Huang et al. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. CIKM 2013.

Representation-based IR Models

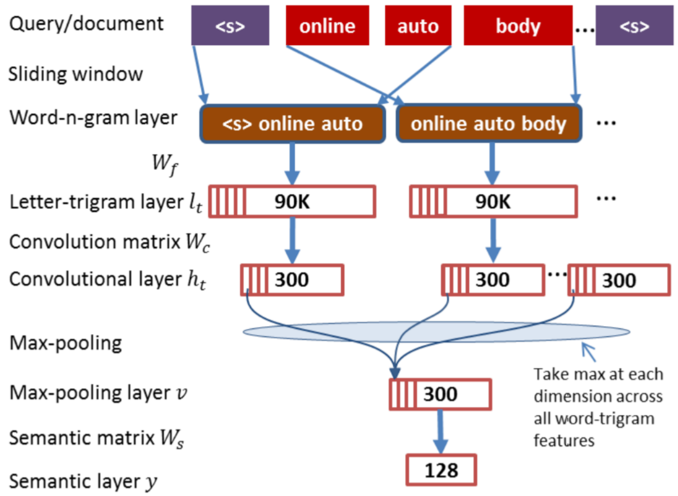
- Deep Semantic Similarity Model (DSSM)
  - Word hashing
    - The word hashing method aims to reduce the dimension of the word representation
      - Given a word
        - **good**
      - Add a mark (#) to the start and end of the word
        - **#good#**
      - Break the word into letter n-grams
        - **trigrams: #go, goo, ood, od#**
      - Represent the word using a vector of letter n-grams



Huang et al. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. CIKM 2013.

Representation-based IR Models

- Convolutional Latent Semantic Model (CLSM)
  - A convolutional layer extract contextual features for each word with its neighboring words
    - Capture context information for queries and docs
      - Word-n-grams obtained by running a sliding window over an input sequence
      - Get the representation of each composition through word-hashing

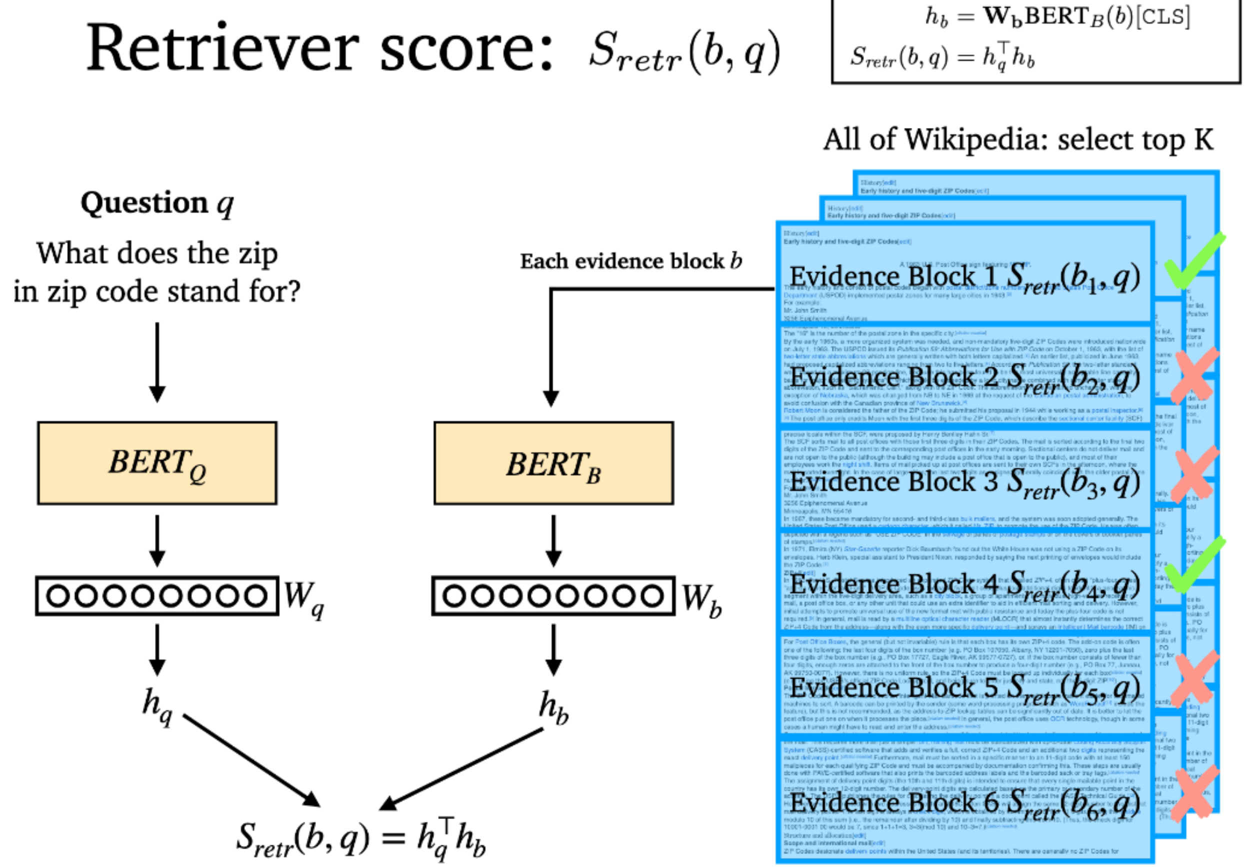


Huang et al. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. CIKM 2013.

Representation-based IR Models

- BERT

$$h_q = \mathbf{W_q}\mathrm{BERT}_Q(q)[\mathrm{CLS}]$$
$$h_b = \mathbf{W_b}\mathrm{BERT}_B(b)[\mathrm{CLS}]$$
$$S_{retr}(b,q) = h_q^\top h_b$$



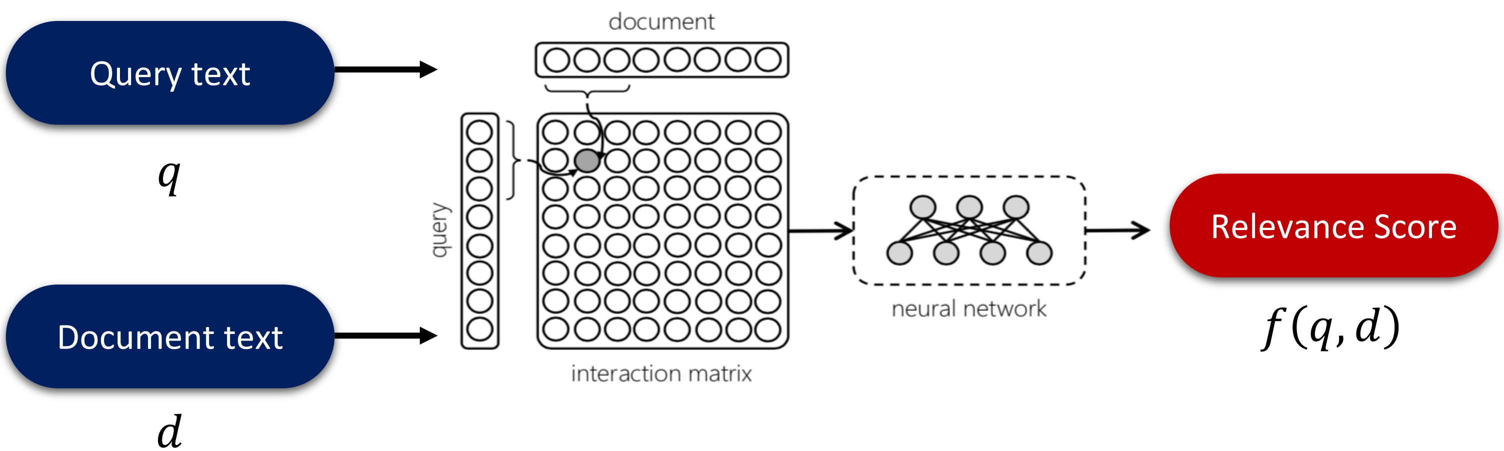
Karpukhin et al., Dense Passage Retrieval for Open-Domain Question Answering. EMNLP 2020.

Outline

- Introduction to IR
- Traditional IR Models
- Neural IR Models
  - Neural Models for IR
  - Representation-based IR Models
  - Interaction-based IR Models
  - Summary of Neu-IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

Interaction-based IR Models

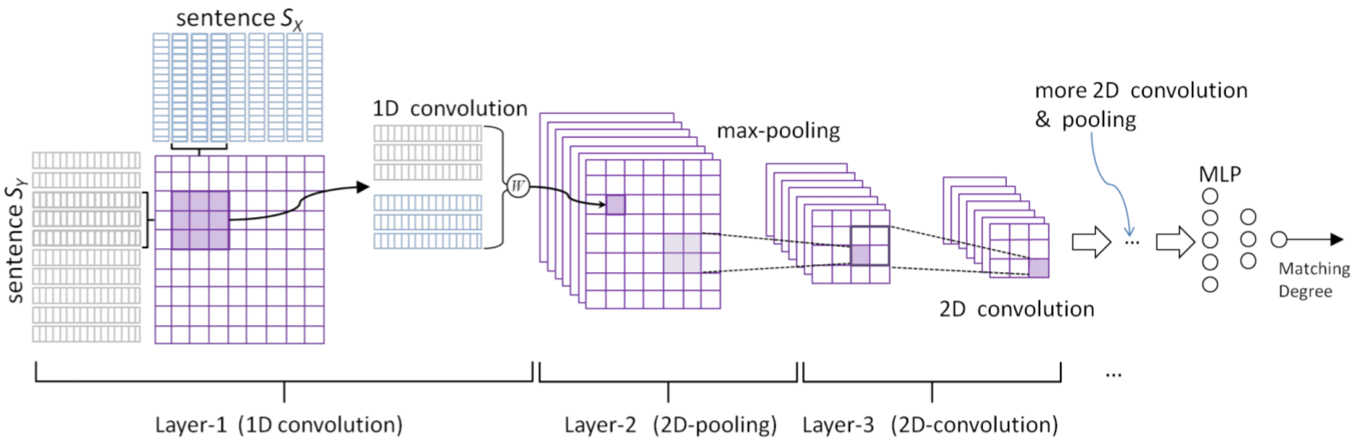
- Interaction-based IR models
  - Establish an interaction matrix M
    - M_{ij} is obtained by comparing the i^{th} word in query and the j^{th} word in doc
      - For example, $M_{ij} = \cos(\vec{v}_{t_i}, \vec{v}_{t_j})$
  - Employ neural networks to extract features and get the ranking score



Kenter et al., Neural Networks for Information Retrieval. WSDM 2018.

Interaction-based IR Models

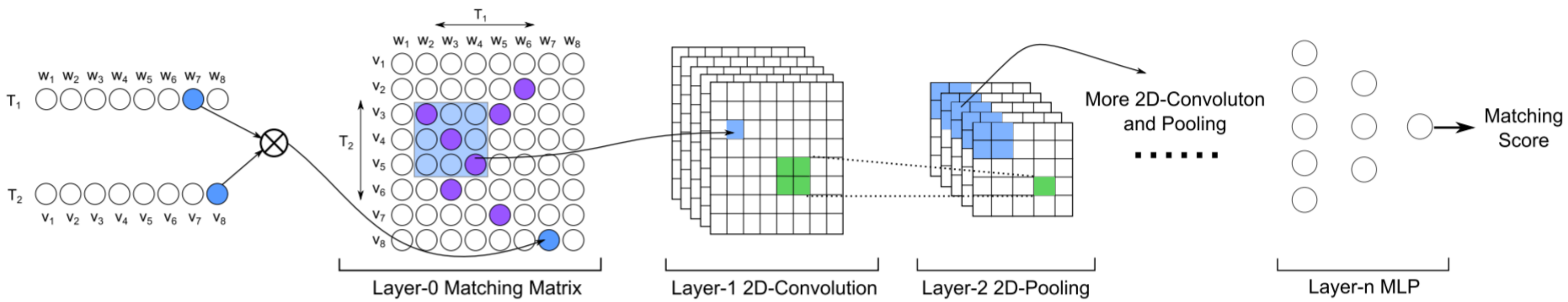
- ARC-II
  - Takes the sliding window on the sentence, and model all word-n-grams through the one-dimensional convolution
  - Obtains an interaction matrix between two sentences (Concatenation word-n-gram representations)
  - Obtains a high level representation through the two-dimensional convolution



Hu et al., Convolutional Neural Network Architectures for Matching Natural Language Sentences. 2014.

Interaction-based IR Models

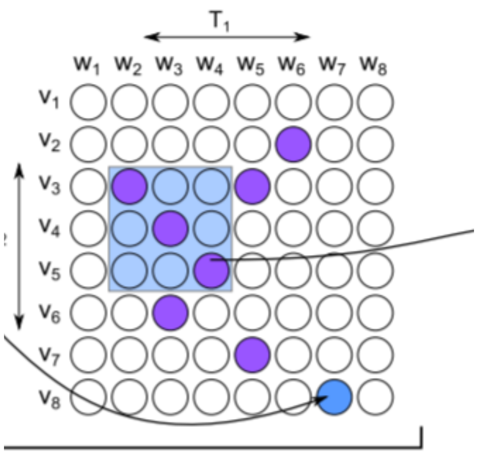
- MatchPyramid
  - MatchPyramid has three parts:
    - Interaction matrix
    - Hierarchical convolution (N convolutional layers)
    - Matching score aggregation (MLP)

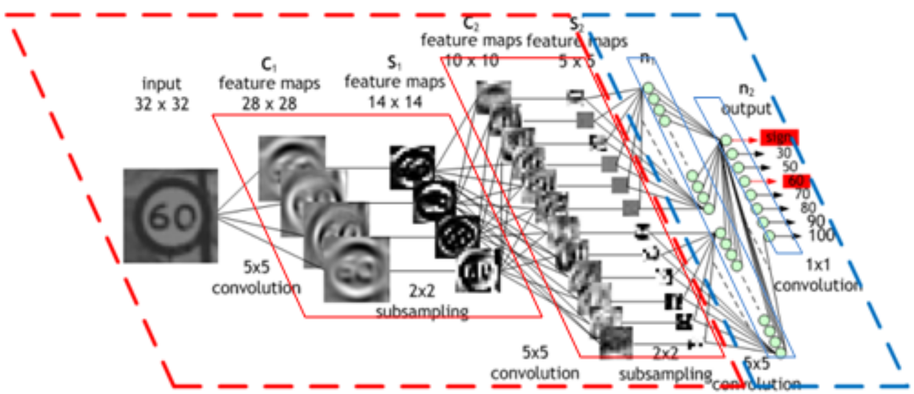


Pang et al., A Study of MatchPyramid Models on Ad-hoc Retrieval. SIGIR 2016.

Interaction-based IR Models

- MatchPyramid
  - Employs a CNN over the interaction between queries and docs to produce the matching score
    - CNN in image recognition often focus on the edge of the object





Pang et al., A Study of MatchPyramid Models on Ad-hoc Retrieval. SIGIR 2016.

Interaction-based IR Models

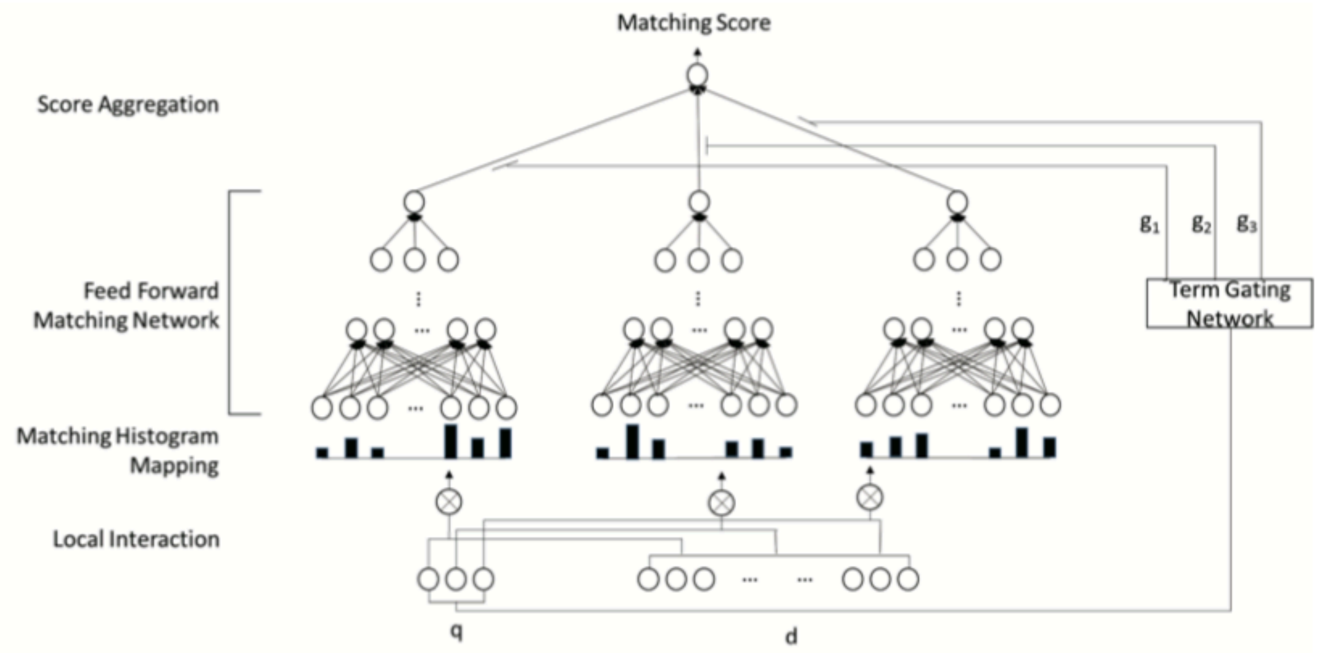
- Similarity Functions in MatchPyramid:
  - Indicator Function produces either 1 or 0 to indicate whether two words are identical
  - Cosine views the angle between two word vectors as the similarity
  - Dot Product further considers the norm of word vectors, as compared to the cosine
  - Gaussian Kernel is a well-known similarity function

Model	MAP	nDCG@20
MP-Ind	0.225	0.387
MP-Dot	0.095	0.149
MP-Cos	0.189	0.340
MP-Gau	**0.226**	**0.403**

Pang et al., A Study of MatchPyramid Models on Ad-hoc Retrieval. SIGIR 2016.

Interaction-based IR Models

- Deep Relevance Matching Model (DRMM)

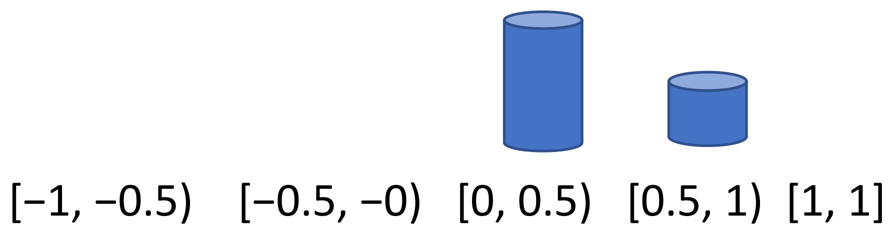


Guo et al., A Deep Relevance Matching Model for Ad-hoc Retrieval. CIKM 2016.

