# Ad-hoc retrieval with BERT

Deeper Text Understanding for IR with Contextual Neural Language Modeling - Dai et al (SIGIR'19)
CEDR: Contextualized Embeddings for Document Ranking - MacAvaney et al (SIGIR'19)
Multi-Stage Document Ranking with BERT - Nogueira et al

**Presenter: Udhav Sethi (u2sethi@uwaterloo.ca)**

# Outline

# Outline

1. **Ad-hoc retrieval and BERT - Introduction**
2. Deeper Text Understanding for IR with Contextual Neural Language Modeling
   a. Motivation
   b. Proposed Methods
3. CEDR: Contextualized Embeddings for Document Ranking
   a. Motivation
   b. Proposed Methods
4. Multi-Stage Document Ranking with BERT
   a. Motivation
   b. Proposed Methods
   c. Experiments
   d. Results & Analysis
5. References

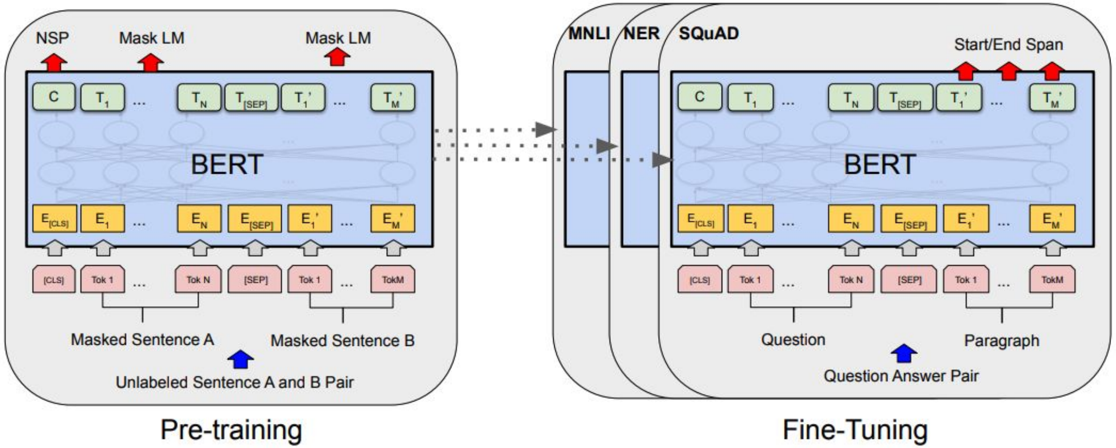
# Ad-hoc Document Retrieval

Standard retrieval task in which the user specifies his information need through a query which initiates a corpus search for documents which are likely to be relevant to the user.

- **Query**: textual description of information need.
- **Corpus**: a collection of textual documents.
- **Relevance**: satisfaction of the user's information need.
- "**Ad-hoc**" because the documents in the collection remain relatively static while new queries are submitted to the system continually.

# BERT

- BERT (Bidirectional Encoder Representations from Transformer) is a contextual neural language model designed to pretrain deep bidirectional representations from unlabeled text.
- The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks.



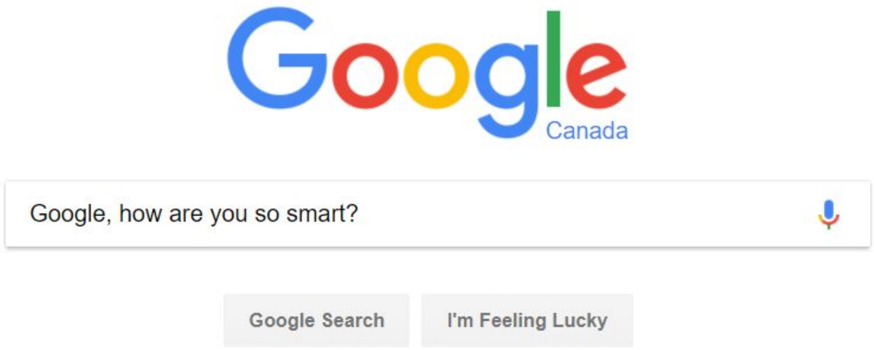
# Outline

1. Ad-hoc retrieval and BERT - Introduction
2. **Deeper Text Understanding for IR with Contextual Neural Language Modeling**
    a. **Motivation**
    b. Proposed Methods
3. CEDR: Contextualized Embeddings for Document Ranking
    a. Motivation
    b. Proposed Methods
4. Multi-Stage Document Ranking with BERT
    a. Motivation
    b. Proposed Methods
    c. Experiments
    d. Results & Analysis
5. References

# Motivation

## Semantic Search

- People have been trained to use keyword queries because bag-of-words retrieval models cannot effectively extract key information from natural language.
- Queries written in natural language actually enable better search results when the system can model language structures.



