Neural Information Retrieval

Prepared by Diana Inkpen, University of Ottawa, 2021,

(partly based on Pretrained Transformers for Text Ranking:

BERT and Beyond, by Jimmy Lin, Rodrigo Nogueira, and Andrew Yates, 2020

Neural IR systems

- Pre-BERT models
- Using BERT-like models

Corpus	$\|C\|$	$\overline{L}(C)$
MS MARCO passage corpus	8,841,823	57.3
MS MARCO document corpus	3,213,835	1128.7
Robust04 corpus (TREC disks 4&5)	528,155	530.2

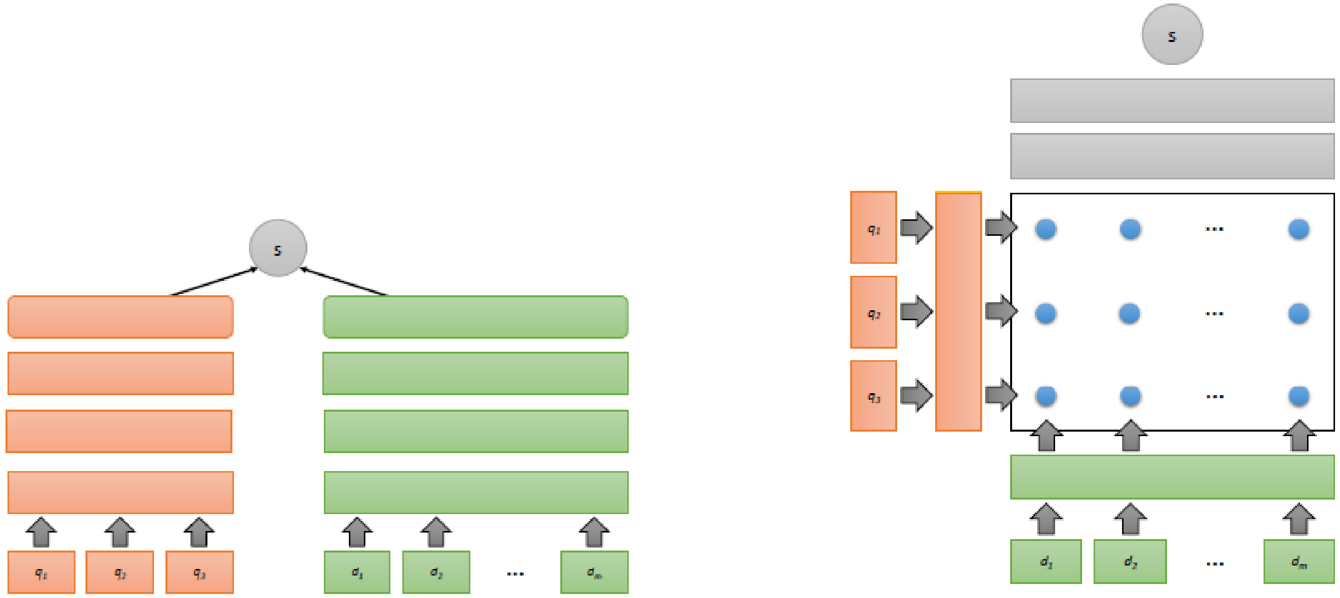
- Three corpora: size of the collection and average document length.
- The MS MARCO document corpus was also used for TREC 2019 Deep Learning Track document retrieval task.
- The MS MARCO passage corpus was also used for the TREC 2019 Deep Learning Track
- passage retrieval task. Passage relevance taken from document relevance.
- Training, development and test queries. Large number of queries.