Interaction-based IR Models

- Deep Relevance Matching Model (DRMM)
  - Matching histogram mapping

	H_1^d	H_2^d	H_3^d	H_4^d	H_5^d	H_5^d	H_7^d	H_8^d
H_1^q	0.18	0.22	0.19	0.23	0.21	0.82	0.78	0.19

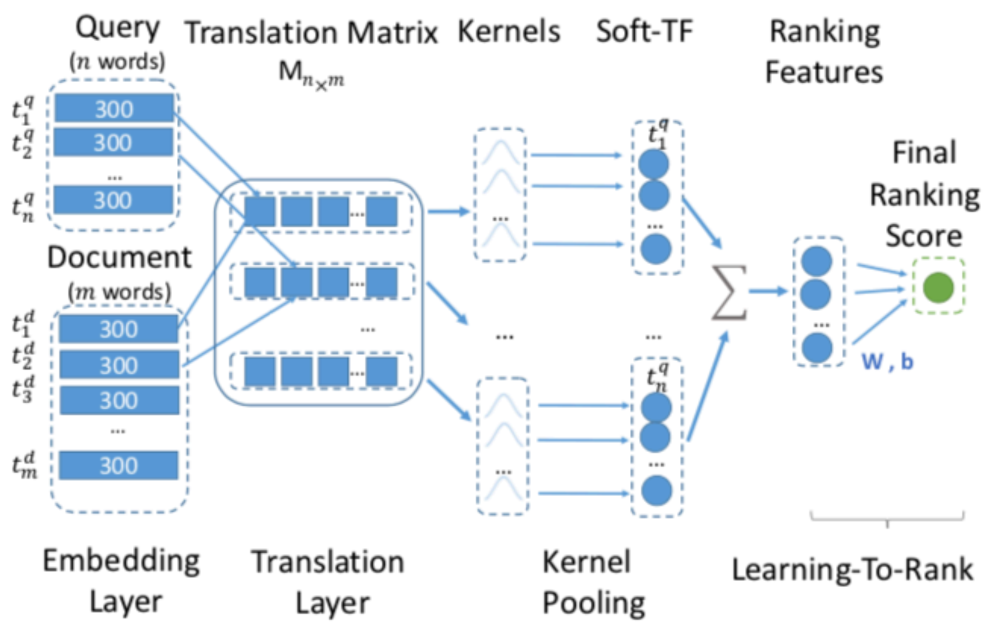


$$(0, 0, 6, 2, 0)$$

Guo et al., A Deep Relevance Matching Model for Ad-hoc Retrieval. CIKM 2016.

Interaction-based IR Models

- Kernel-based Neural Ranking Model (K-NRM)
  - Learning embedding tailored for relevance ranking
  - End-to-end training from user feedback (User click signal)
  - Soft-matching at word level



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

Interaction-based IR Models

- Kernel-based Neural Ranking Model (K-NRM)
  - Embedding layer maps each word to an L-dimension vector
  - Then K-NRM constructs an interaction matrix M
  - Kernel-Pooling converts word-word interactions to the query-document ranking feature
  - Learning-to-Rank (LeToR) combines the ranking feature to produce the final ranking score

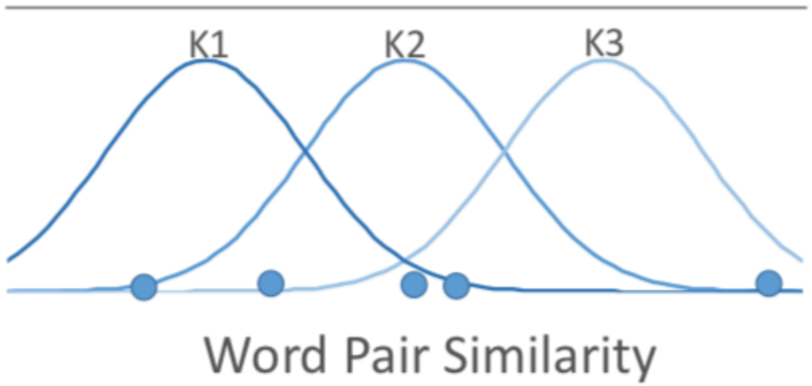
Xiong et al., End-to-End Neural Ad-hoc Ranking with Kernel Pooling. SIGIR 2017.

Interaction-based IR Models

- Kernel-based Neural Ranking Model (K-NRM)
  - Radial Basis Function (RBF) Kernel:

$$K_k(M_i) = \sum_j \exp(-\frac{(M_{ij} - \mu_k)^2}{2\sigma_k^2})$$

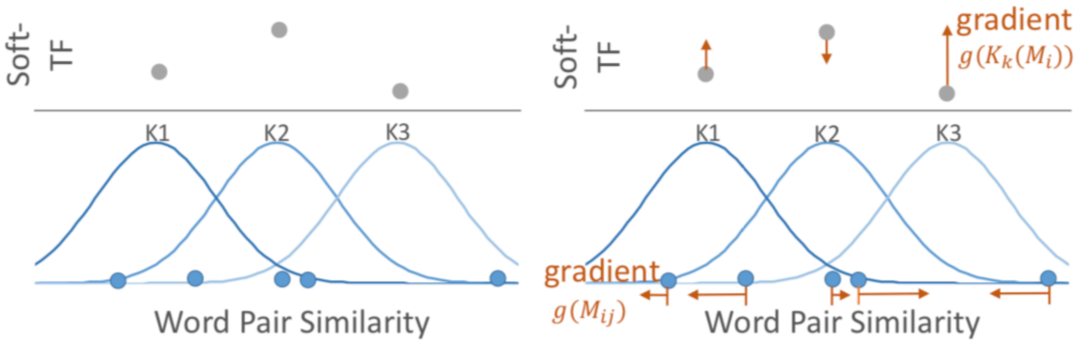
    - Where K_k is the k-th kernel, μ_k is the mean of kernel k, σ defines the kernel width, and M is the interaction matrix



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

Interaction-based IR Models

- Kernel-Pooling in K-NRM
  - Soft-TF
    - Uses kernels to softly count the frequencies of word pairs at different similarity levels
    - Counts soft-match pairs at multiple similarity levels using Kernels



(a) Ranking (b) Learning

$$\phi(M) = \sum_{i=1}^{n} log\vec{K}(M_i)$$

$$\vec{K}(M_i) = \{K_1(M_i), \ldots, K_k(M_i)\}$$

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

Interaction-based IR Models

- Kernel-based Neural Ranking Model (K-NRM)
  - Examples of word pairs:
    - During training, K-NRM adjusts word embeddings to produce soft matches that can better separate relevant and irrelevant docs

From	To	Word Pairs
$\mu = 0.9$ (0.20, –)	$\mu = 0.1$ (0.23, –)	(wife, husband), (son, daughter), (China-Unicom, China-Mobile)
$\mu = 0.5$ (0.26, –)	$\mu = 0.1$ (0.23, –)	(Maserati, car),(first, time) (website, homepage)
$\mu = 0.1$ (0.23, –)	$\mu = -0.3$ (0.30, +)	(MH370, search), (pdf, reader) (192.168.0.1, router)
$\mu = 0.1$ (0.23, –)	$\mu = 0.3$ (0.26, –)	(BMW, contact-us), (Win7, Ghost-XP)
$\mu = 0.5$ (0.26, –)	$\mu = -0.3$ (0.30, +)	(MH370, truth), (cloud, share) (HongKong, horse-racing)
$\mu = -0.3$ (0.30, +)	$\mu = 0.5$ (0.26, –)	(oppor9, OPPOR), (6080, 6080YY), (10086, www.10086.com)

Values in parenthesis are MRR of the individual kernel, indicating the importance of the kernel.

'+' means word pair appearances in the corresponding kernel are positively correlated with relevance; '–' means negatively correlated.

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

Interaction-based IR Models

- Conv-KNRM
  - Queries and docs often match at n-gram level
    - For example:
      - Query: "Convolutional Neural Networks"
      - Doc: "Deep Learning Tutorial for beginners..."
    - Traditional IR approach: exact match n-grams
    - Interaction-based Neural IR models
      - Capture soft match using word embeddings

Dai et al., Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. WSDM 2018.

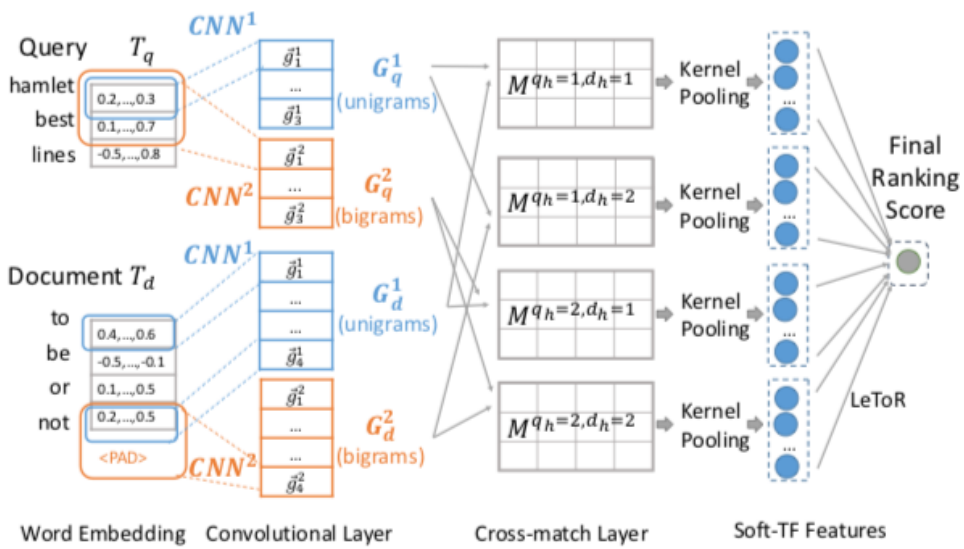
Interaction-based IR Models

- Conv-KNRM
  - Convolutional layer
    - Applys convolution layers to compose n-grams from the text
  - Cross-Match Layer
    - Builds similarity matrices between n-grams
      - Query unigrams to document unigrams
      - Query unigrams to document bigrams
      - Query bigrams to document unigrams
      - Query bigrams to document bigrams
      - …

Dai et al., Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. WSDM 2018.

Interaction-based IR Models

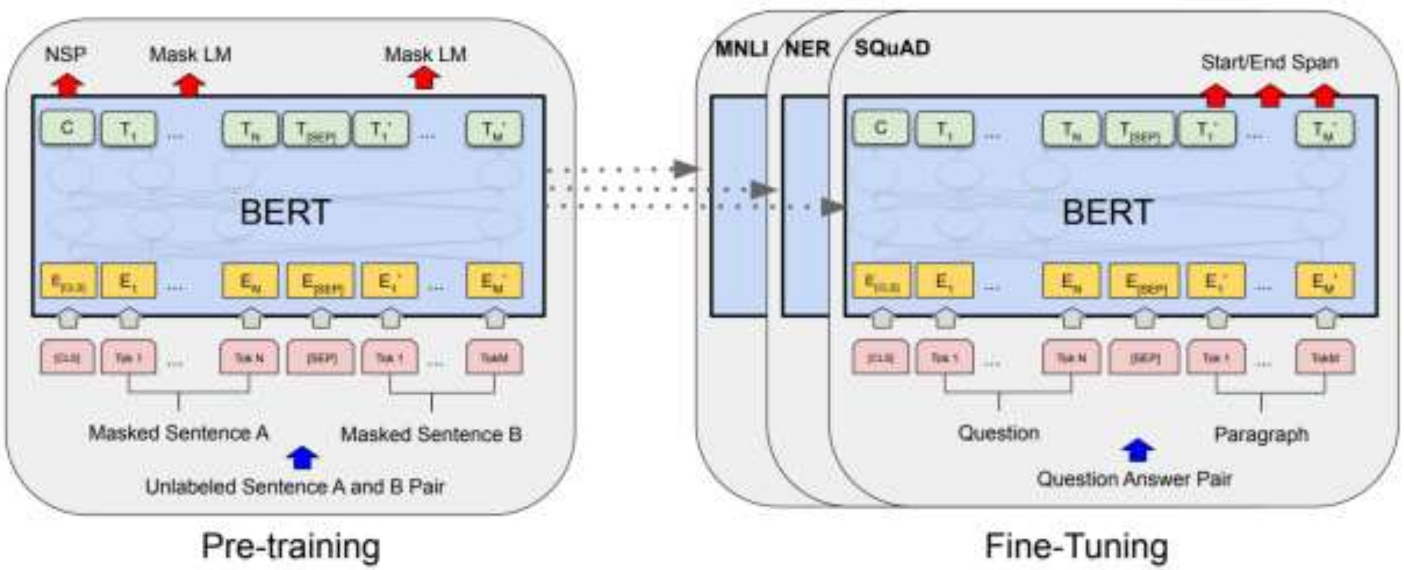
- Conv-KNRM
  - Ranking with N-gram Translations:
    - Kernel-Pooling
      - Using K Gaussian kernels to extract features of word n-gram pairs
    - Learning-to-Rank (LeToR):
      - Combining soft-TF ranking features into a ranking score



Dai et al., Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. WSDM 2018.

Interaction-based IR Models

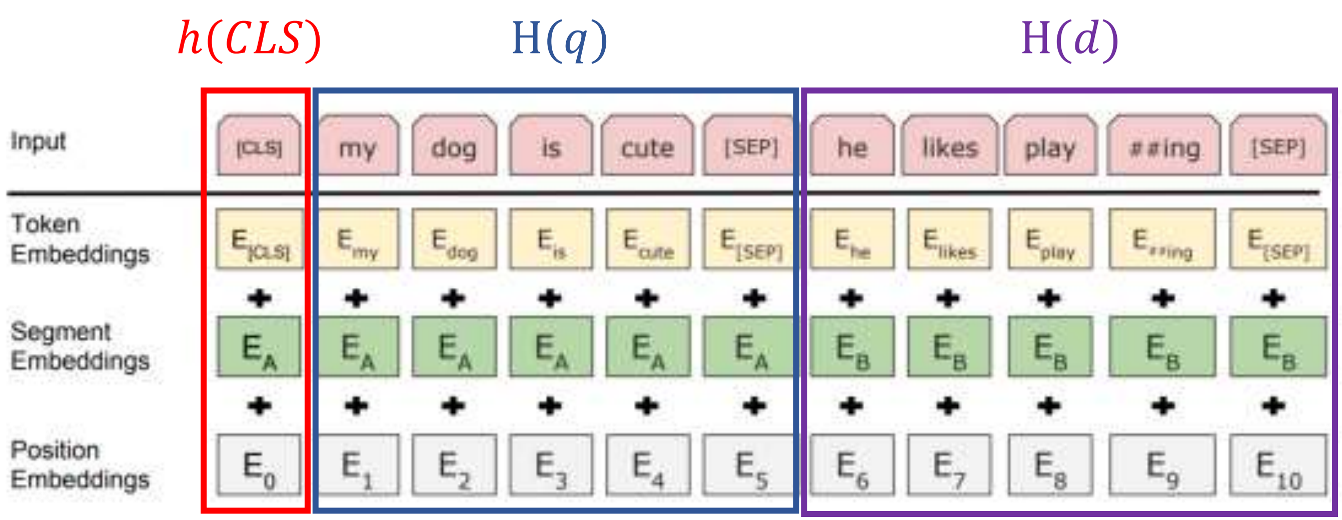
- BERT
  - Stacked transformer layers
  - BERT is pretrained on two tasks
    - Masked language modeling
    - Next sentence prediction



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

Interaction-based IR Models

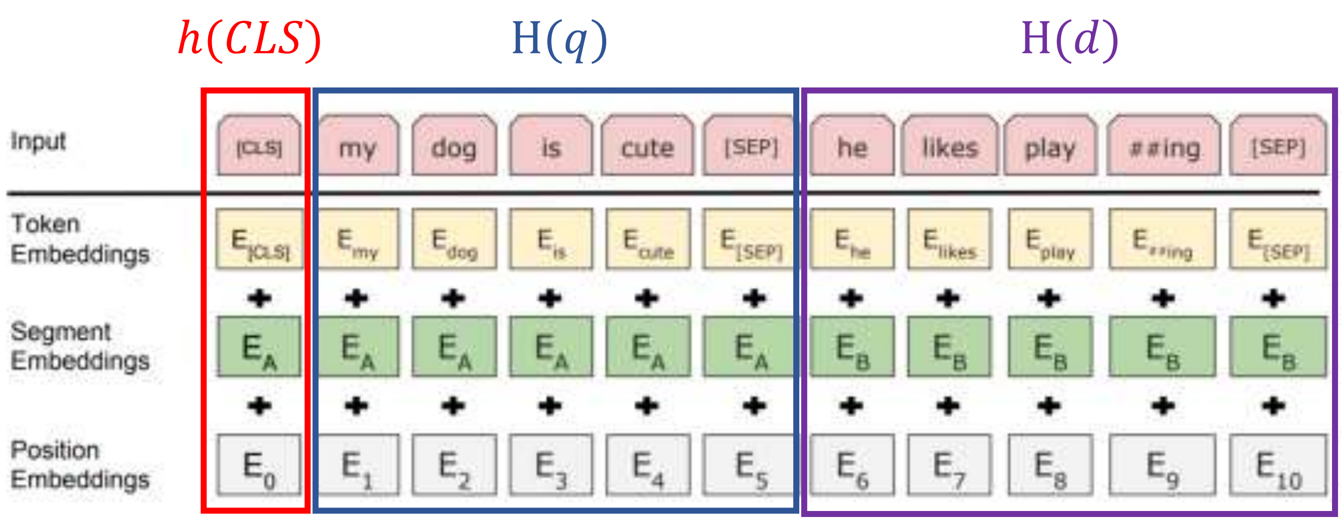
- BERT ranker
  - Given a query q and a document d.
    - Three kinds of representations are calculated
      - [CLS] representation $h(CLS)$
      - Query representation $\mathrm{H}(q)$
      - Document representation $\mathrm{H}(d)$



Qiao et al. Understanding the Behaviors of BERT in Ranking. 2019.

Interaction-based IR Models

- BERT ranker
  - Given a query q and a document d
    - The relevance score $f(q,d)$ can be calculated:
      - $f(q,d) = MLP(h(CLS))$ with [CLS] representation
      - Or $f(q,d) = MLP(\phi(H(q),H(d)))$ with query and document representations. ϕ can be interaction-based architectures



Qiao et al. Understanding the Behaviors of BERT in Ranking. 2019.