# Outline

1. Ad-hoc retrieval and BERT - Introduction
2. **Deeper Text Understanding for IR with Contextual Neural Language Modeling**
    a. Motivation
    b. **Proposed Methods**
3. CEDR: Contextualized Embeddings for Document Ranking
    a. Motivation
    b. Proposed Methods
4. Multi-Stage Document Ranking with BERT
    a. Motivation
    b. Proposed Methods
    c. Experiments
    d. Results & Analysis
5. References

# Model Architecture

- **Input Tokens -** concatenation of the query tokens and the document tokens, with token '[SEP]' separating the two segments, [CLS] at the beginning of the first segment..
- **Segment Embeddings -** 'Q' (for query tokens) and 'D' (for document tokens), to further separate the query from the document.
- **Position Embeddings -** To capture word order.
- **Output** - Embedding of the first token is used as a representation for the entire query-document pair. It is fed into a multi-layer perceptron (MLP) to predict the possibility of relevance (binary classification).

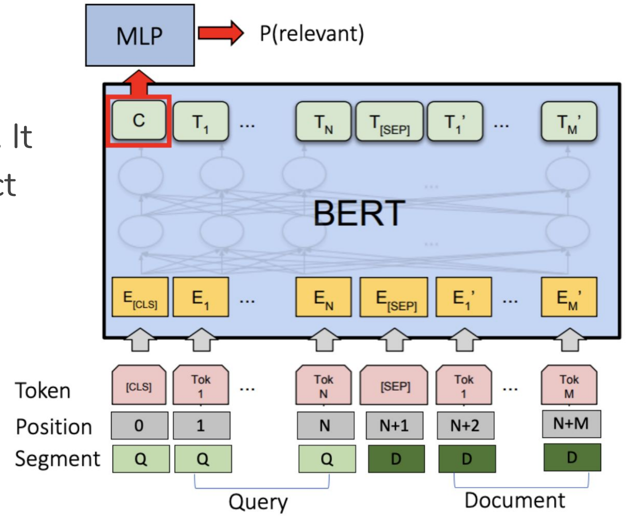



**Figure 1: BERT sentence pair classification architecture [3].**

# Sources of Effectiveness

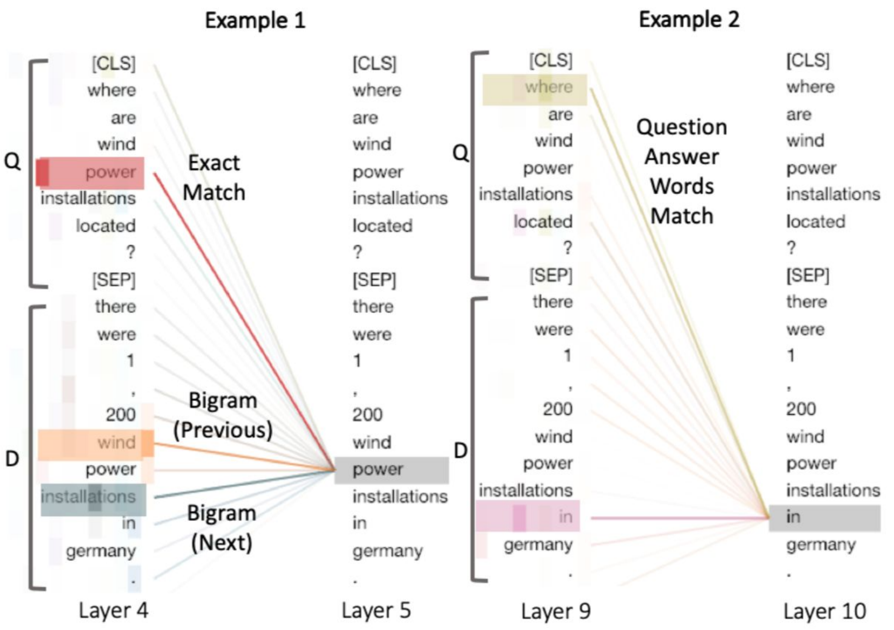



**Figure 2: Visualization of BERT. Colors represent different attention heads; deeper color indicates higher attention.**

**Table 1: Example of Robust04 search topic (Topic 697).**

| | |
|---|---|
| Title | air traffic controller |
| Description | What are working conditions and pay for U.S. air traffic controllers? |
| Narrative | Relevant documents tell something about working conditions or pay for American controllers. Documents about foreign controllers or individuals are not relevant. |