Large collections, many queries

Dataset	$\|q\|$	$\overline{L}(q)$	$\|J\|$	$\|J\|/q$	$\|\text{Rel}\|/q$
MS MARCO passage retrieval (train)	502,939	6.06	532,761	1.06	1.06
MS MARCO passage retrieval (development)	6,980	5.92	7,437	1.07	1.07
MS MARCO passage retrieval (test)	6,837	5.85	-	-	-
MS MARCO document retrieval (train)	367,013	5.95	367,013	1.0	1.0
MS MARCO document retrieval (development	5,193	5.89	5,193	1.0	1.0
MS MARCO document retrieval (test)	5,793	5.85	-	-	-
TREC 2019 DL passage	43	5.39	9,260	215.4	95.4
TREC 2019 DL document	43	5.51	16,258	378.1	153.4
Robust04	249	(title) 2.7 (narr.) 15.3 (desc.) 40.2	311,410	1250.6	69.9

- Size of the set of evaluation topics, in terms of the number of queries and the average length of each query L(q).
- The amount of relevance judgments available, in terms of positive and negative labels. Average number of judgments per query, and the number of relevant labels per query.

Pre-BERT models



(a) a generic representation-based neural ranking model (b) a generic interaction-based neural ranking model

Representation-based models (left)

- independently learn vector representations of query and documents that can be compared to compute
- relevance scores using simple metrics such as cosine similarity.

Interaction-based models (right)

- explicitly model term interactions in a similarity matrix that undergoes further processing to arrive at
- a relevance score.

Using BERT for IR

Method		MS MARCO Passage Development MRR@10	MS MARCO Passage Test MRR@10
BM25 (Microsoft Baseline)		0.167	0.165
IRNet (Deep CNN/IR Hybrid Network)	January 2nd, 2019	0.278	0.281
BERT [Nogueira and Cho, 2019]	January 7th, 2019	0.365	0.359

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

by Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova,

Oct 2018, Google

Large pre-trained language model, available for download, started to be used in many NLP applications.

IR evaluation measures

- MAP – mean average precision over all queries
- P@10 – precision in the first 10 retrieved
- Mean Reciprocal Rank – mean over all queries of Reciprocal Rank (RR)
  - RR (q) = 1 / rank i

  where ranki is the smallest rank number of a relevant document.
  - if a relevant document appears in the first position, reciprocal rank = 1, 1/2 if it appears in the second position, 1/3 if it appears in the third position, etc.
- Normalized Discounted Cumulative Gain (nDCG)
  - used to measure the quality of web search results
  - designed for graded relevance judgements
  - https://en.wikipedia.org/wiki/Discounted_cumulative_gain

Mean Average Precision (MAP score)

- Mean average precision for a set of Q queries is the mean of the average precision scores for each query (uninterpolated).
- MAP $= \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$

Computing Recall/Precision Points

- For a given query, produce the ranked list of retrievals.
- Adjusting a threshold on this ranked list produces different sets of retrieved documents, and therefore different recall/precision measures.
- Mark each document in the ranked list that is relevant according to the gold standard.
- Compute a recall/precision pair for each position in the ranked list that contains a relevant document.

Computing Recall/Precision Points: An Example

n	doc #	relevant
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

Let total # of relevant docs = 6
Check each new recall point:

R=1/6=0.167; P=1/1=1

R=2/6=0.333; P=2/2=1

R=3/6=0.5; P=3/4=0.75

R=4/6=0.667; P=4/6=0.667

R=5/6=0.833; p=5/13=0.38

Missing one relevant document. Never reach 100% recall

Average Precision

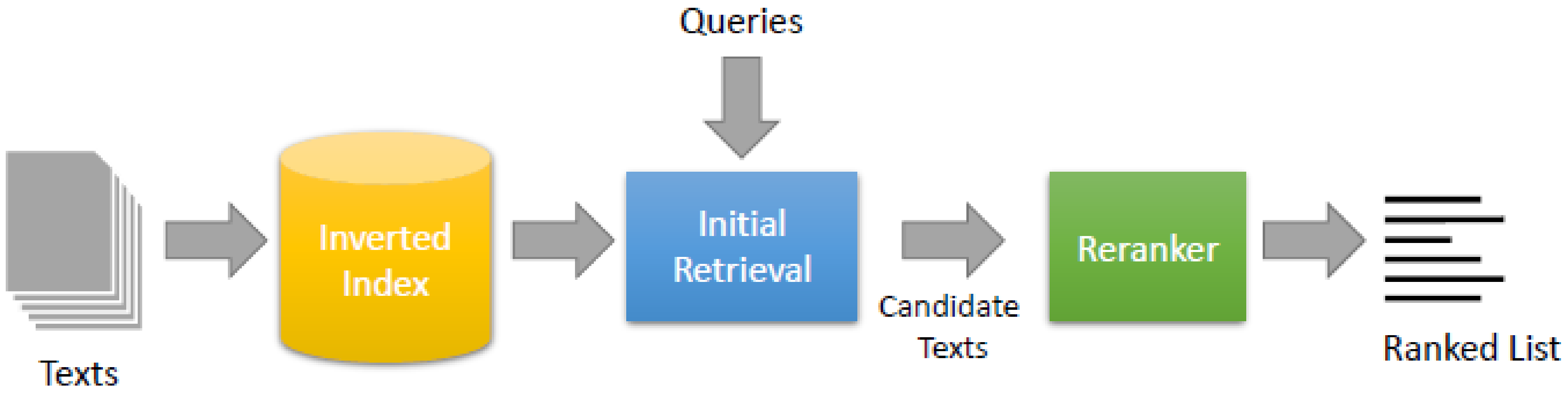
- $\text{AveP} = \frac{\sum_k P(k) * rel(k)}{number\ of\ relevant\ documents}$
- rel(k) is an indicator function equaling 1 if the item at rank k is a relevant document, zero otherwise.

For the previous query:

AveP = (1+1+0.75+0.667+0.38)/6 = 0.632

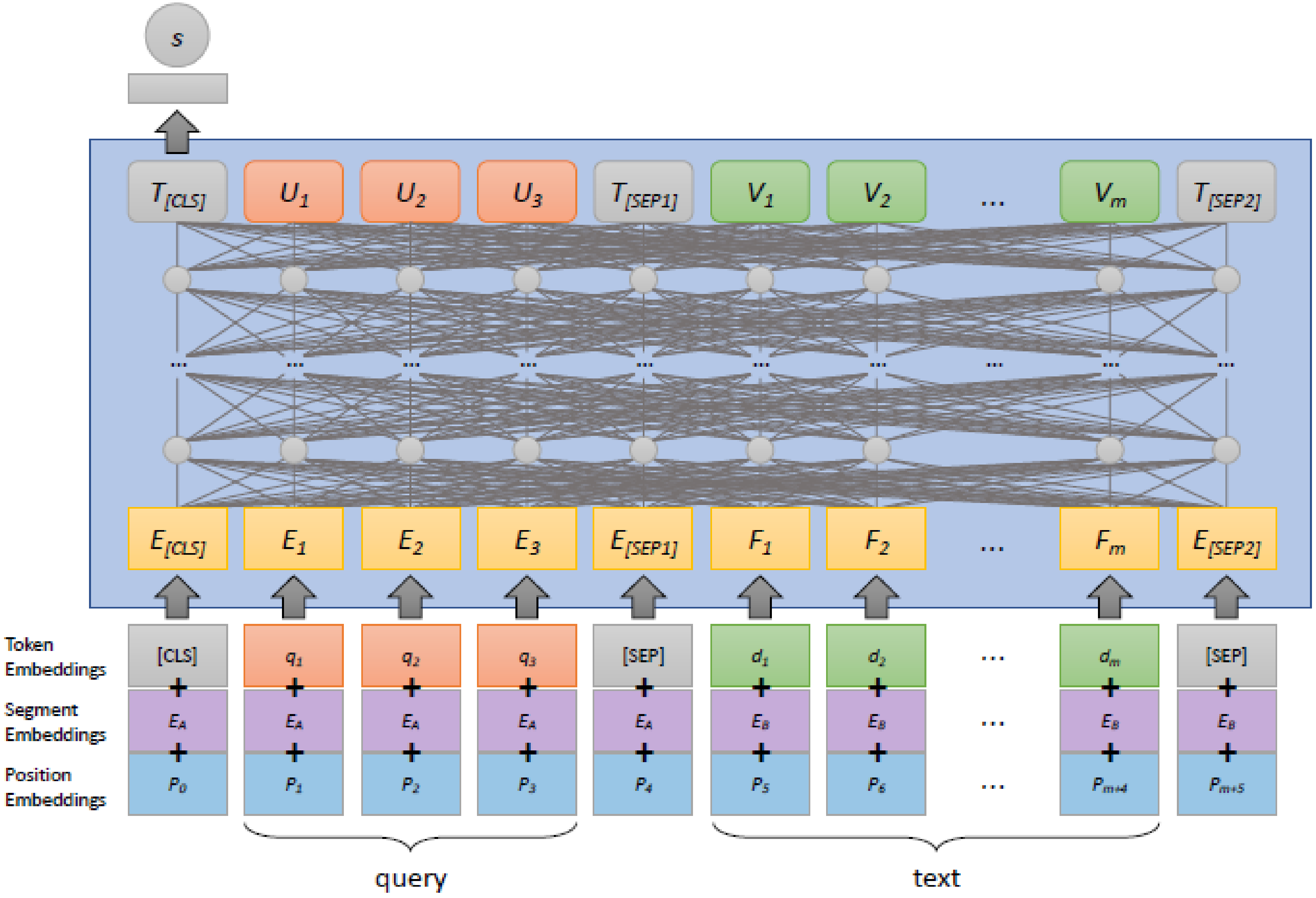
We need averages over all queries in the test set.