Outline

- Introduction to IR
- Traditional IR Models
- Neural IR Models
  - Neural Models for IR
  - Representation-based IR Models
  - Interaction-based IR Models
  - Summary of Neu-IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

Summary of Neu-IR Models

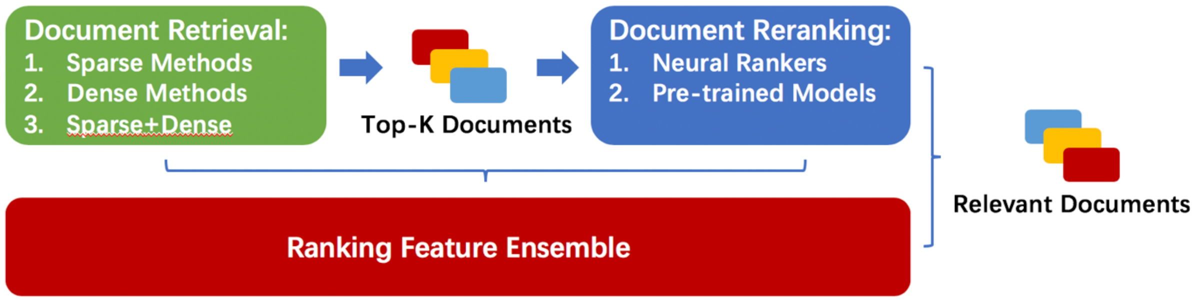
- Neural IR models
  - Can be divided into representation-based and interaction-based categories
  - Neural IR models can deal with vocabulary mismatch problem with word embeddings
  - Neural IR models help better understand natural language with sophisticated neural architectures
  - There are also some challenges in neural IR area, such as data challenge

Outline

- Introduction to IR
- Traditional IR Models
- Neural IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

Neural Models in IR Systems

- IR Pipeline
  - Document Retrieval
    - Retrieval documents from large scale document collection (Efficiency)
    - Need to recall more relevant documents
  - Document Reranking
    - Reranking documents from retrieved candidates (Effectiveness)
    - Need to provide more precision ranking results



Neural Models in IR Systems

- IR Pipeline
  - Document Retrieval
    - Sparse Models
      - Traditional IR models, such as BM25, SDM and TF-IDF
    - Dense Models
      - Representation based IR models, such as DPR and ANCE
  - Document Reranking
    - Neural Reranking Models
      - Conv-KNRM, KNRM, TK
      - BERT

Neural Models in IR Systems

- IR Pipeline
  - Document Retrieval
    - Sparse Models
      - Traditional IR models, such as BM25, SDM and TF-IDF
    - Dense Models
      - Representation based IR models, such as DPR and ANCE
  - Document Reranking
    - Neural Reranking Models
      - Conv-KNRM, KNRM, TK
      - BERT

Neural Models in IR Systems

- Improve Sparse Retrieval Models with Term Weighting
  - For the query Chinese river, word embedding gives several neighbors
    - The neighbor phrases are semantically related to the input
    - Weight query terms with averaged embeddings

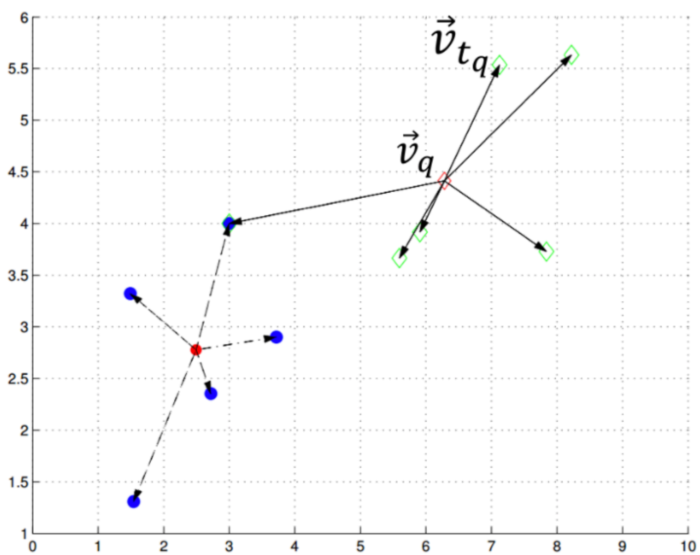
Query: Chinese river

Word	Cosine similarity
Yangtze_River	0.667376
Yangtze	0.644091
Qiantang_River	0.632979
Yangtze_tributary	0.623527
Xiangjiang_River	0.615482
Huangpu_River	0.604726
Hanjiang_River	0.598110
Yangtze_river	0.597621
Hongze_Lake	0.594108
Yangtse	0.593442

Zheng and Callan, Learning to reweight terms with distributed representations. SIGIR 2015

Neural Models in IR Systems

- Improve Sparse Retrieval Models with Term Weighting
  - Pre-trained word embedding



We calculate $|\vec{x}_{t_q}|$ to measure the semantic distance of a term to the whole query:

$$\vec{x}_{t_q} = \vec{v}_{t_q} - \frac{1}{|q|} \sum_{t'_q \in q} \vec{v}_{t'_q}$$

Where $\vec{v}_{t_q}$ is the embedding of term t_q and t'_q is the word from query other than t_q

Zheng and Callan, Learning to reweight terms with distributed representations. SIGIR 2015

Neural Models in IR Systems

- Improve Sparse Retrieval Models with Term Weighting
  - Deep Contextualized Term Weighting (DeepCT)
  - Using BERT to predict term weight
    - Document Term Weight Prediction
      - $QTR_{t,d} = |Q_{d,t}|/|Q_d|$
      - $|Q_d|$ denotes the number of queries that related with d
      - $|Q_{d,t}|$ denotes the number of queries that related with d and contain term t
    - Query Term Weight Prediction
      - $TR_{t,q} = |Q_{q,t}|/|Q_q|$
      - $|Q_q|$ denotes the number of documents that related with q
      - $|Q_{q,t}|$ denotes the number of documents that related with q and contain term t

Dai and Callan Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval. SIGIR 2020.

Neural Models in IR Systems

- Improve Sparse Retrieval Models with Term Weighting
  - Deep Contextualized Term Weighting (DeepCT)

	0 10% 20% 30% 40% >50%
Query	**do atoms make up dna**
On-Topic	**DNA** only has 5 different **atoms** - carbon, hydrogen, oxygen, nitrogen and phosphorous. According to one estimation, there are about 204 billion **atoms** in each **DNA**.
Off-Topic	Genomics in Theory and Practice. What is Genomics. Genomics is a study of the genomes of organisms. It main task is to determine the entire sequence of **DNA** or the composition of the **atoms** that make up the **DNA** and the chemical bonds between the **DNA** **atoms**.

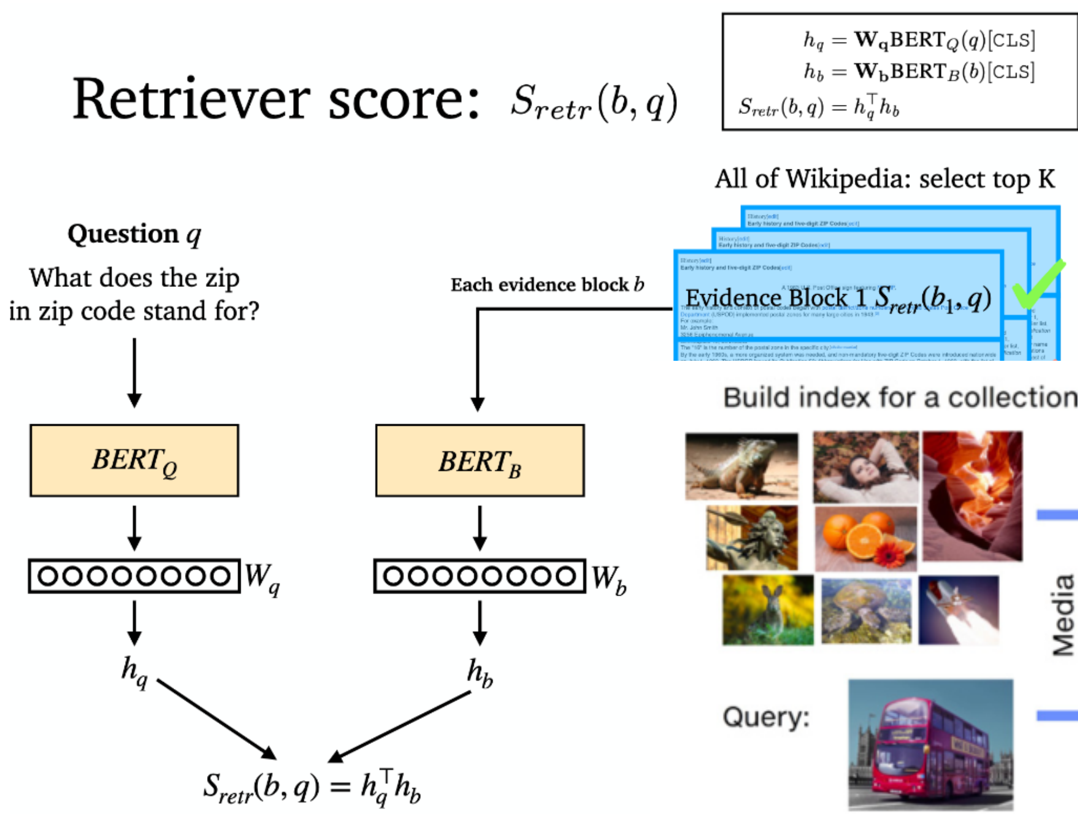
Dai and Callan Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval. SIGIR 2020.

Neural Models in IR Systems

- IR Pipeline
  - Document Retrieval
    - Sparse Models
      - Traditional IR models, such as BM25, SDM and TF-IDF
    - Dense Models
      - Representation based IR models, such as DPR and ANCE
  - Document Reranking
    - Neural Reranking Models (Usually Representation based IR models)
      - Conv-KNRM, KNRM, TK
      - BERT

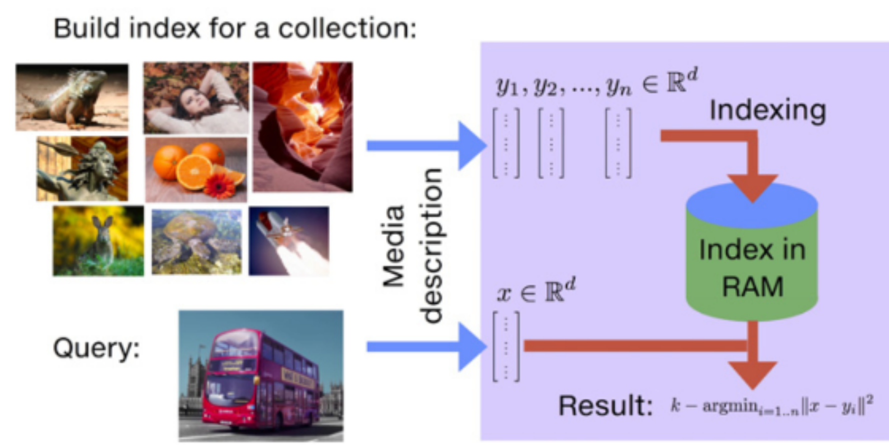
Neural Models in IR Systems

- Dense Retrieval Models



FAISS

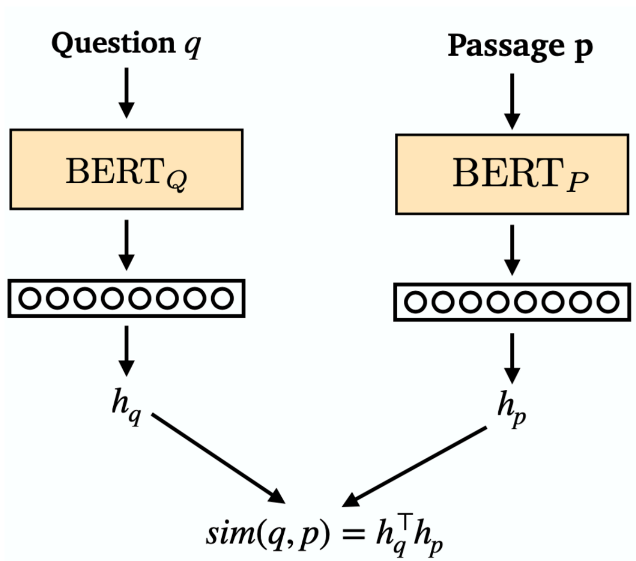
Local Sensitive Hash Indexing



(FAISS) Johnson et al - 2017 - Billion-scale similarity search with GPUs
https://github.com/danqi/acl2020-openqa-tutorial/blob/master/slides/part5-dense-retriever-e2e-training.pdf

Neural Models in IR Systems

- Dense Passage Retrieval (DPR)
  - How to Train DPR?
    - Contrastive Training



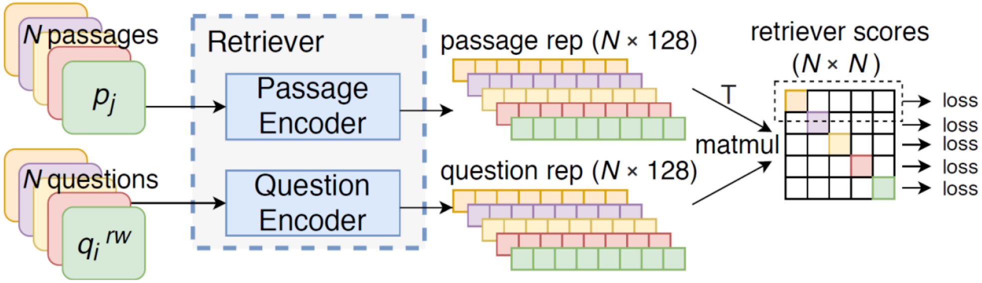
$$\mathcal{D} = \{\langle q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^- \rangle\}_{i=1}^m$$

$$\boxed{\begin{aligned} & L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-) \\ = & -\log \frac{e^{\mathrm{sim}(q_i, p_i^+)}}{e^{\mathrm{sim}(q_i, p_i^+)} + \sum_{j=1}^n e^{\mathrm{sim}(q_i, p_{i,j}^-)}} \end{aligned}}$$

Karpukhin et al., Dense Passage Retrieval for Open-Domain Question Answering. EMNLP 2020.

Neural Models in IR Systems

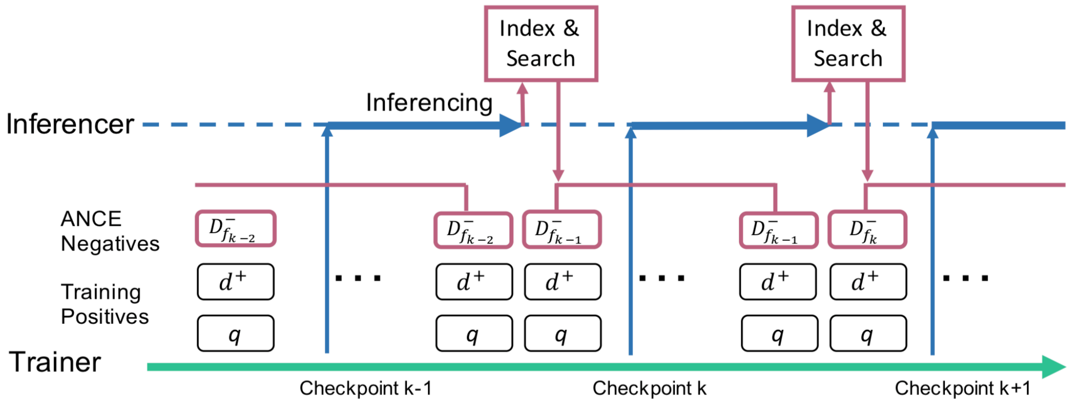
- Dense Passage Retrieval (DPR)
  - Positives
    - Provided in the reading comprehension datasets
    - Passages of high BM25 scores that contain the answer string
  - Negatives
    - Random negatives: Random passages from the corpus
    - BM25 negatives: Passages of high BM25 scores that DO NOT contain the answer string
    - In-batch negatives: Positive passages of OTHER questions



Karpukhin et al., Dense Passage Retrieval for Open-Domain Question Answering. EMNLP 2020.

Neural Models in IR Systems

- ANCE
  - ANCE provides efficient encoding methods
    - Asynchronously updated ANN index
  - Warm up with BM25 negatives
    - Training is not stable
  - Train with ANCE retrieved documents
    - To avoid Diminishing Gradients



Xiong et al., Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. ICLR 2021.

Neural Models in IR Systems

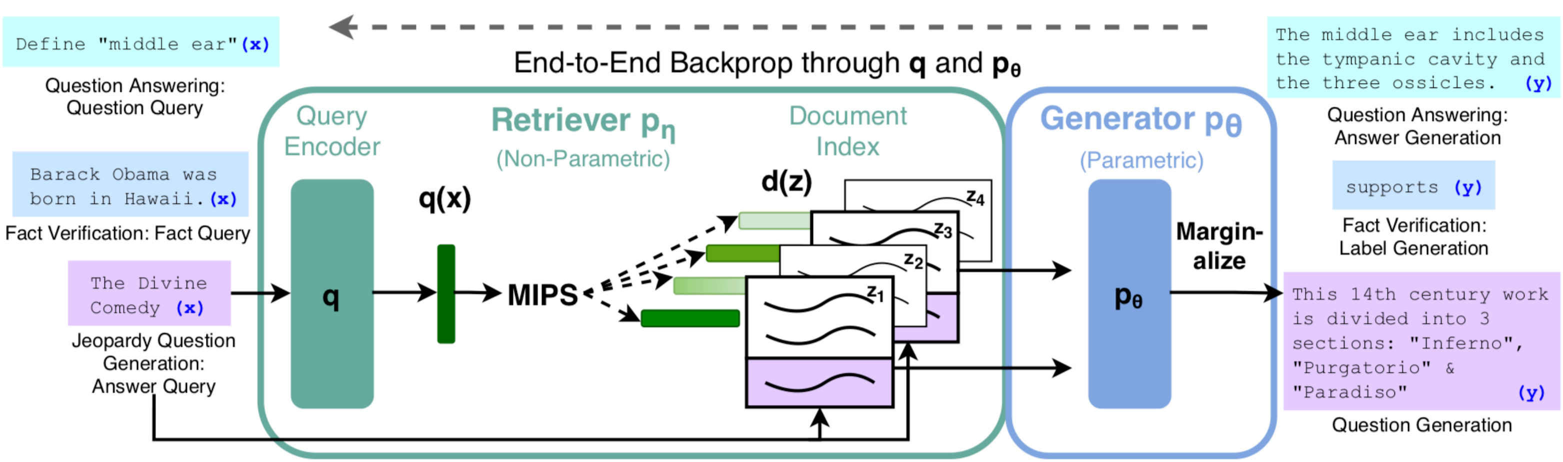
- ANCE

	MARCO Dev Passage Retrieval		TREC DL Passage NDCG@10		TREC DL Document NDCG@10	
	MRR@10	Recall@1k	Rerank	Retrieval	Rerank	Retrieval
Sparse & Cascade IR						
BM25	0.240	0.814	–	0.506	–	0.519
Best DeepCT	0.243	n.a.	–	n.a.	–	0.554
Best TREC Trad Retrieval	0.240	n.a.	–	0.554	–	0.549
BERT Reranker	–	–	**0.742**	–	0.646	–
Dense Retrieval						
Rand Neg	0.261	0.949	0.605	0.552	0.615	0.543
NCE Neg	0.256	0.943	0.602	0.539	0.618	0.542
BM25 Neg	0.299	0.928	0.664	0.591	0.626	0.529
DPR (BM25 + Rand Neg)	0.311	0.952	0.653	0.600	0.629	0.557
BM25 $\rightarrow$ Rand	0.280	0.948	0.609	0.576	0.637	0.566
BM25 $\rightarrow$ NCE Neg	0.279	0.942	0.608	0.571	0.638	0.564
BM25 $\rightarrow$ BM25 + Rand	0.306	0.939	0.648	0.591	0.626	0.540
ANCE (FirstP)	**0.330**	**0.959**	0.677	**0.648**	0.641	0.615
ANCE (MaxP)	–	–	–	–	**0.671**	**0.628**

Karpukhin et al., Dense Passage Retrieval for Open-Domain Question Answering. EMNLP 2020.