# Outline

1. Ad-hoc retrieval and BERT - Introduction
2. Deeper Text Understanding for IR with Contextual Neural Language Modeling
    a. Motivation
    b. Proposed Methods
3. **CEDR: Contextualized Embeddings for Document Ranking**
    a. **Motivation**
    b. Proposed Methods
4. Multi-Stage Document Ranking with BERT
    a. Motivation
    b. Proposed Methods
    c. Experiments
    d. Results & Analysis
5. References

# Motivation

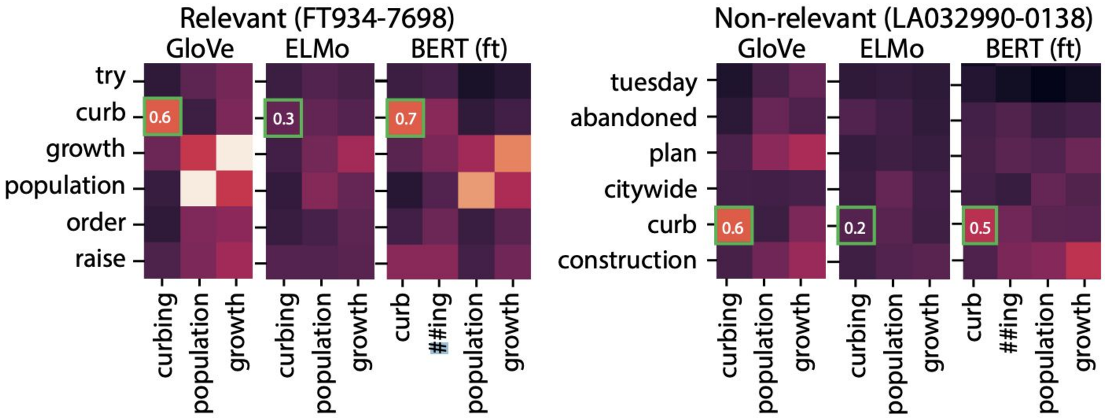



**Figure 1: Example similarity matrix excerpts from GloVe, ELMo, and BERT for relevant and non-relevant document for Robust query 435. Lighter values have higher similarity.**

# Outline

1. Ad-hoc retrieval and BERT - Introduction
2. Deeper Text Understanding for IR with Contextual Neural Language Modeling
    a. Motivation
    b. Proposed Methods
3. **CEDR: Contextualized Embeddings for Document Ranking**
    a. Motivation
    b. **Proposed Methods**
4. Multi-Stage Document Ranking with BERT
    a. Motivation
    b. Proposed Methods
    c. Experiments
    d. Results & Analysis
5. References

# Traditional Similarity Tensors

- **Q :** query consisting of query terms **$\{q1, q2, ..., q_{|Q|}\}$**
- **D :** document consisting of terms **$\{d1, d2, ..., d_{|D|}\}$**
- **ranker(Q,D) $\in$ R :** Real-valued relevance estimate for the document to the query.
- Neural relevance ranking architectures generally use a similarity matrix as input.

  **Similarity matrix: $S \in R^{|Q| \times |D|}$**, where each cell represents a similarity score between the query terms and document terms: **$S_{i,j} = sim(q_i, d_j)$.**

# New Contextualized Similarity Tensors

- Contextualized language models typically consist of multiple stacked layers of representations (e.g., recurrent or transformer outputs)
- New similarity representation (conditioned on the query and document context):

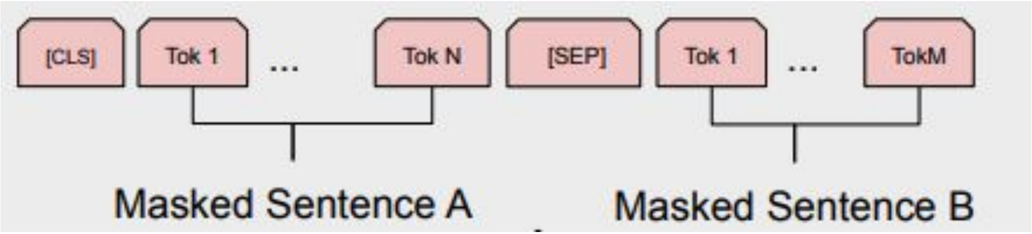
$$\mathbf{S}_{\mathbf{Q},\mathbf{D}}[l, q, d] = cos(context_{\mathbf{Q},\mathbf{D}}(q, l), context_{\mathbf{Q},\mathbf{D}}(d, l))$$

For each query term $q \in Q$, document term $d \in D$, and layer $l \in [1..L]$, where $context_{Q,D}(t,l) \in R^D$ is the contextualized representation for token t in layer l

- The representations from the stacked layers of contextualized language models like BERT can benefit general neural ranking models like PACRR, KNRM, DRMM.