Retrieve then re-rank using BERT



- Candidate texts are retrieved from the document collection, typically with exact-match bag-of-words queries against inverted indexes.
- These candidates are then re-ranked with a transformer model such as BERT.

Learning to Rank: monoBERT



Learning to Re-Rank with BERT

- monoBERT adapts BERT for relevance classification by taking as input the query and a candidate text (surrounding by appropriate special tokens).
- The input vector representations comprise the element-wise summation of token embeddings, segment embeddings, and position embeddings.
- The output of the BERT model is a contextual embedding for each input token.
- The final representation of [CLS] token is fed to a fully-connected layer that produces the relevance score s of that text to the query.
- P(Relevant=1|di,q)

Performance improvements with BERT

Method		TREC 2019 DL Passage		
		nDCG@10	MAP	Recall@1k
(3a)	BM25 (Anserini, $k = 1000$)	0.5058	0.3773	0.7389
(3b)	+ monoBERT$_{\text{Large}}$	0.7383	0.5058	0.7389
(4a)	BM25 + RM3 (Anserini, $k = 1000$)	0.5180	0.4270	0.7882
(4b)	+ monoBERT$_{\text{Large}}$	0.7421	0.5291	0.7882

- The effectiveness of monoBERT on the TREC 2019 Deep Learning Track passage retrieval test collection

Extensions

- BERT is restricted to short texts (512 tokens).
- Sentence models.
- Extension to work with longer documents.

Examples of results for longer documents.

Method		Robust04		Core17		Core18	
		MAP	nDCG@20	MAP	nDCG@20	MAP	nDCG@20
(1)	BM25 + RM3	0.2903	0.4407	0.2823	0.4467	0.3135	0.4604
(2a)	1S: BERT(MB)	0.3408†	0.4900†	0.3091†	0.4628	0.3393†	0.4848†
(2b)	2S: BERT(MB)	0.3435†	0.4964†	0.3137†	0.4781	0.3421†	0.4857†
(2c)	3S: BERT(MB)	0.3434†	0.4998†	0.3154†	0.4852†	0.3419†	0.4878†
(3a)	1S: BERT(MS MARCO)	0.3028†	0.4512	0.2817†	0.4468	0.3121	0.4594
(3b)	2S: BERT(MS MARCO)	0.3028†	0.4512	0.2817†	0.4468	0.3121	0.4594
(3c)	3S: BERT(MS MARCO)	0.3028†	0.4512	0.2817†	0.4468	0.3121	0.4594
(4a)	1S: BERT(MS MARCO $\rightarrow$ MB)	0.3676†	0.5239†	0.3292†	0.5061†	0.3486†	**0.4953**†
(4b)	2S: BERT(MS MARCO $\rightarrow$ MB)	**0.3697**†	0.5324†	**0.3323**†	**0.5092**†	0.3496†	0.4899†
(4c)	3S: BERT(MS MARCO $\rightarrow$ MB)	0.3691†	**0.5325**†	0.3314†	0.5070†	**0.3522**†	0.4899†