Neural Models in IR Systems

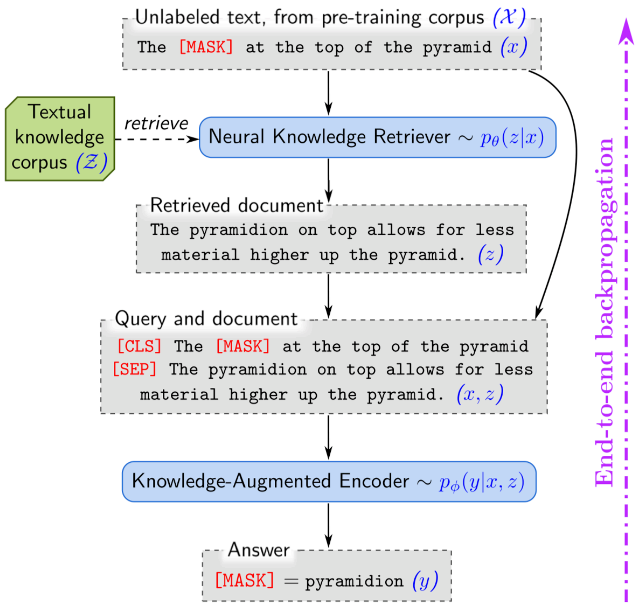
- Dense Retrieval Application
  - Retrieval-Augmented Generation (RAG)
    - For “knowledge-intensive” tasks
    - Initialized from DPR, fix document representations
    - Seq2seq generator: BART
    - Joint training: supervised with (x, y) pairs



Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, NamanGoyal, Heinrich Küttler, Mike Lewis, We tau Yih, Tim Rocktäschel, Sebastian Riedel, DouweKiela: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS2020

Neural Models in IR Systems

- Dense Retrieval Application
  - REALM
    - Retrieve and predict
    - Knowledge Retriever
    - Knowledge-Augmented Encoder



Guu et al. REALM: Retrieval-Augmented Language Model Pre-Training. 2020.

Outline

- Introduction to IR
- Traditional IR Models
- Neural IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

Challenges in Neural IR

Previous work in 2017-2019

Soft-TF with Kernel-Pooling
- KNRM [SIGIR 2017]

N-gram Soft Match with CNN
- Conv-KNRM [WSDM 2018]

Knowledge Memories
- EDRM [ACL 2018]

The Key:
- E2E relevance learned embeddings

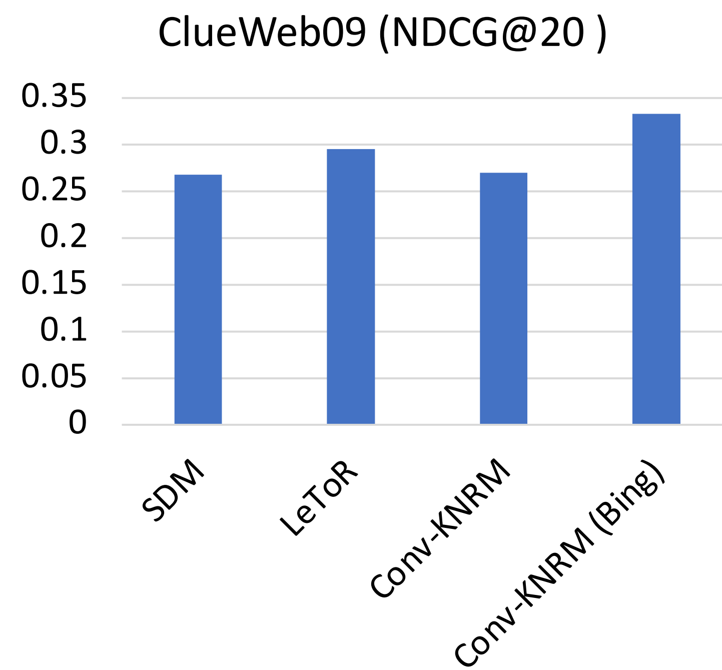
From

Cat Dog

To

Cat Videos

Effective with Search Logs
- Effective adaptation to ClueWeb [WSDM 2018]



Challenges in Neural IR

- BERT Reranker
  - Compared to Conv-KNRM, BERT mainly improves ranking performance on the question answering task
  - BERT performs better on natural language understanding than keyword matching

	MS MARCO Passage Ranking				ClueWeb09-B Ad hoc Ranking			
Method	MRR@10 (Dev)		MRR@10 (Eval)		NDCG@20		ERR@20	
Base	0.1762	−9.45%	0.1649	+13.44%	0.2496§	−6.89%	0.1387	−14.25%
LeToR	0.1946	–	0.1905	–	0.2681	–	0.1617	–
K-NRM	0.2100†‡	+7.92%	0.1982	+4.04%	0.1590	−40.68%	0.1160	−28.26%
Conv-KNRM	0.2474†‡§	+27.15%	0.2472	+29.76%	0.2118§	−20.98%	0.1443§	−10.78%
Conv-KNRM (Bing)	n.a.	n.a.	n.a.	n.a.	0.2872†‡§¶	+7.12%	0.1814†‡§¶	+12.18%
BERT (Rep)	0.0432	−77.79%	0.0153	−91.97%	0.1479	−44.82%	0.1066	−34.05%
BERT (Last-Int)	0.3367†‡§¶	+73.03%	0.3590	+88.45%	0.2407§¶	−10.22%	0.1649†§¶	+2.00%
BERT (Mult-Int)	0.3060†‡§¶	+57.26%	0.3287	+72.55%	0.2407§¶	−10.23%	0.1676†§¶	+3.64%
BERT (Term-Trans)	0.3310†§¶	+70.10%	0.3561	+86.93%	0.2339§¶	−12.76%	0.1663†§¶	+2.81%

Qiao et al. Understanding the Behaviors of BERT in Ranking. 2019.

Challenges in Neural IR

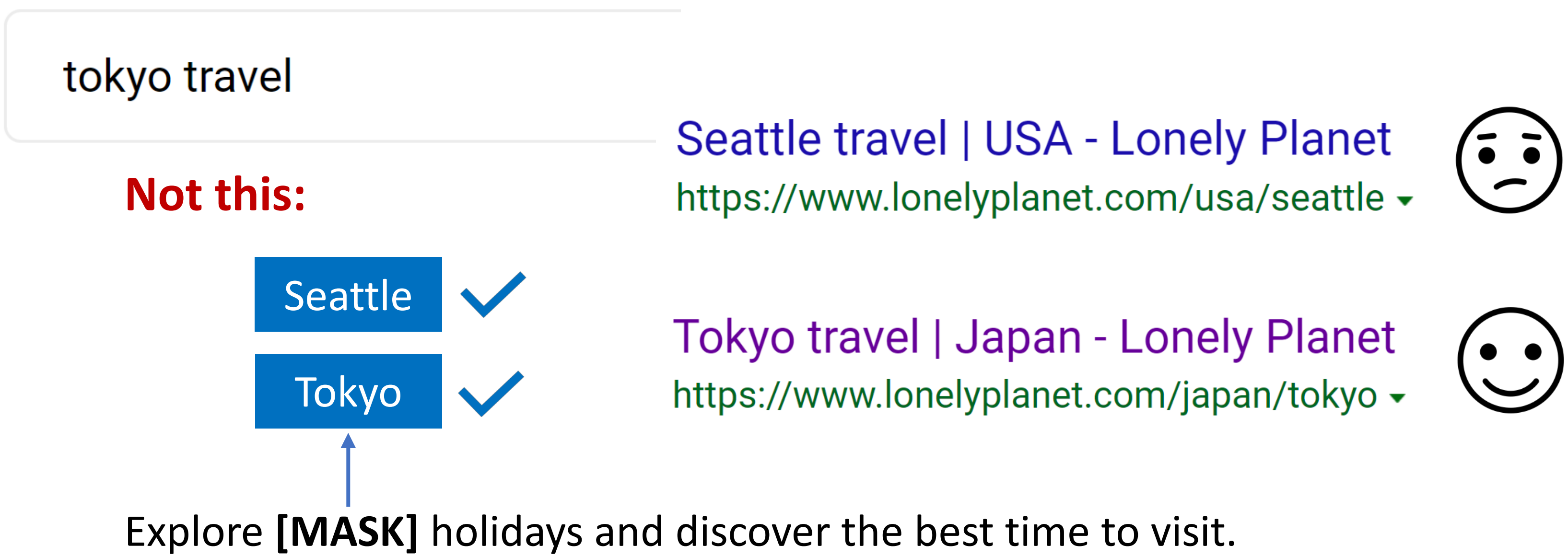
- Using Pre-trained Models
  - BERT learns an Anisotropic Embedding Space
    - Word Frequency Biases the Embedding Space
    - Low-Frequency Words Disperse Sparsely

Dataset	STS-B	SICK-R	STS-12	STS-13	STS-14	STS-15	STS-16
Published in (Reimers and Gurevych, 2019)							
Avg. GloVe embeddings	58.02	53.76	55.14	70.66	59.73	68.25	63.66
Avg. BERT embeddings	46.35	58.40	38.78	57.98	57.98	63.15	61.06
BERT CLS-vector	16.50	42.63	20.16	30.01	20.09	36.88	38.03
Our Implementation							
$\text{BERT}_{\text{base}}$	47.29	58.21	49.07	55.92	54.75	62.75	65.19

Li et al. On the Sentence Embeddings from Pre-trained Language Models. EMNLP 2020.

Challenges in Neural IR

- Using Pre-trained Models



Challenges in Neural IR

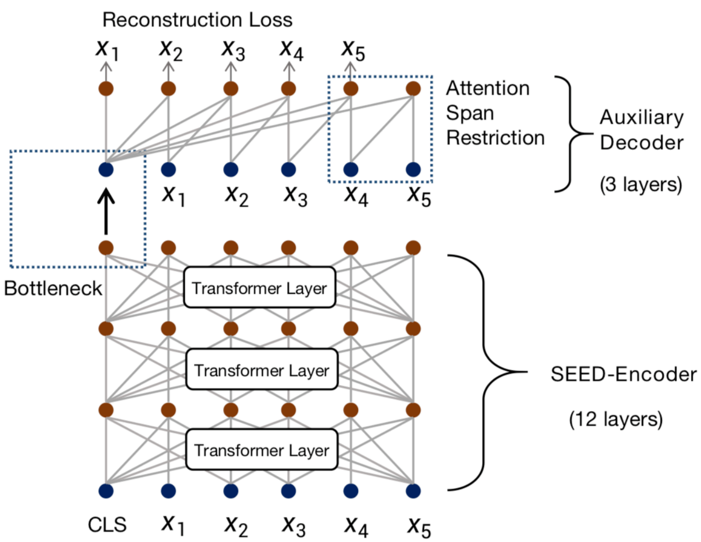
- How to better train neural IR models in IR?
  - Better Pretraining methods
  - Using large scale relevance labels

Challenges in Neural IR

- How to better train neural IR models in IR?
  - Better Pretraining methods
  - Using large scale relevance labels

Challenges in Neural IR

- Better Pretraining methods
  - Train BERT encoder with autoencoding
  - The decoder modules uses a shallow neural network



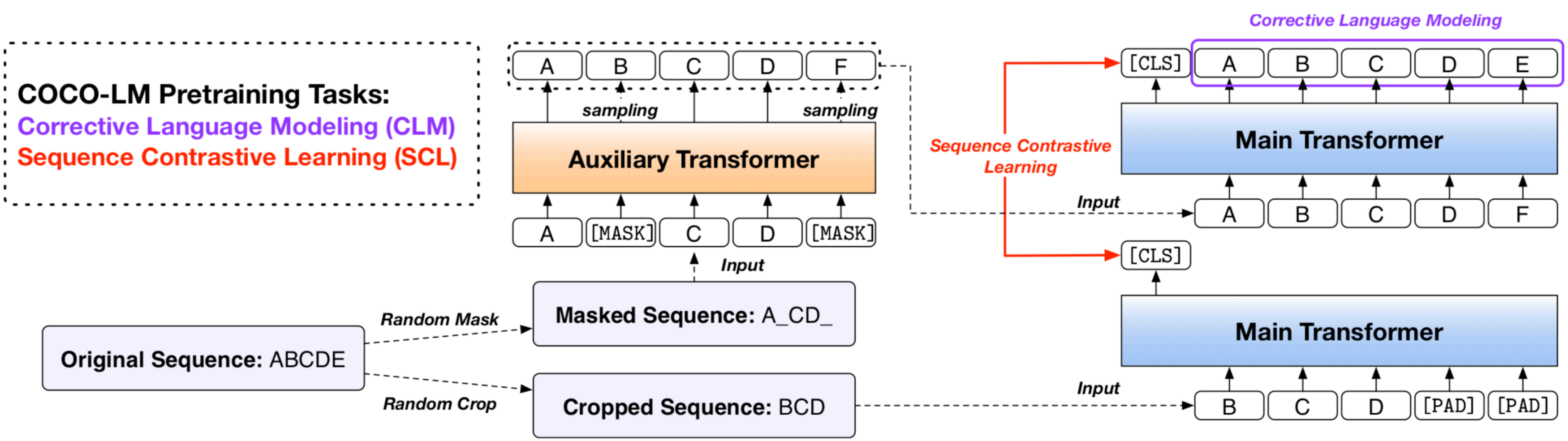
Model	Rerank MRR@10	Retrieval MRR@10	Retrieval Recall@1k
BM25 (Craswell et al., 2020)	-	0.240	0.814
Best DeepCT (Dai & Callan, 2019)	-	0.243	n.a.
Best TREC Trad IR (Craswell et al., 2020)	-	0.240	n.a.
DPR (RoBERTa) (Karpukhin et al., 2020)	-	0.311	0.952
With Siamese (BM25 Neg)			
BERT (Devlin et al., 2018)	0.317	0.310	0.929
ELECTRA (Clark et al., 2020)	0.300	0.258	0.876
ERNIE2.0 (Sun et al., 2020)	0.324	0.320	0.934
RoBERTa (Liu et al., 2019)	-	0.299	0.928
RoBERTa (Ours)	0.326	0.320	0.933
SEED-Encoder	**0.329**†	**0.329**†	**0.953**†
With ANCE (FirstP)			
RoBERTa (Liu et al., 2019)	-	0.330	0.959
RoBERTa (Ours)	0.327	0.332	0.952
SEED-Encoder	**0.334**†	**0.339**†	**0.961**†

Table 2. First stage retrieval results on MS MARCO Passage ranking Dev set. Rerank MRR is for reference only. Statistically significant improvements over RoBERTa (Ours) are marked by †.

Lu et al. Less is More: Pre-training a Strong Siamese Encoder Using a Weak Decoder. 2021.

Challenges in Neural IR

- Better Pretraining methods
  - COCO-LM



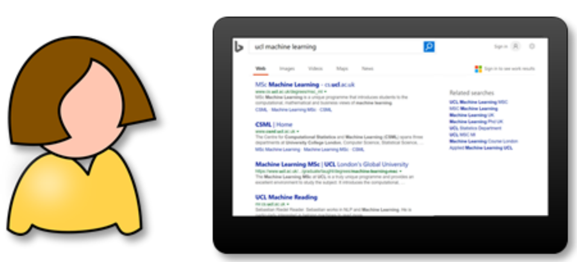
Meng et al. Pretrain Language Models by Correcting and Contrasting Text Sequences. 2021.