# Joint BERT approach

- BERT utilizes the [CLS] token for making judgments about the text pairs. Its representation can be fine-tuned for other tasks.
- The [CLS] token representation is incorporated into existing neural ranking models as the **Joint BERT approach.**
- This allows neural rankers to benefit from deep semantic information from BERT in addition to individual contextualized token matches.



# Outline

# Motivation

**Representational learning**:  Learn some non-linear transformation of queries and documents such that documents relevant to a query have high similarities in terms of a simple metric such as cosine similarity.

- Search-related tasks need to consider a large corpus, and thus it is impractical to apply inference over all documents for a given query.
- It is unclear whether representational learning is sufficient to boil the complex notion of relevance down to simple similarity computations.
- The complete end-to-end retrieval architecture will need to involve multiple stages.

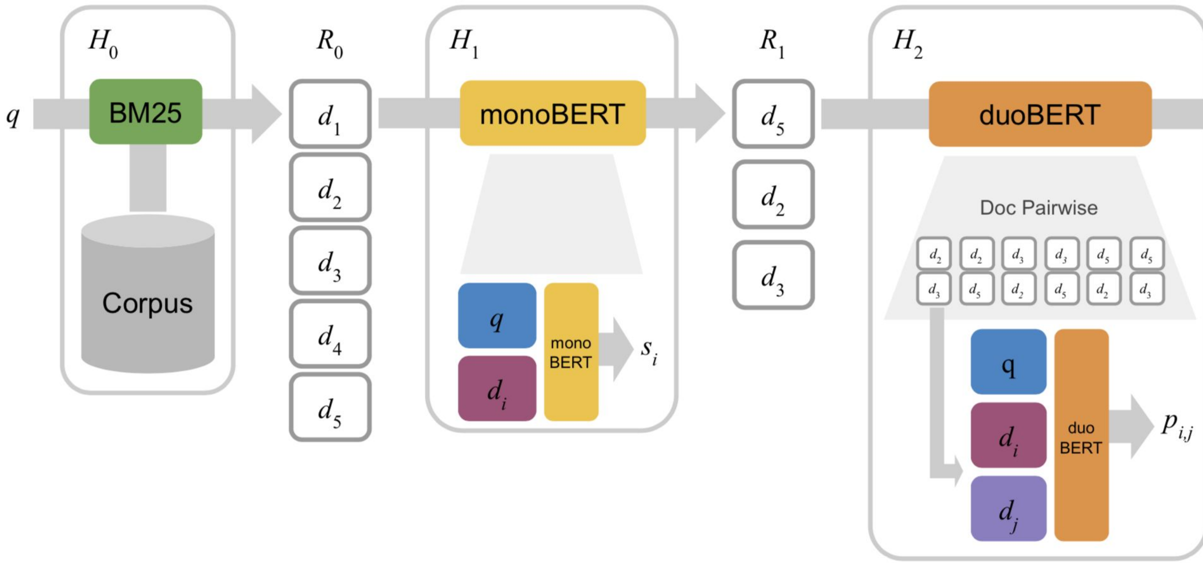
# Outline

1. Ad-hoc retrieval and BERT - Introduction
2. Deeper Text Understanding for IR with Contextual Neural Language Modeling
   a. Motivation
   b. Proposed Methods
3. CEDR: Contextualized Embeddings for Document Ranking
   a. Motivation
   b. Proposed Methods
4. **Multi-Stage Document Ranking with BERT**
   a. Motivation
   b. **Proposed Methods**
   c. Experiments
   d. Results & Analysis
5. References

# Multi Stage Ranking

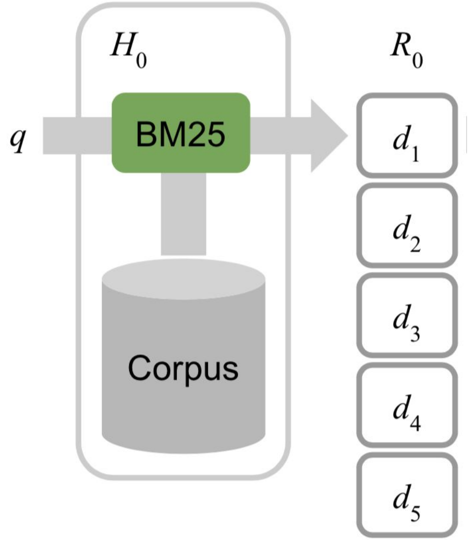
- A multi-stage ranking architecture comprises a number of stages, denoted $H_0$ to $H_N$.
- $H_0$ retrieves $k_0$ candidates from an inverted index
- $H_n$ receives a ranked list $R_{n-1}$ comprising $k_{n-1}$ candidates from the previous stage.
- $H_n$ provides a ranked list $R_n$ comprising $k_n$ candidates to the subsequent stage ($k_n \leq k_{n-1}$.
- The ranked list generated by the final stage is designated for consumption by the searcher.

# Model Architecture



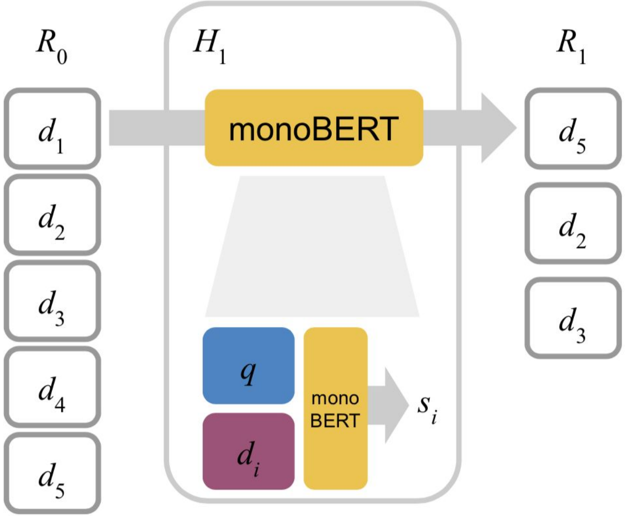
# $H_0$: “Bag of Words” BM25

- **Input:** user query q
- **Output:** top-$k_0$ candidates $R_0$
- Query is treated as a “bag of words” based on the BM25 scoring function.
- BM25 looks for exact term matches, but later BERT stages have the ability to identify relevant candidates that do not have many matching terms.
- Critical to optimize for recall to provide subsequent stages a diverse set of documents to work with; precision is less of a concern because non-relevant documents can be discarded by later stages.



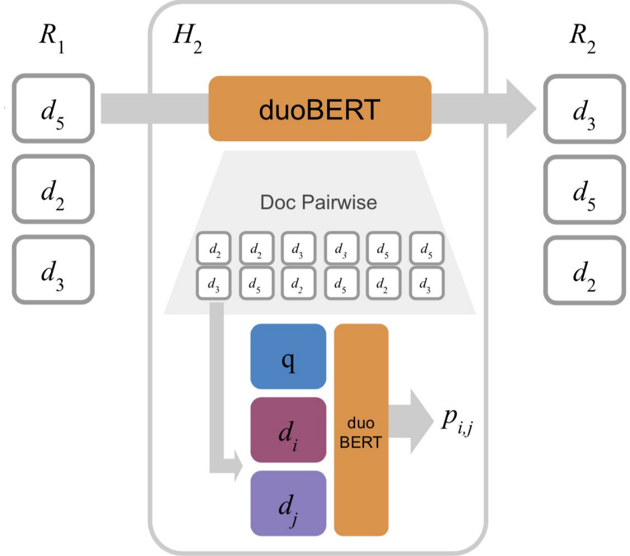
# $H_1$ : monoBERT

- **Input:** Query q as sentence A and text of candidate $d_i$ as sentence B
- **Output: $R_1$**, i.e., top-$k_1$ candidates based on $s_i$ scores
- **monoBERT:** pointwise re-ranker, i.e., a BERT model used as a binary relevance classifier.
- Truncate so concatenation of query, candidate, and separator tokens have a maximum length of 512 tokens
- Use [CLS] vector as input to a single layer neural network to obtain a probability $s_i$ of the candidate $d_i$ being relevant to q



# $H_2$: duoBERT

- **Input:** query as sentence A, candidate $d_i$ as sentence B, and candidate $d_j$ as sentence C
- **Output:** $R_2$, obtained by re-ranking the candidates in $R_1$ according to their scores $s_i$
- **duoBERT:** pairwise re-ranker, i.e., estimates the probability $p_{i,j}$ of the candidate $d_i$ being more relevant than $d_j$
- Truncate so concatenation of query, candidate, and separator tokens have a maximum length of 512 tokens.
- Use [CLS] vector as input to a single layer neural network to obtain the probability $p_{i,j}$
- Aggregate the pairwise scores $p_{i,j}$ so that each document receives a single score $s_i$