Challenges in Neural IR

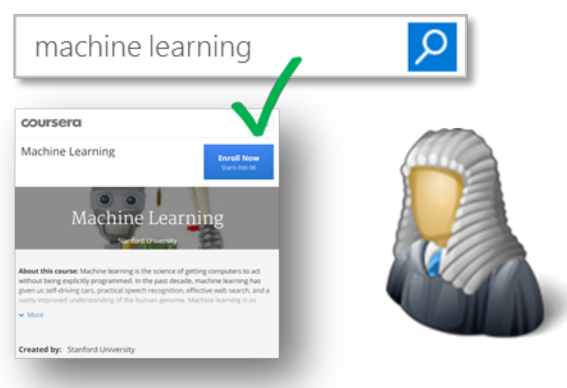
- How to better train neural IR models in IR?
  - Better Pretraining methods
  - Using large scale relevance labels

Challenges in Neural IR

- Neural IR models are fully supervised
  - Traditional IR uses human labels as ground truth for evaluation
  - So ideally we want to train our ranking models on human labels
  - User interaction data from industry is usually not available for most people and may contain different biases compared to human annotated labels



user interaction / click data



human annotated labels

Challenges in Neural IR

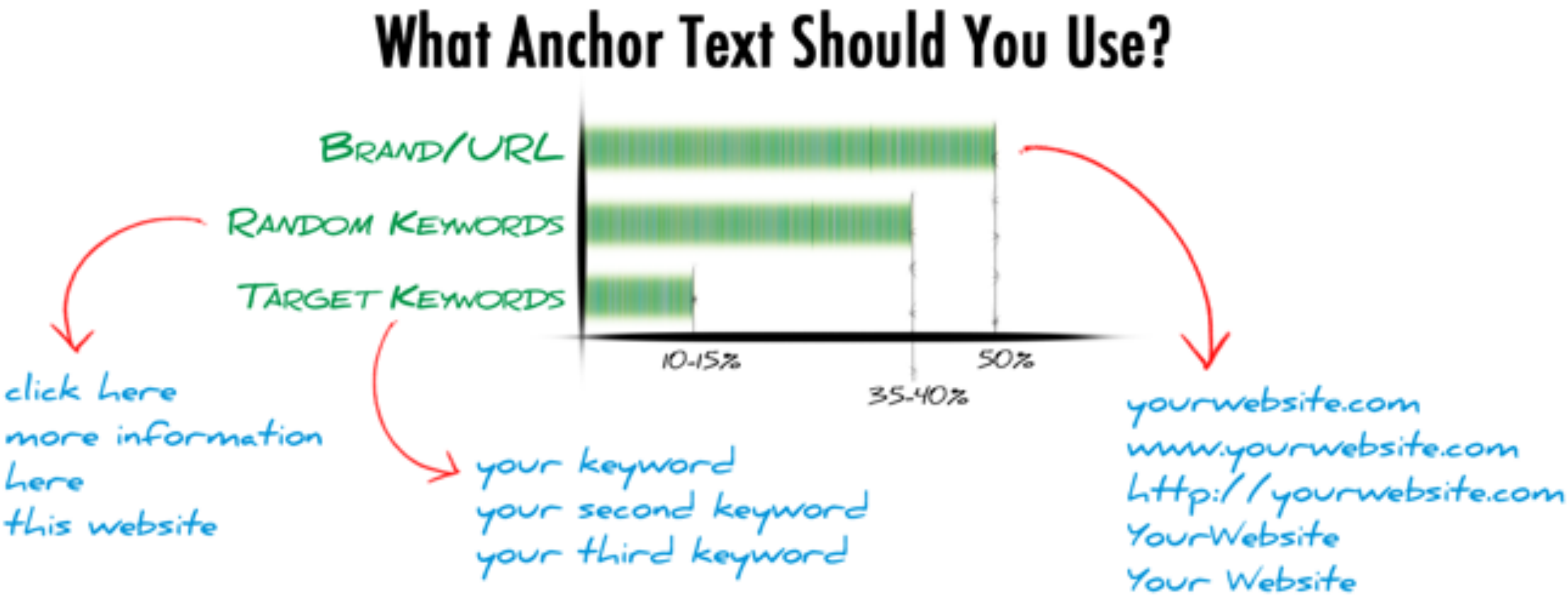
- Anchor texts are similar to query texts
- Anchor-document relations are approximate to the relevance between query and document

New York City Transit Police

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- Anchor-document data could be very noisy, and the noise data may hurt performance of neural IR methods



Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

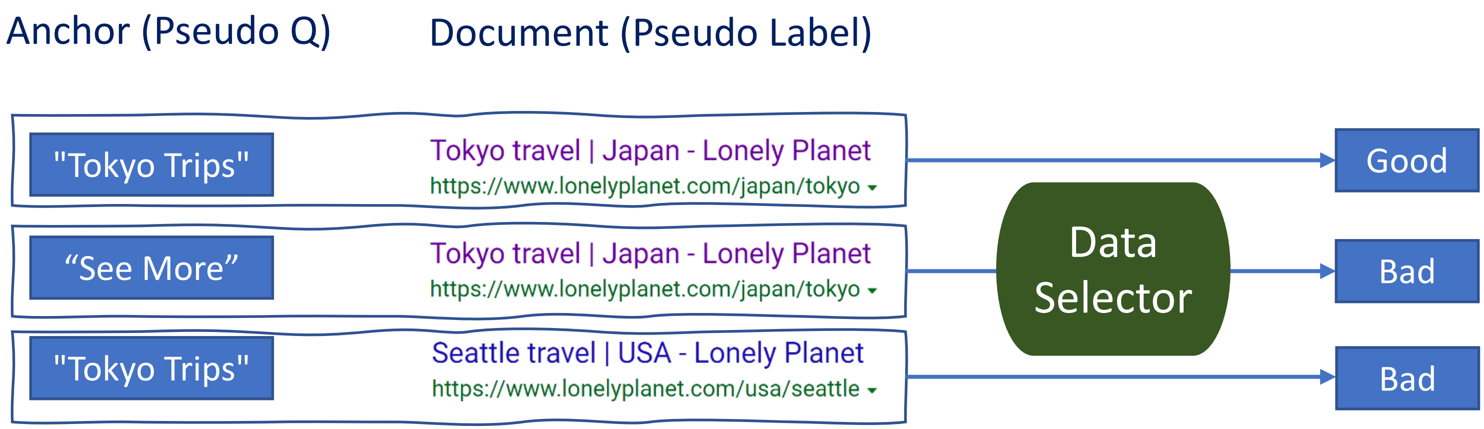
Challenges in Neural IR

- Reinforcement data selection (ReinfoSelect)

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

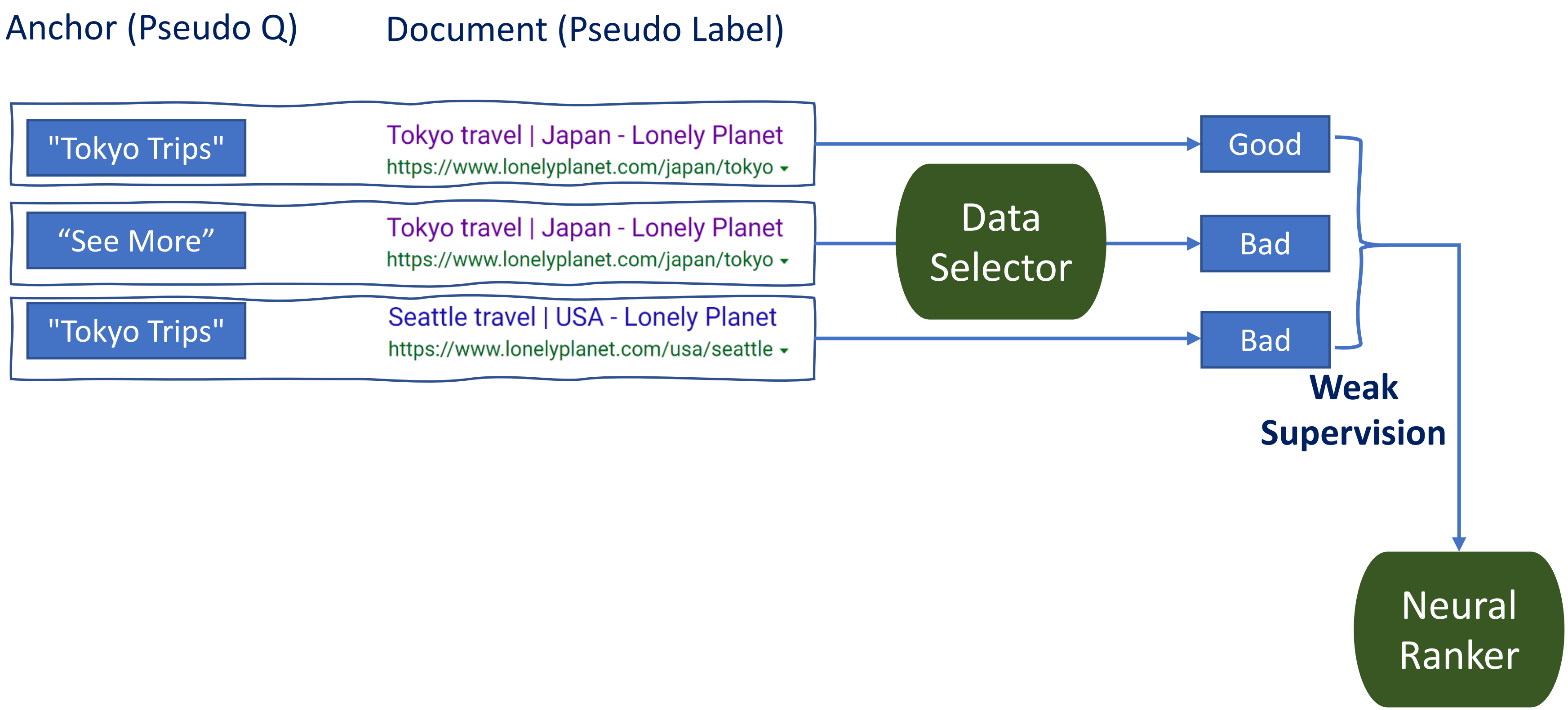
- Reinforcement data selection (ReinfoSelect)



Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

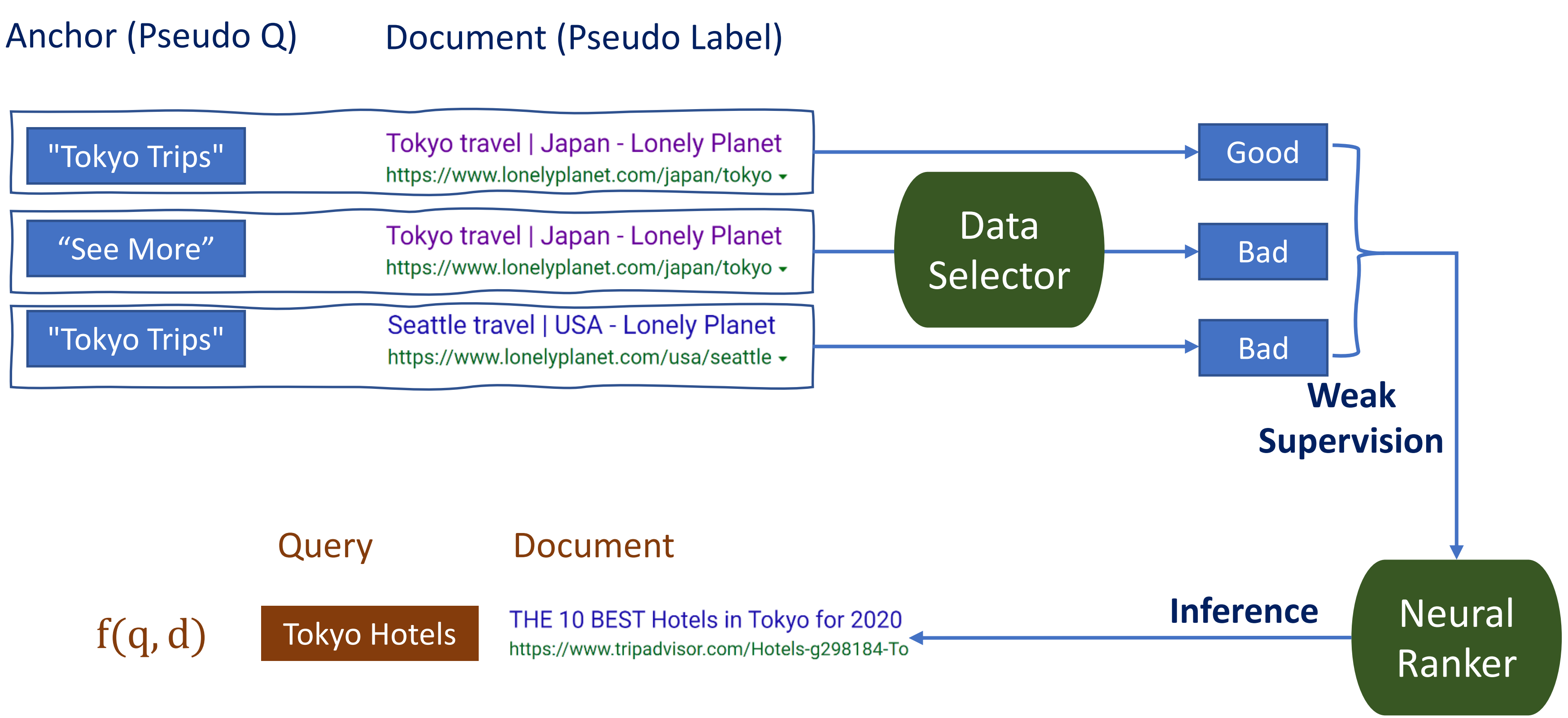
- Reinforcement data selection (ReinfoSelect)



Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

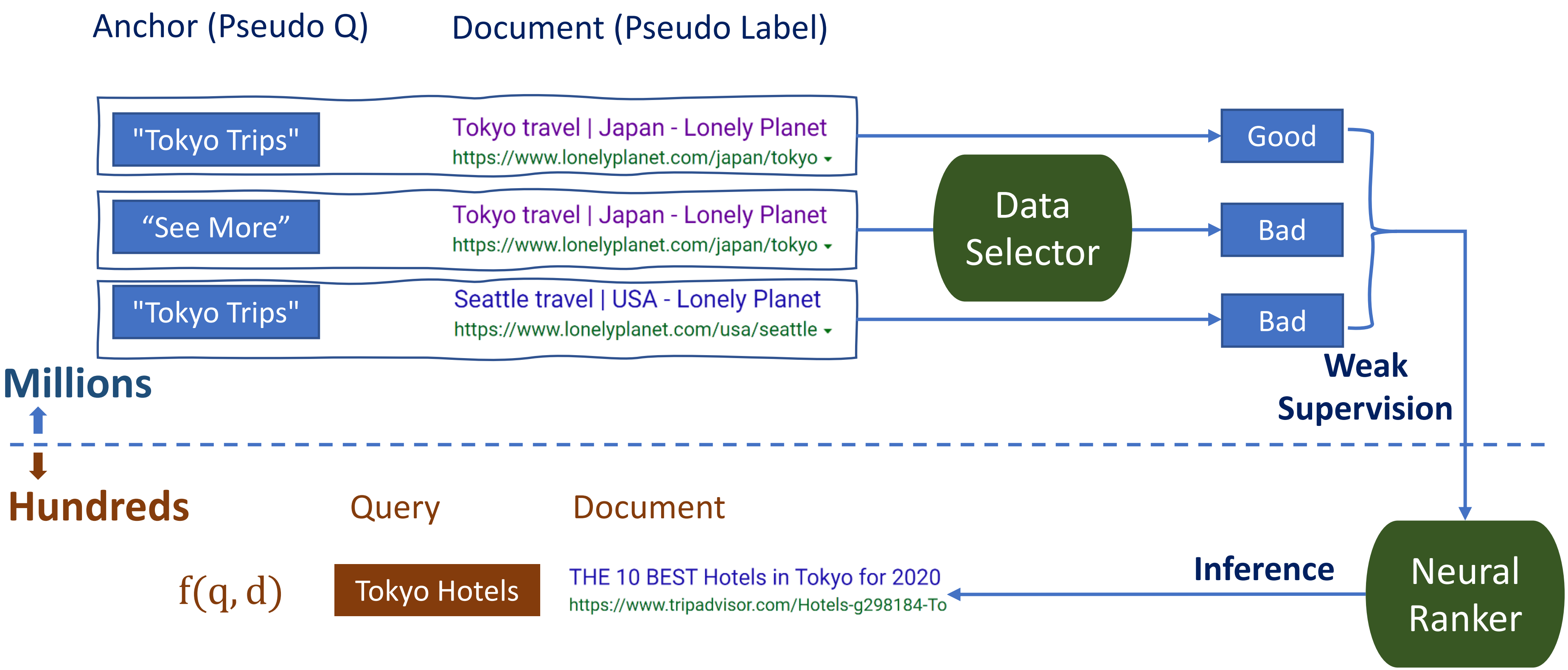
- Reinforcement data selection (ReinfoSelect)



Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

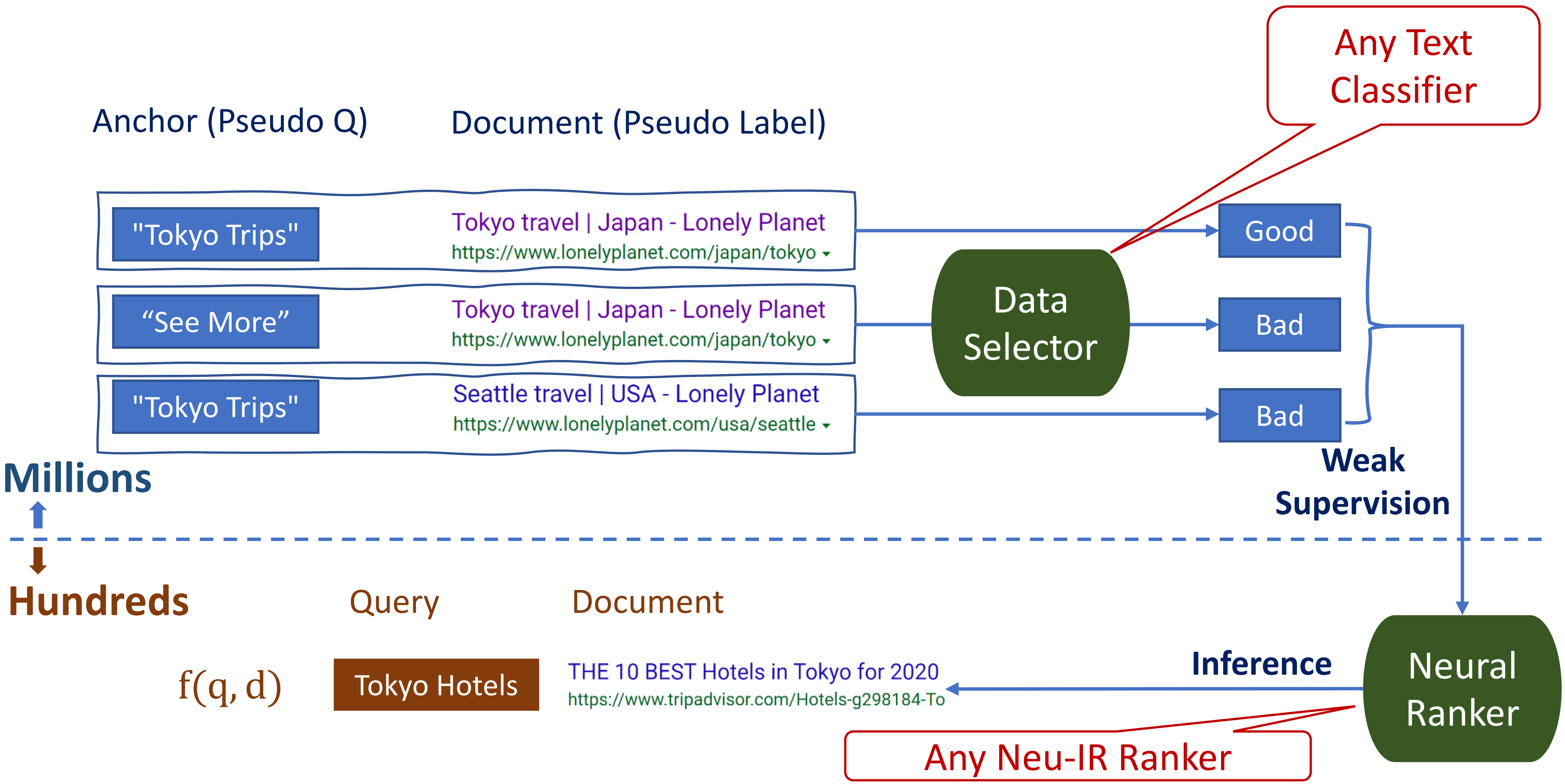
- Reinforcement data selection (ReinfoSelect)



Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

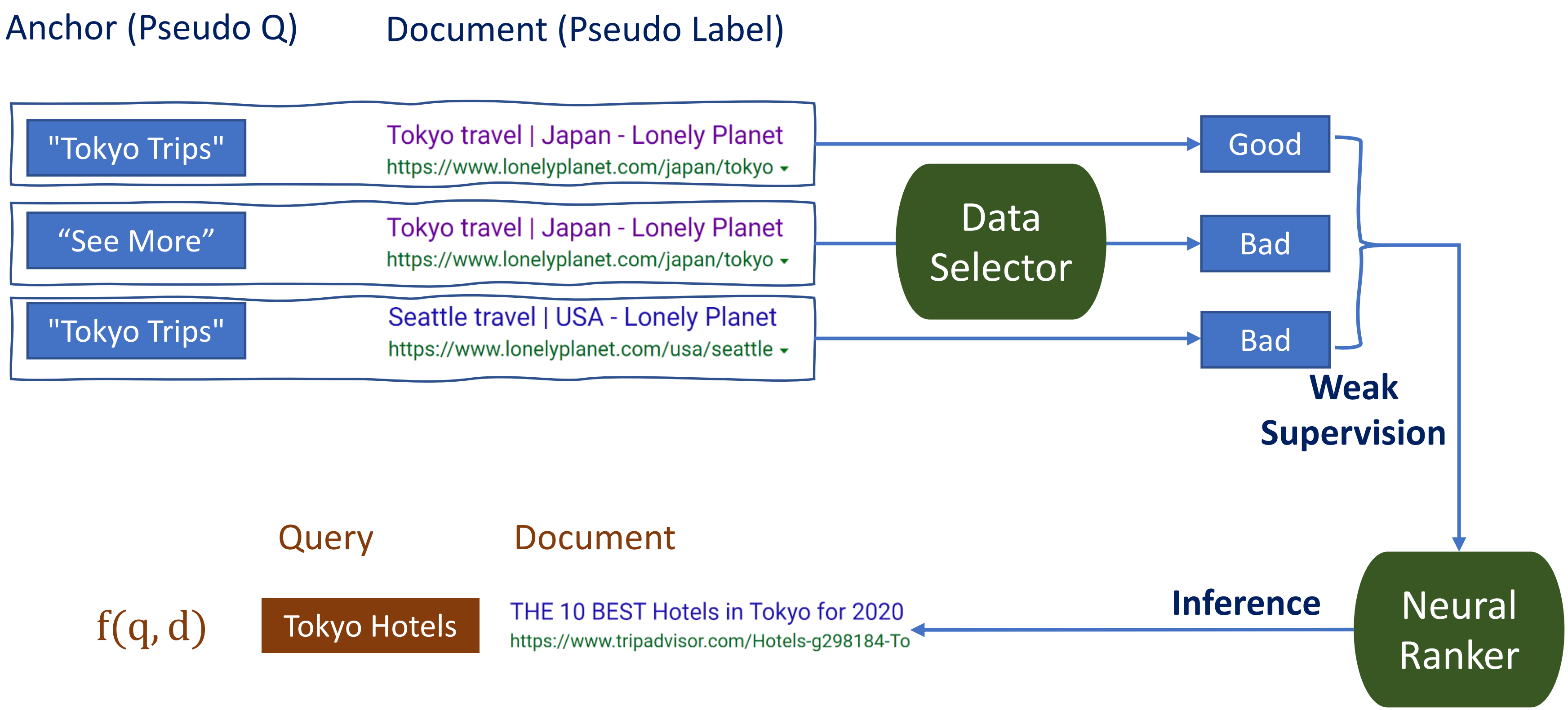
- Reinforcement data selection (ReinfoSelect)



Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

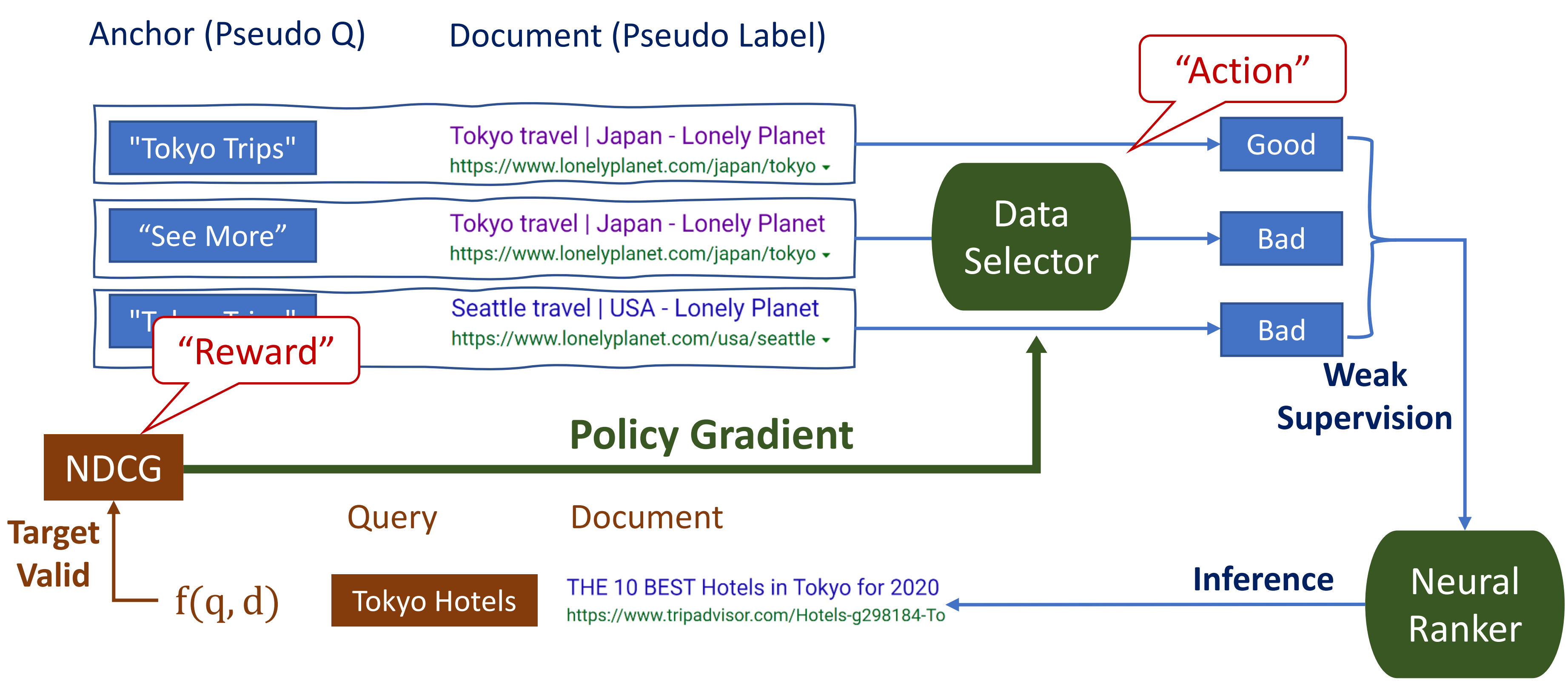
- Reinforcement data selection (ReinfoSelect)



Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

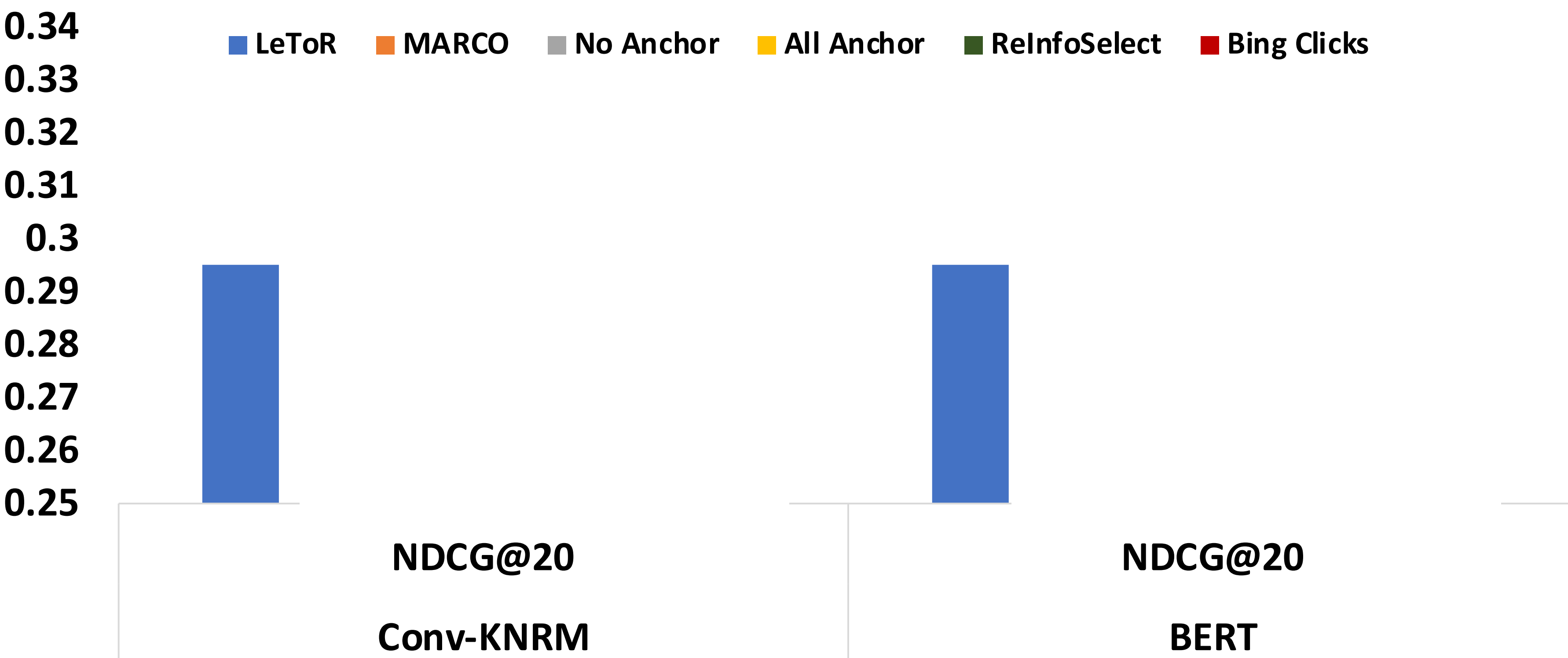
- Reinforcement data selection (ReinfoSelect)
  - Policy gradient [Ronald J. Williams. 1992] is used



Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- Reinforcement data selection (ReinfoSelect)

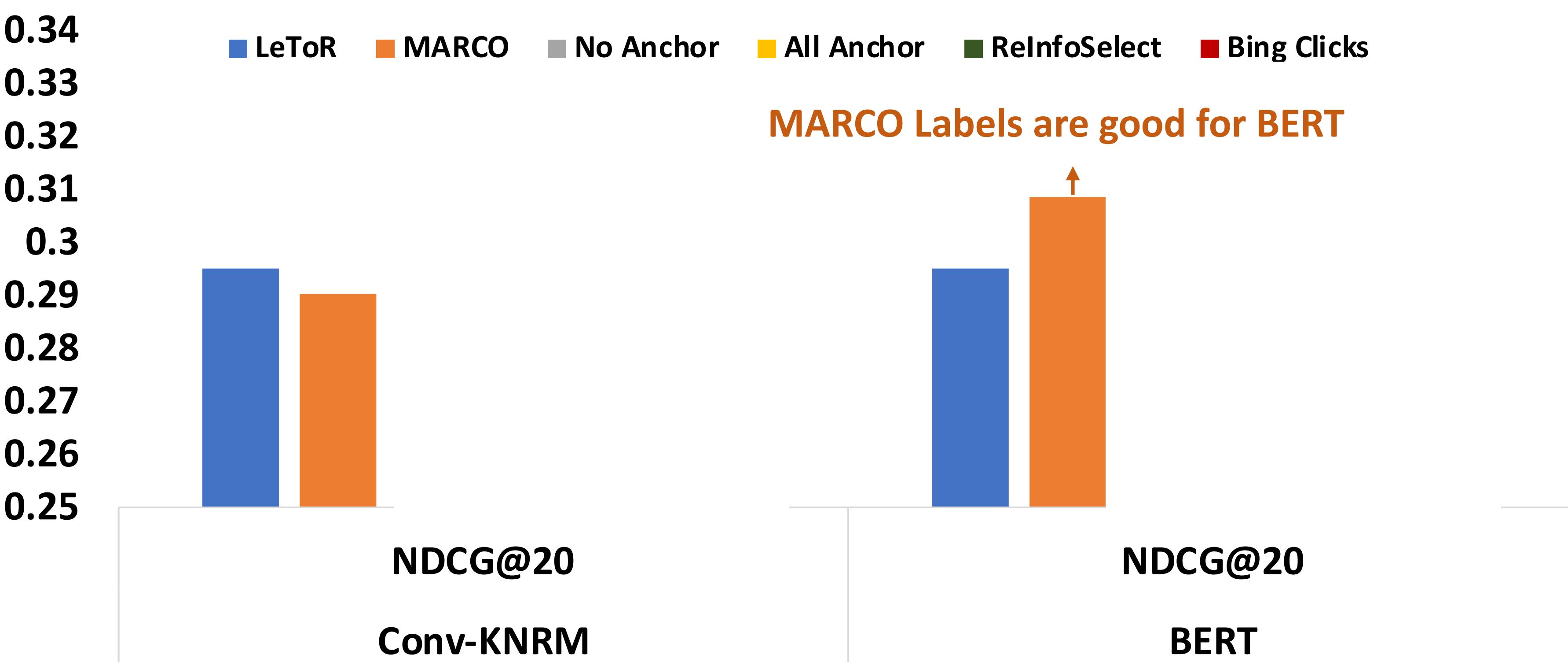


Same results on ClueWeb12 and Robust04

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- Reinforcement data selection (ReinfoSelect)

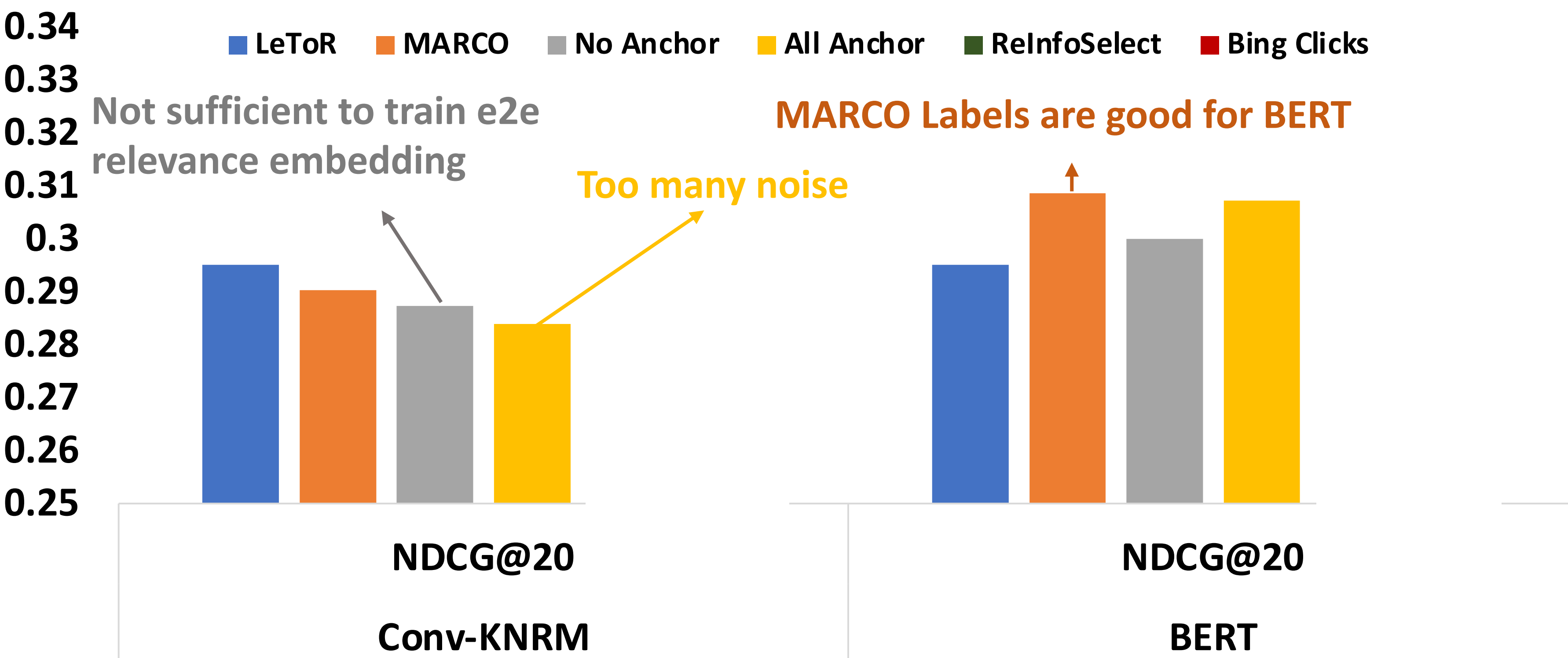


Same results on ClueWeb12 and Robust04

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- Reinforcement data selection (ReinfoSelect)



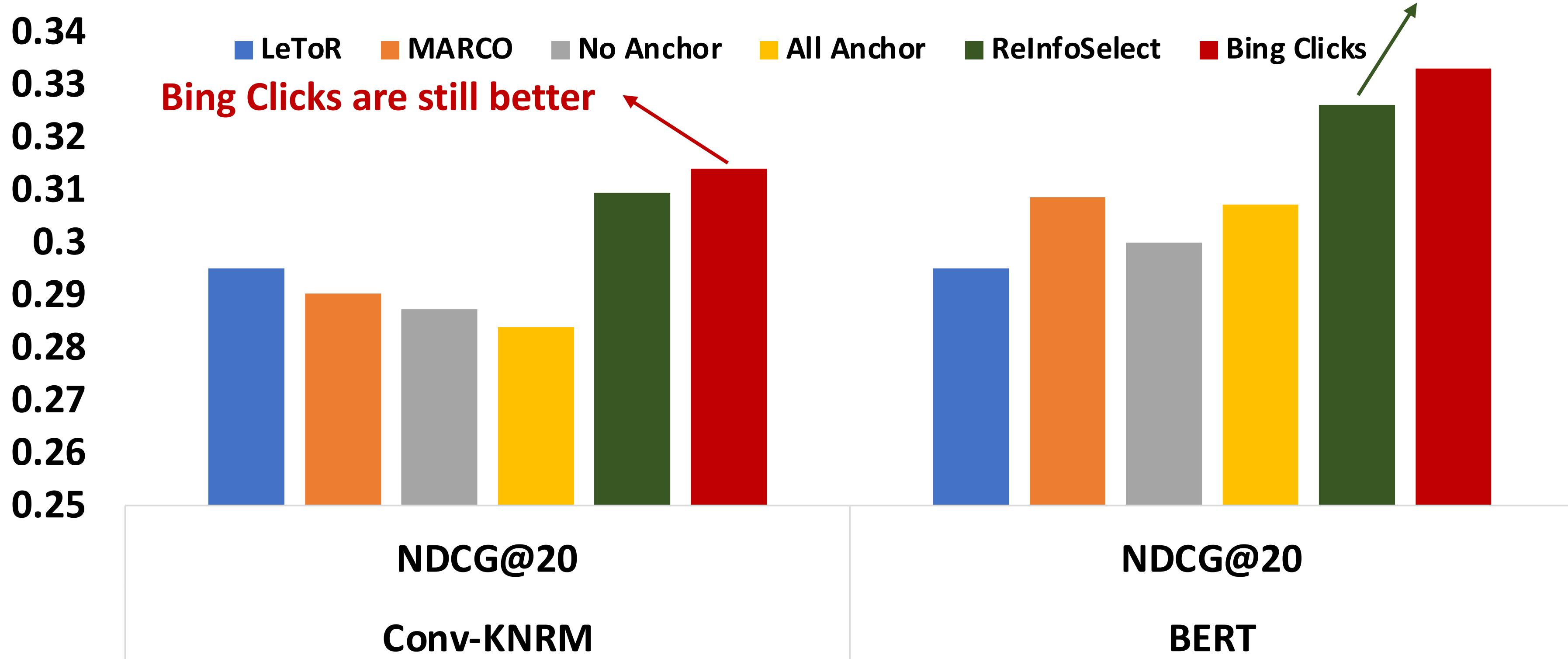
Same results on ClueWeb12 and Robust04

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- Reinforcement data selection (ReinfoSelect)

ReInfoSelect alleviates the necessity of *One Millions Labels* or *Search Log*.