# $H_2$: duoBERT - Aggregation methods

$$\textsc{Sum} : s_i = \sum_{j \in J_i} p_{i,j},$$

$$\textsc{Binary} : s_i = \sum_{j \in J_i} \mathbb{1}_{p_{i,j} > 0.5},$$

$$\textsc{Min} : s_i = \min_{j \in J_i} p_{i,j},$$

$$\textsc{Max} : s_i = \max_{j \in J_i} p_{i,j},$$

$$\textsc{Sample} : s_i = \sum_{j \in J_i(m)} p_{i,j},$$

where $J_i = \{0 \leq j < |R_1|, j \neq i\}$ and $m$ is the number of samples drawn without replacement from the set $J_i$.

# Outline

# Datasets - I

**MS MARCO (Microsoft MAchine Reading COmprehension)**: created from half a million anonymized questions sampled from Bing's search query logs.

- 8.8M passages extracted from 3.6M web documents, 55 words per passage.
- **Training set:** 500k pairs of query and relevant document, 400M pairs of query and non-relevant documents.
- **Development set:** 6,980 queries, with, on average, one relevant document per query.
- **Evaluation set:** 6,837 queries without relevance judgments.
- Official metric for dataset: MRR@10

# Datasets - II

**TREC CAR (Complex Answer Retrieval):** consists of cleaned paragraphs from English Wikipedia.

- 29M documents, with an average of 60 words per document.
- **Training set:** 3M queries
- **Validation set:** 700k queries
- **Evaluation set:** 2,254 queries
- Official metric for dataset: Mean Average Precision (MAP)

# Outline

# MS MARCO Results

| **Method** | Dev | Eval |
|---|---|---|
| BM25 (Microsoft Baseline) | 16.7 | 16.5 |
| IRNet | 27.8 | 28.1 |
| monoBERT (Jan 2019) | 36.5 | 35.9 |
| Anserini (BM25) | 18.7 | 19.0 |
| + monoBERT | 37.2 | 36.5 |
| + monoBERT + duoBERT$_{\text{MAX}}$ | 32.6 | - |
| + monoBERT + duoBERT$_{\text{MIN}}$ | 37.9 | - |
| + monoBERT + duoBERT$_{\text{SUM}}$ | 38.2 | 37.0 |
| + monoBERT + duoBERT$_{\text{BINARY}}$ | 38.3 | - |
| + monoBERT + duoBERT$_{\text{SUM}}$ + TCP | 39.0 | 37.9 |
| Leaderboard best | 39.7 | 38.3 |

Table 1: MS MARCO Results.

# TREC CAR Results

| **Method** | MAP |
|---|---|
| BM25 (Kashyapi et al., 2018) | 13.0 |
| Co-PACRR (MacAvaney et al., 2017) | 14.8 |
| BM25 (Anserini) | 15.3 |
| + monoBERT | 34.8 |
| + monoBERT + duoBERT$_{\text{MAX}}$ | 32.6 |
| + monoBERT + duoBERT$_{\text{SUM}}$ | 36.9 |
| + monoBERT + duoBERT$_{\text{BINARY}}$ | 36.9 |

Table 2: Main Result on TREC 2017 CAR.

# Tradeoffs with monoBERT

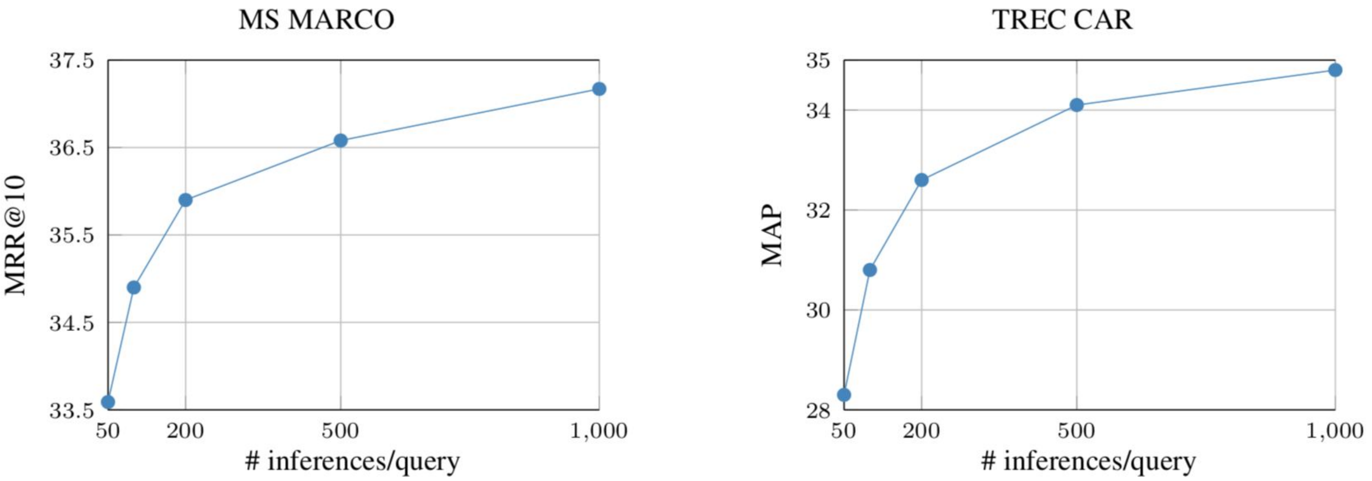



Figure 2: Number of inferences per query vs. effectiveness on the MS MARCO and the TREC CAR datasets when varying the number of candidates $k_0$ fed to monoBERT.

# Tradeoffs with duoBERT

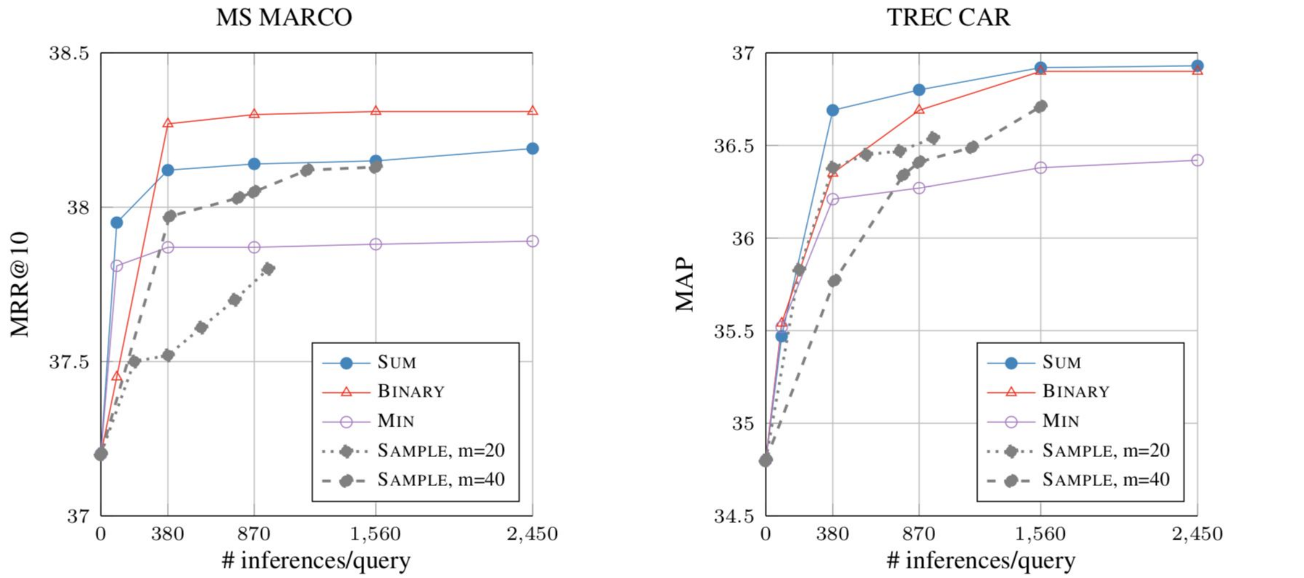



Figure 3: Number of inferences per query vs. the effectiveness of duoBERT when varying the number of candidates $k_1$. Each curve has six points that correspond to $k_1 = \{0, 10, 20, 30, 40, 50\}$, where $k_1 = 0$ corresponds to monoBERT. The values in the $x$-axis are computed as $k_1 \times (k_1 - 1)$ for SUM, BINARY, and MIN, and $k_1 \times (m - 1)$ for SAMPLE. To avoid clutter, plots for SAMPLE at $m = \{10, 30\}$ are omitted.