Same results on ClueWeb12 and Robust04

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- Reinforcement data selection (ReinfoSelect)

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- Reinforcement data selection (ReinfoSelect)

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- Reinforcement data selection (ReinfoSelect)
  - Some selected cases
    - One row is TREC queries and the other is selected anchors.
    - Can you tell?

Query	Anchor
dieting	crash dieting
french lick resort and casino	tropicana casino & resort atlantic city
diabetes education	vegan menu for people with diabetes
income tax return online	personal income taxes
orange county convention center	orange county convention center

Zhang et al., Selective Weak Supervision for Neural Information Retrieval. WWW 2020.

Challenges in Neural IR

- However Anchor-Document data is only available in the Web domain
  - E.g. TREC COVID contains only 50 labeled queries

Enterprise Search

Extreme Verticals

Cloud Search

Personalized Search

IR Community

Challenges in Neural IR

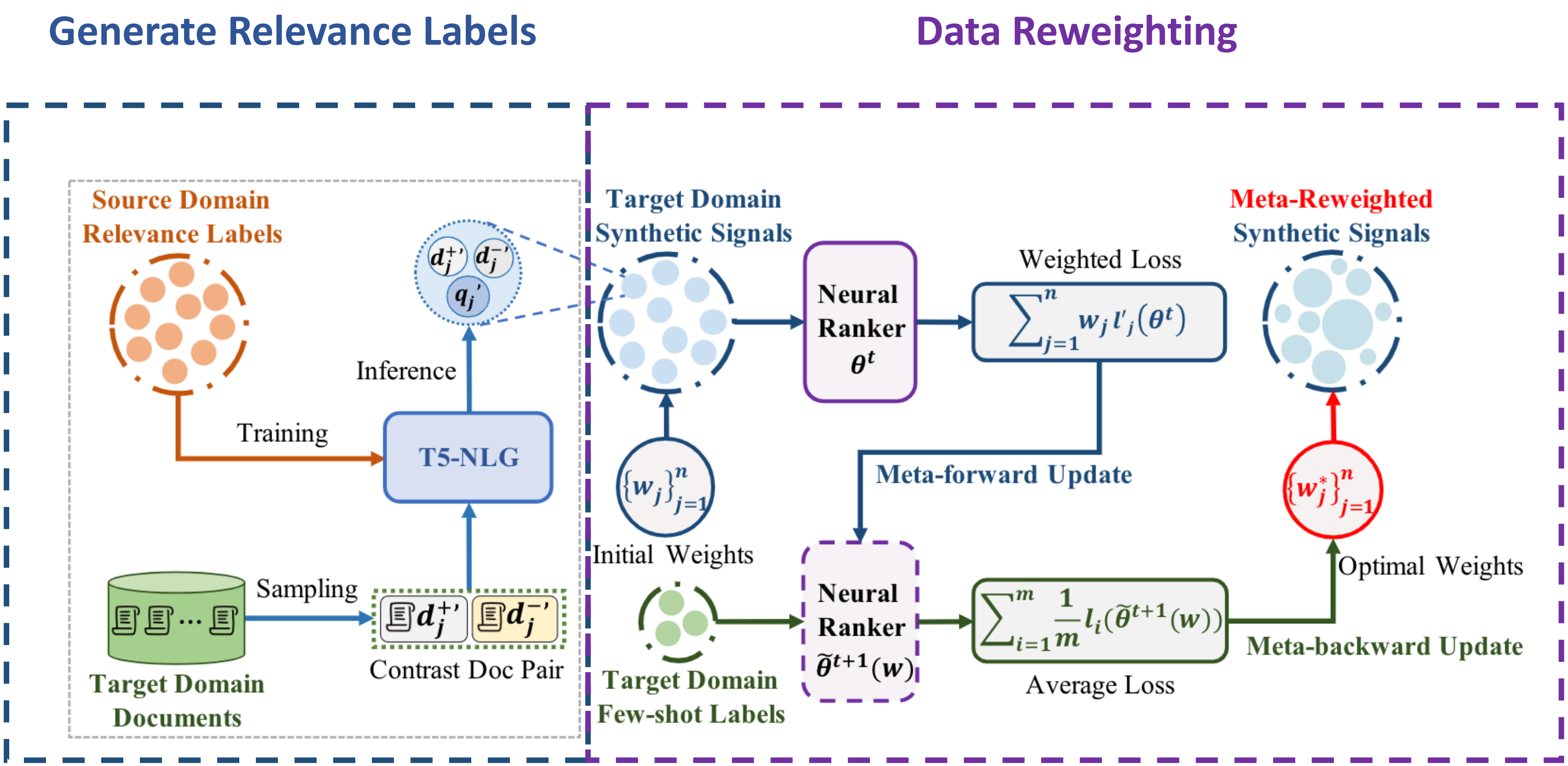
- However Anchor-Document data is only available in the Web domain
  - E.g. TREC COVID contains only 50 labeled queries

Can we generate some relevance labels for different ranking scenarios ?

Enterprise Search	Extreme Verticals	Cloud Search	Personalized Search	IR Community

Challenges in Neural IR

- MetaAdaptRank



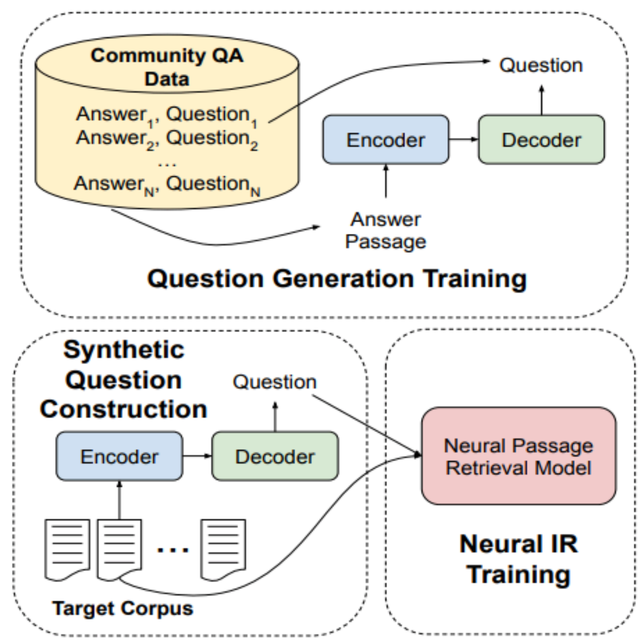
Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

- Generate relevance labels
  - Neu-IR models are trained with relevance labels (q, d^+, d^-)
  - Generate pseudo query with a Query Generator (QG)

Train generator with large-scale corpus in general domain

Generate query q for document d of the target domain



Ma et al., Zero-shot Neural Retrieval via Domain-targeted Synthetic Query Generation. 2020.

Challenges in Neural IR

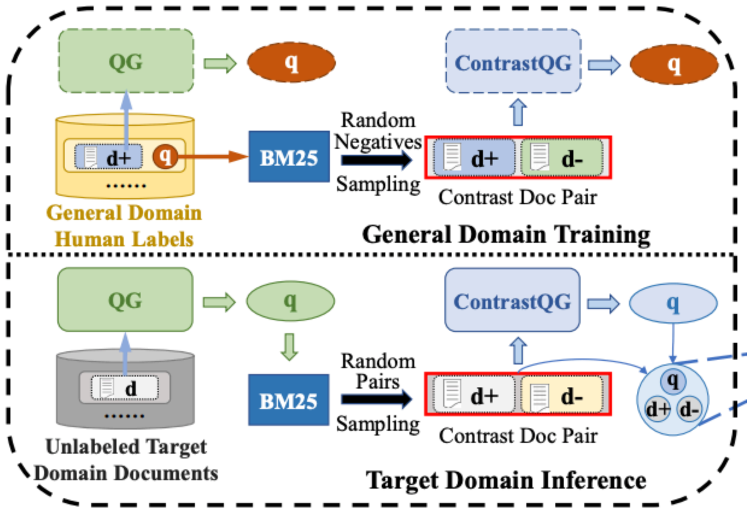
- Generate relevance labels
  - Using this method to generate some queries
    - The generated queries are too general
    - These queries may be related with multi-documents
    - It is hard to select the negative documents for training

`SyncSup:` covid outbreak **symptoms**	The importance of the timing of **quarantine** measures before **symptom** onset to **prevent** COVID-19 outbreaks how **quarantine**-based measures can **prevent** or suppress an outbreak ...	Furthermore, the effect of infectiousness prior to **symptom** onset combined with a significant proportion we evaluate two procedures: monitoring individuals for **symptoms** onset ...

Ma et al., Zero-shot Neural Retrieval via Domain-targeted Synthetic Query Generation. 2020.

Challenges in Neural IR

- Generate better relevance labels
  - Using two contrastive documents to generate a query (ContrastQG)
    - Generate query q^* with QG for document d
    - Select two confused documents d^+ and d^- according to q^*
    - Generate q with d^+ and d^-, and synthesis relevance label (q, d^+, d^-)



Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

- Generate better relevance labels
  - Using two contrastive documents to generate a query (ContrastQG)

Synthetic Methods	BLEU-1	BLEU-2	ROUGE-1	ROUGE-2	ROUGE-L	NIST@1	NIST@2	METEOR
SyncSup (Ma et al., 2020)	0.5672	0.4527	0.5928	0.3764	0.5745	5.8070	7.3315	0.3089
Reverse-CTSyncSup	0.3185	0.1807	0.3528	0.1088	0.3395	3.0076	3.3665	0.1610
CTSyncSup	**0.5909**	**0.4627**	**0.6238**	**0.3844**	**0.5955**	**6.1282**	**7.6314**	**0.3191**

Supervision Sources		ClueWeb09-B (Web)		Robust04 (News)		TREC-COVID (BioMed)	
		NDCG@20	ERR@20	NDCG@20	ERR@20	NDCG@20	P@20
(a)	MS MARCO (Nguyen et al., 2016)	$0.3205^{\flat}$	0.1690	$0.4674^{\ddagger}$	$0.1304^{\ddagger}$	$0.8054^{\ddagger}$	$0.8610^{\ddagger}$
(b)	Anchor (Zhang et al., 2020b)	0.3072	0.1609	0.4449	0.1223	0.7677	0.8260
(c)	SyncSup (Ma et al., 2020)	0.3036	0.1602	$0.4685^{\ddagger}$	$\mathbf{0.1311}^{\ddagger}$	0.7867	0.8470
(d)	CTSyncSup	0.3123	$\mathbf{0.1764}^{\flat}$	$\mathbf{0.4769}^{\ddagger}$	$0.1293^{\ddagger}$	$0.8006^{\ddagger}$	$0.8610^{\ddagger}$
(e)	MARCO + CTSyncSup	$\mathbf{0.3214}^{\flat}$	$0.1739^{\ddagger\flat}$	$0.4727^{\ddagger}$	$0.1297^{\ddagger}$	$\mathbf{0.8182}^{\ddagger\flat}$	$\mathbf{0.8720}^{\ddagger\flat}$

Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

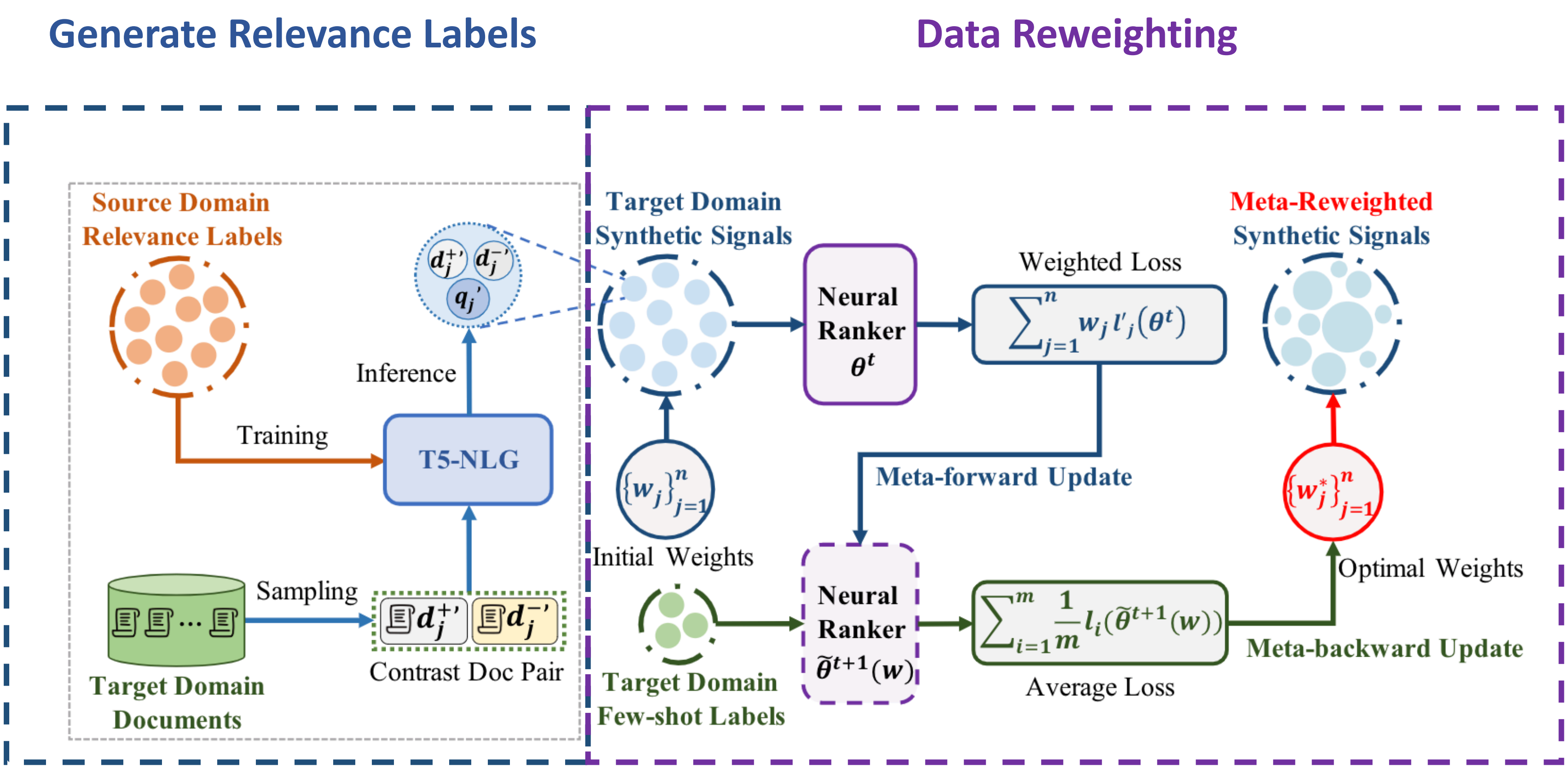
- Generate better relevance labels
  - Using two contrastive documents to generate a query (ContrastQG)

Synthetic Query		Positive Document	Negative Document
1 (↑)	`CTSyncSup`: us military **radars** in **colombia** `SyncSup`: what is the **pentagon**	... One month ago, the **Pentagon** issued an order to suspend operations of the two **radars** that detect aircraft. These **radars** operate in **Colombia** as a result of that agreement. ...	... provide for more funding and retain more forces than the $1.5-trillion five-year budget Cheney presented to Congress in January, **Pentagon** officials say ...
2 (↑)	`CTSyncSup`: what percent of the economy was **increased** in **1993** `SyncSup`: what is the economic issue in **peru**	... This letter explains the **Peruvian** Government's economic policy. The development of the economy in **1993** was in general much better. It is estimated that the real GDP has **increased** by 7 percent ...	... Only three economies - Guyana, Argentina and **Peru** - grew by more than than 5 per cent this year, with **Peru** expanding by 11 per cent...

Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

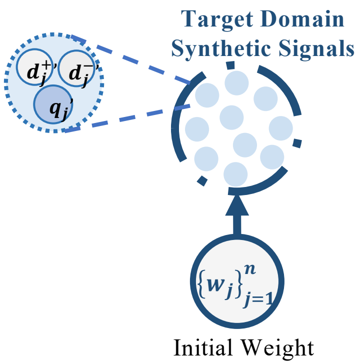
- MetaAdaptRank



Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

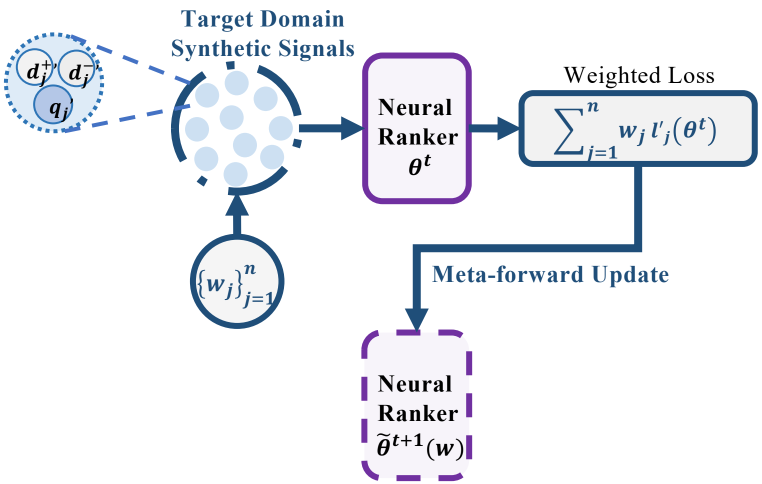
- Reweight relevance labels
  - Assign initial weights to relevance labels



Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

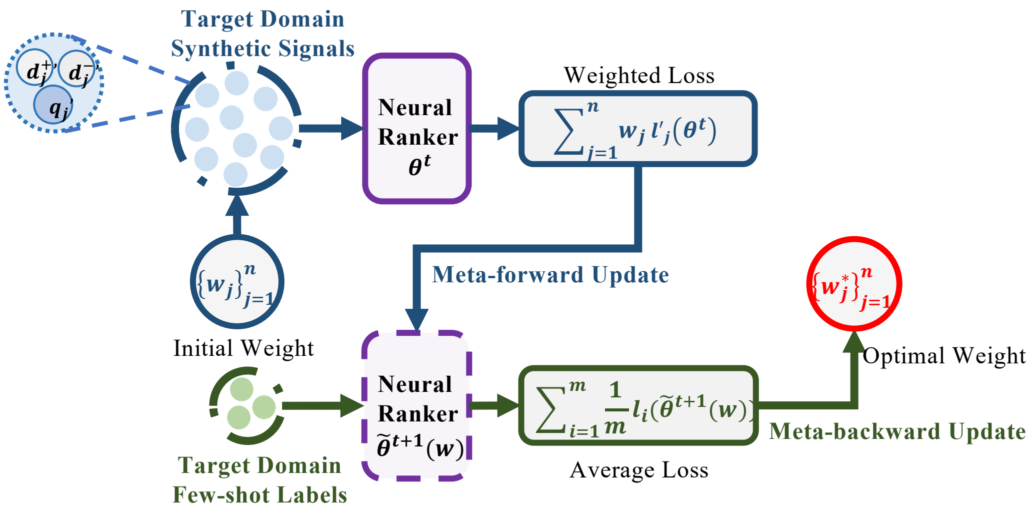
Challenges in Neural IR

- Reweight relevance labels
  - Assign initial weights to relevance labels
  - Meta-forward Update: Pseudo update Neu-IR models



Challenges in Neural IR

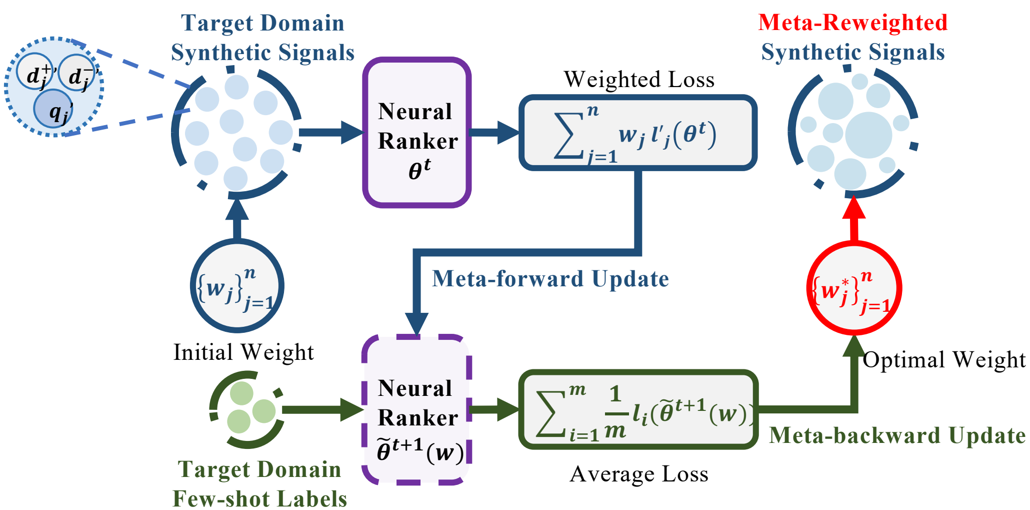
- Reweight relevance labels
  - Assign initial weights to relevance labels
  - Meta-forward Update: Pseudo update Neu-IR models
  - Meta-backward Update: Calculate the actual weights



Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

- Reweight relevance labels
  - Assign initial weights to relevance labels
  - Meta-forward Update: Pseudo update Neu-IR models
  - Meta-backward Update: Calculate the actual weights
  - Train Neu-IR model with meta-reweighted synthetic signals



Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

- Reweight relevance labels
    - Performance

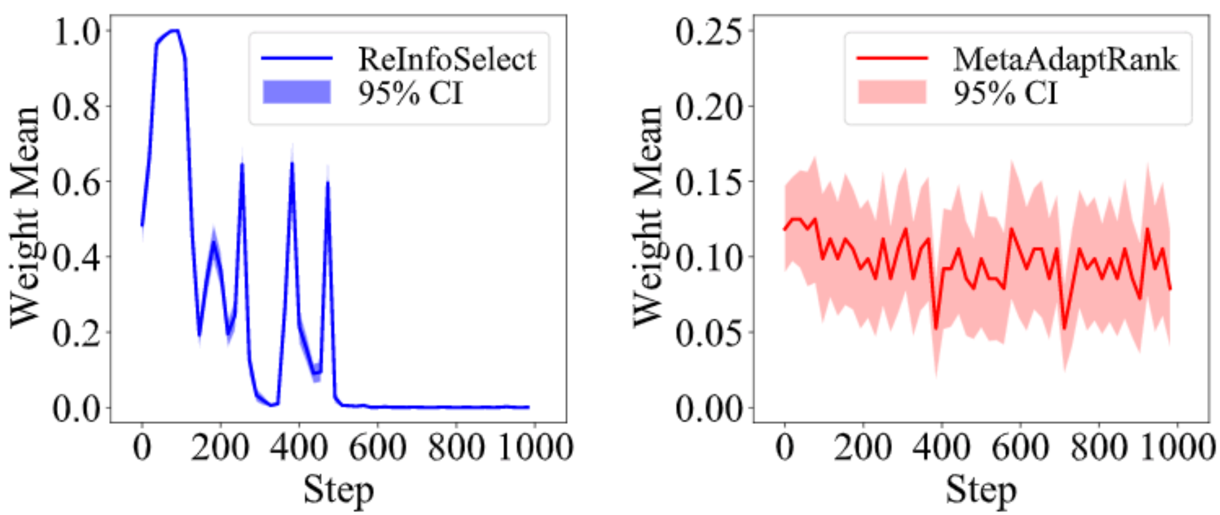
Methods (Supervision Sources)		ClueWeb09-B (Web) NDCG@20	ClueWeb09-B (Web) ERR@20	Robust04 (News) NDCG@20	Robust04 (News) ERR@20	TREC-COVID (BioMed) NDCG@20	TREC-COVID (BioMed) P@20
(a)	ReInfoSelect (MS MARCO)	0.3294	0.1760	0.4756	0.1291	$0.8229^{\ddagger}$	$0.8780^{\ddagger}$
(b)	ReInfoSelect (Anchor)	0.3261	0.1669	0.4703	0.1313	0.7891	0.8430
(c)	ReInfoSelect (CTSyncSup)	0.3243	0.1742	$0.4816^{\ddagger}$	0.1334	$0.8230^{\ddagger}$	$0.8800^{\ddagger}$
(d)	MetaAdaptRank (MS MARCO)	$0.3453^{\dagger\ddagger\flat}$	$\mathbf{0.2018}^{\dagger\ddagger\flat\sharp}$	$0.4853^{\ddagger}$	0.1331	$0.8354^{\ddagger\sharp}$	$0.8730^{\ddagger}$
(e)	MetaAdaptRank (Anchor)	0.3374	0.1730	0.4797	0.1314	0.8045	0.8650
(f)	MetaAdaptRank (CTSyncSup)	$0.3416^{\flat}$	$0.1893^{\ddagger\sharp}$	$0.4916^{\dagger\ddagger\sharp}$	$0.1362^{\dagger\sharp}$	$0.8378^{\ddagger\sharp}$	$0.8790^{\ddagger}$
(g)	MetaAdaptRank (MARCO + CTSyncSup)	$\mathbf{0.3498}^{\dagger\ddagger\sharp}$	$0.1926^{\ddagger\flat\sharp}$	$\mathbf{0.4989}^{\dagger\ddagger\flat\natural\sharp}$	$\mathbf{0.1366}^{\dagger\natural}$	$\mathbf{0.8488}^{\dagger\ddagger\flat\natural\sharp}$	$\mathbf{0.8910}^{\ddagger\natural\sharp}$

Table 5: Ranking accuracy of ReInfoSelect and MetaAdaptRank using different supervision sources. Superscripts $\dagger, \ddagger, \flat, \natural, \sharp, \S$ indicate statistically significant improvements over *(a)*†, *(b)*‡, *(c)*$^{\flat}$, *(d)*$^{\natural}$, *(e)*$^{\sharp}$ and *(f)*§.

Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

- Reweight relevance labels
  - Performance
    - MetaAdaptRank assigns more fine-grained weights to weak supervision



(a) ReInfoSelect. (b) MetaAdaptRank.

Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Challenges in Neural IR

- Data Synthesis with Data Reweighting
  - Performance

Methods	ClueWeb09-B (Web) NDCG@20	ClueWeb09-B (Web) ERR@20	Robust04 (News) NDCG@20	Robust04 (News) ERR@20	TREC-COVID (BioMed) NDCG@20	TREC-COVID (BioMed) P@20
BM25 (Yang et al., 2017)	0.2773	0.1426	0.4129	0.1117	0.6979	0.7670
SDM (Dai and Callan, 2019)	0.2774	0.1380	0.4269	0.1172	0.7030	0.7770
RankSVM (Dai and Callan, 2019)	0.289	n.a.	0.420	n.a.	n.a.	n.a.
RankSVM (OpenMatch)	0.2825	0.1476	0.4309	0.1173	0.6995	0.7570
Coor-Ascent (Dai and Callan, 2019)	0.295	n.a.	0.427	n.a.	n.a.	n.a.
Coor-Ascent (OpenMatch)	$0.2969^{\dagger}$	$0.1581^{\dagger}$	$0.4340^{\dagger}$	0.1171	0.7041	0.7770
Few-shot Supervision (Zhang et al., 2020b)	0.2999	0.1631	0.4258	0.1163	n.a.	n.a
Few-shot Supervision (Ours)	$0.3033^{\dagger}$	0.1519	$0.4572^{\dagger\ddagger}$	0.1234	$0.7713^{\dagger\ddagger}$	$0.8400^{\dagger\ddagger}$
Bing User Click (Dai and Callan, 2019)	0.333	n.a.	n.a.	n.a.	n.a.	n.a.
MS MARCO (Nguyen et al., 2016)	$0.3205^{\dagger\ddagger\flat\S}$	$0.1690^{\dagger\flat}$	$0.4674^{\dagger\ddagger}$	$0.1304^{\dagger\ddagger\flat}$	$0.8054^{\dagger\ddagger\flat}$	$0.8610^{\dagger\ddagger\flat}$
Title Filter (MacAvaney et al., 2019b)	0.3021	0.1513	0.4379	0.1202	n.a.	n.a.
Anchor (Zhang et al., 2020b)	$0.3072^{\dagger}$	$0.1609^{\dagger}$	$0.4449^{\dagger\ddagger}$	$0.1223^{\dagger\ddagger}$	$0.7677^{\dagger\ddagger}$	$0.8260^{\dagger\ddagger}$
ReInfoSelect (Zhang et al., 2020b)	$0.3261^{\dagger\ddagger\flat\S}$	$0.1669^{\dagger\flat}$	$0.4703^{\dagger\ddagger\flat}$	$0.1313^{\dagger\ddagger\flat}$	$0.7833^{\dagger\ddagger}$	$0.8420^{\dagger\ddagger}$
SyncSup (Ma et al., 2020)	$0.3036^{\dagger}$	$0.1602^{\dagger}$	$0.4685^{\dagger\ddagger}$	$0.1311^{\dagger\ddagger\flat}$	$0.7867^{\dagger\ddagger}$	$0.8470^{\dagger\ddagger}$
CTSyncSup	$0.3123^{\dagger}$	$0.1764^{\dagger\flat\S}$	$0.4769^{\dagger\ddagger\flat}$	$0.1293^{\dagger\ddagger\flat}$	$0.8006^{\dagger\ddagger\flat}$	$0.8610^{\dagger\ddagger}$
MetaAdaptRank	**0.3416**$^{\dagger\ddagger\flat\natural\S}$	**0.1893**$^{\dagger\ddagger\flat\natural\sharp\S}$	**0.4916**$^{\dagger\ddagger\flat\natural\sharp\S}$	**0.1362**$^{\dagger\ddagger\flat\natural\S}$	**0.8378**$^{\dagger\ddagger\flat\natural\sharp\S}$	**0.8790**$^{\dagger\ddagger\flat\sharp\S}$

Table 2: Ranking accuracy of MetaAdaptRank and baselines. $\dagger, \ddagger, \flat, \natural, \sharp, \S$ indicate statistically significant improvements over SDM†, Coor-Ascent‡, Few-shot Supervision$^{\flat}$, MS MARCO$^{\natural}$, ReInfoSelect$^{\sharp}$ and SyncSup§.

Sun et al., Meta Adaptive Neural Ranking with Contrastive Synthetic Supervision. 2020.

Outline

- Introduction to IR
- Traditional IR Models
- Neural IR Models
- Neural Models in IR Systems
- Challenges in Neural IR
- Summary

Summary

- Neu-IR models conducts semantic match to deal with vocabulary mismatch problem
- Neu-IR models can be applied in both retrieval and reranking stages
- Neu-IR models need well training
  - Existing pretraining methods may be not suitable for IR
  - Lots of few-shot ranking scenarios lack training data

Summary

- OpenMatch Tookit
  - https://github.com/thunlp/OpenMatch

MS MARCO Document Ranking Leaderboard

date	description	team
2021/03/24	PROP_step400K base + doc2query top1000(single)	Yingyan Li, Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Y
2021/04/01	PROP_step400K base + doc2query top1000(ensemble v0.1)	Yingyan Li, Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Y
2021/01/02 🏆	PROP_step400K base (ensemble v0.1)	Yingyan Li, Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Y
2021/01/20	PROP_step400K base, long query + doc2query top100 (single)	Yingyan Li, Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Y
2020/12/16	PROP_step400k base + doc2query top100 (single)	Yingyan Li, Xinyu Ma - ICT, CAS
2020/10/28	Bert-ranker (our implementation)	Yingyan Li, Xinyu Ma - ICT, CAS
2021/02/10 🏆	DML	Xuanyu Zhang - AI-Lab, DXM
2021/03/12	ANCE MaxP	XJTU
2021/03/02	ANCE FirstP	XJTU
2020/11/18	PyTerrier framework + DPH Divergence from Randomness model, with stemming	University of Glasgow Terrier Team
2021/03/30	ANCE+HDCT+BERT pretrained(ensemble)	TJ-university
2020/11/13	ANCE + BERT Base MaxP	THU-MSR

README.md

OpenMatch

An Open-Source Package for Information Retrieval.

😃 What's New

- Top Spot on TREC-COVID Challenge (May 2020, Round2)

  The twin goals of the challenge are to evaluate search algorithms and systems for helping scientists, clinicians, policy makers, and others manage the existing and rapidly growing corpus of scientific literature related to COVID-19, and to discover methods that will assist with managing scientific information in future global biomedical crises.
  >> Reproduce Our Submit >> About COVID-19 Dataset >> Our Paper

Overview

OpenMatch integrates excellent neural methods and technologies to provide a complete solution for deep text matching and understanding.

> 1/ Document Retrieval

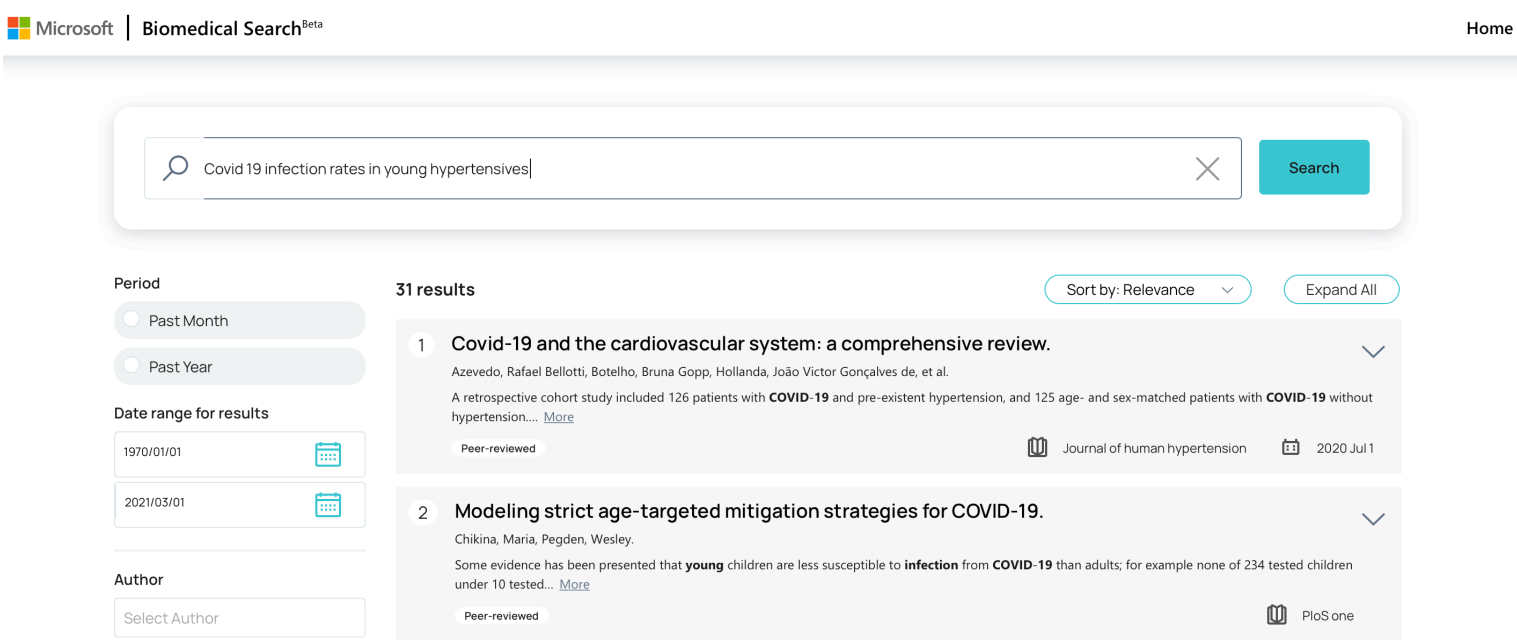
Document Retrieval refers to extracting a set of related documents from large-scale document-level data based on user queries.

*** Sparse Retrieval**

OpenMatch provides some valuable experimental results for researchers

Summary

- BioMedical Search
  - We achieve the fist place in the TREC COVID round 2
  - Our method is used in Microsoft Biomedical Search



https://biomedsearch.microsoft.com/en-us/
https://blogs.microsoft.com/ai-for-business/biomedical-search/

Q&A

THUNLP