# Multi-Stage Tradeoffs

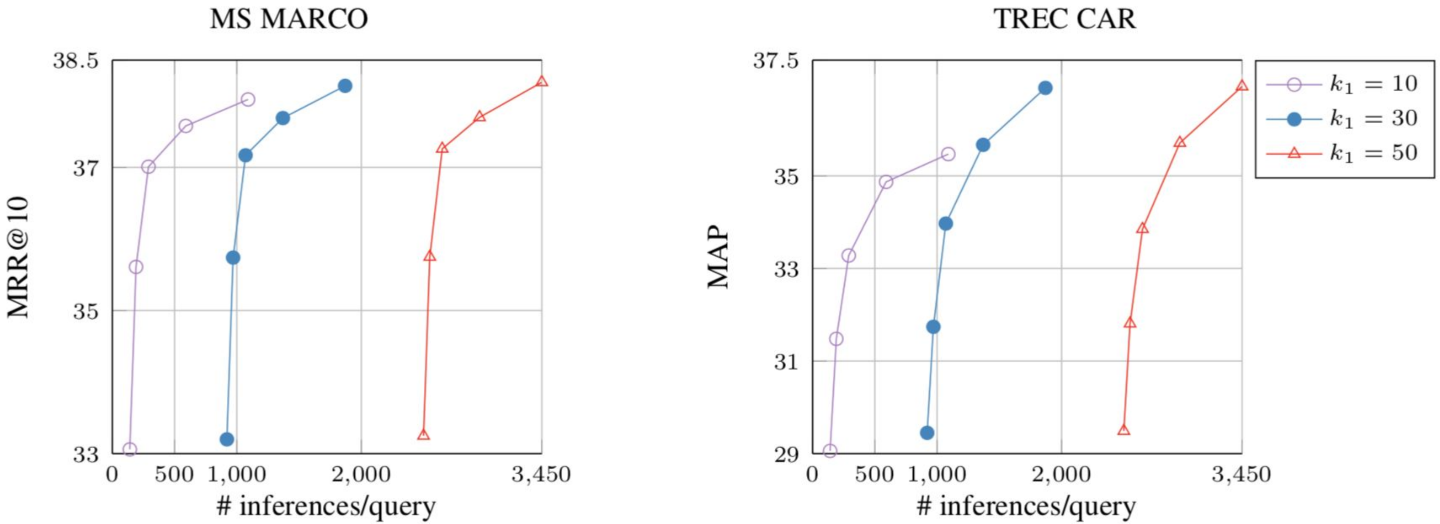



Figure 4: Number of inferences per query vs. the effectiveness of duoBERT$_{\text{SUM}}$ when varying the number of candidates $k_0$ and $k_1$. Each curve has five points that correspond to $k_0 = \{50, 100, 200, 500, 1000\}$. The number of inferences per query is calculated as $k_0 + k_1(k_1 - 1)$.

# Qualitative Analyses

| Query | Sample Passage | Label | Rank | |
|---|---|---|---|---|
| | | | Baseline | Comparison |
| **who wrote song killing the blues** | Killing The Blues by Robert Plant and Alison Krauss. This was written by Chris Isaak's bass guitarist Roly Salley, and was originally the title track of Salley's 2005 solo album. This song was used in an advertising campaign for the chain store JC Penney, which features sentimental images of heartland Americana, such as family reunions and Fourth of July celebrations. | R | BM25: 621 | monoBERT: 1 |
| | Who wrote the blues song Crossroads Cross Road Blues is one of Delta Blues singer Robert Johnson's most famous songs. Who wrote the song 'Blue Shades.. Frank Ticheli wrote the song 'Blue Shades'. It is a concert piece with allusions... | N | BM25: 1 | monoBERT: 9 |
| **what causes low liver enzymes** | Reduced production of liver enzymes may indicate dysfunction of the liver. This article explains the causes and symptoms of low liver enzymes. Scroll down to know how the production of the enzymes can be accelerated. | R | monoBERT: 47 | duoBERT: 1 |
| | Other causes of elevated liver enzymes may include: Alcoholic hepatitis (severe liver inflammation caused by excessive alcohol consumption) Autoimmune hepatitis (liver inflammation caused by an autoimmune disorder) Celiac disease (small intestine damage caused by gluten) Cytomegalovirus (CMV) infection. | N | monoBERT: 1 | duoBERT: 7 |

Table 3: Comparison of BM25 vs. monoBERT, and monoBERT vs. duoBERT, showing result ranks of answers. (**N**: not relevant, **R**: relevant)

# References


1. Nogueira, Rodrigo, et al. "Multi-stage document ranking with BERT." *arXiv preprint arXiv:1910.14424* (2019).
2. Dai, Zhuyun, and Jamie Callan. "Deeper text understanding for IR with contextual neural language modeling." *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019.
3. MacAvaney, Sean, et al. "CEDR: Contextualized embeddings for document ranking." *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019.
4. Nogueira, Rodrigo, and Kyunghyun Cho. "Passage Re-ranking with BERT." *arXiv preprint arXiv:1901.04085* (2019).
5. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

# THANK YOU!
# Q&A