University of Waterloo E-Thesis Template for LaTeX

by

Zeynep Akkalyoncu Yilmaz

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2019

Abstract

Standard bag-of-words term-matching techniques in document retrieval fail to exploit rich semantic information embedded in the document texts. One promising recent trend in facilitating context-aware semantic matching has been the development of massively pre-trained language models, culminating in BERT as its most popular example today. In this work, we propose adapting BERT as a neural reranker for document retrieval with large improvements on news articles. Two fundamental issues arise in applying BERT to "ad hoc" document retrieval on newswire collections: relevance judgements in existing test collections are provided only at the document level, and documents often exceed the length that BERT was designed to handle. To overcome these challenges, we compute and aggregate sentence-level relevance scores to rank documents. We solve the problem of lack of appropriate relevance judgements by leveraging sentence-level and passage-level relevance judgements available in collections from other domains to capture cross-domain notions of relevance. We demonstrate that models of relevance can be transferred across domains. By leveraging semantic cues learned across various domains, we propose a model that achieves state-of-the-art results across three standard TREC newswire collections. We explore the effects of cross-domain relevance transfer, and trade-offs between using document and sentence scores for document ranking. We also present an end-to-end document retrieval system that incorporates the open-source Anserini information retrieval toolkit, discussing the related technical challenges and design decisions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Document retrieval refers to the task of generating a ranking of documents from a large corpus D in response to a query Q. In a typical document retrieval pipeline, an inverted index is constructed in advance from the collection, which often comprises unstructured text documents, for fast access during retrieval. When the user issues a query, the query representation is matched against the index, computing a similarity score for each document. The top most relevant documents based on their closeness to the query are returned to the user in order of relevance. This procedure may be followed by a subsequent re-ranking stage where the candidate documents outputted by the previous step are further re-ranked in a way that maximizes some retrieval metric such as average precision (AP).

Document retrieval systems traditionally rely on term-matching techniques, such as BM25, to judge the relevance of documents in a corpus. More specifically, the more common terms a document shares with the query, the more relevant it is considered. As a result, these systems may fail to detect documents that do not contain exact query terms, but are nonetheless relevant. For example, consider a document that expresses relevant information in a way that cannot be resolved without external semantic analysis. Figure 1 displays

Query: international art crime
Text: The thieves demand a ransom of $2.2 million for the works and return one of them.

Figure 1.1: An example of a query-text pair from the TREC Robust04 collection where a relevant piece of text does not contain direct query matches.

one such query-text pair where words semantically close to the query need to be identified to establish relevance. This "vocabulary mismatch" problem represents a long-standing challenge in information retrieval. To put its significance into context, Zhao et al. [38] show in their paper on term necessity prediction that, statistically, the average query terms do not appear in as many as 30% of relevant documents in TREC 3 to 8 "ad hoc" retrieval datasets.

Clearly, the classic exact matching approach to document retrieval neglects to exploit rich semantic information embedded in the document texts. To overcome this shortcoming, a number of models such as Latent Semantic Analysis [4], which map both queries and documents into high-dimensional vectors, and measure closeness between the two based on vector similarity, has been proposed. This innovation has enabled semantic matching to improve document retrieval by extracting useful semantic signals. With the advent of neural networks, it has become possible to learn better distributed representations of words that capture more fine-grained semantic and syntatic information [14], [19]. More recently, massively unsupervised language models that learn context-specific semantic information from copious amounts of data have changed the tide in NLP research (e.g: ELMo [20], GPT-2 [22]). These models can be applied to various downstream tasks with minimal task-specific fine-tuning, highlighting the power of transfer learning from large pre-trained models. Arguably the most popular example of these deep language representation models is the Bidirectional Encoder Representations from Transformers (BERT) [5]. BERT has achieved state-of-the-art results across a broad range of NLP tasks from question answering to machine translation.

While BERT has enjoyed widespread adoption across the NLP community, its application in information retrieval research has been limited in comparison. Guo et al. [7] suggest that the lackluster success of deep neural networks in information retrieval may be owing to the fact that they often do not properly address crucial characteristics of the "ad hoc" document retrieval task. Specifically, the relevance matching problem in information retrieval and semantic matching problem in natural language processing are fundamentally different in that the former depends heavily on exact matching signals, query term importance and diverse matching requirements. In other words, it is crucial to strike a good balance between exact and semantic matching in document retrieval. For this reason, we employ both document scores based on term-matching and semantic relevance scores to determine the relevance of documents.

In this thesis, we extend the work of Yang et al. [35] by presenting a novel way to apply BERT to "ad hoc" document retrieval on long documents – particularly, newswire articles – with significant improvements. Following Nogueira et al. [16], we adapt BERT for binary relevance classification over text to capture notions of relevance. We then deploy the BERT-based re-ranker as part of a multi-stage architecture where an initial list of candidate documents is retrieved with a standard bag-of-words term matching technique. The BERT model is used to compute a relevance score for each constituent sentence, and the candidate documents are re-ranked by combining sentence scores with the original document score.

We emphasize that applying BERT to document retrieval on newswire documents is not trivial due to two main challenges: Firist of all, BERT has a maximum input length of 512 tokens, which is insufficient to accommodate the overall length of most news articles. To put this into perspective, a typical TREC Robust04 document has a median length of 679 tokens, and in fact, 66% of all documents are longer than 512 tokens. Secondly, most collections provide relevance judgements only at the document level. Therefore, we only know what documents are relevant for a given query, but not the specific spans within the document. To further aggravate this issue, a document is considered relevant as long as some part of it is relevant, and most of the document often has nothing to do with the query.

We address the abovementioned challenges by proposing two effective innovations: First, instead of relying solely on document-level relevance judgements, we aggregate sentence-level evidence to rank documents. As mentioned before, since standard newswire collections lack sentence level judgements to facilitate this approach, we instead explore leveraging sentence-level or passage-level judgements already available in collections in other domains, such as tweets and reading comprehension. To this end, we fine-tune BERT models on these out-of-domain collections to learn models of relevance. Surprisingly, we demonstrate that models of relevance can indeed be successfully transferred across domains. It is important to note that the representational power of neural networks come at the cost of challenges in interpretability. For this reason, we dedicate a portion of this thesis to error analysis experiments in an attempt to qualify and better understand the cross-domain transfer effects. We also elaborate on our engineering efforts to ensure reproducibility and replicability, and the technical challenges involved in bridging the worlds of natural language processing and information retrieval from a software engineering perspective.

1.1 Contributions

The main contributions of this thesis can be summarized as follows:

- We present two innovations to successfully apply BERT to *ad hoc* document retrieval with large improvements: integrating sentence-level evidence to address the fact that BERT cannot process long spans posed by newswire documents, and exploiting cross-domain models of relevance for collections without sentence- or passage-level annotations. With the proposed model, we establish state-of-the-art effectiveness on three standard TREC newswire collections at the time of writing. Our results on Robust04 exceed the previous highest known score of 0.3686 [2] with a non-neural method based on ensembles, which has stood unchallenged for ten years.
- We explore through various error analysis experiments the effects of cross-domain relevance transfer with BERT as well as the contributions of BM25 and sentence scores to the final document ranking. Elaborate more?
- We release an end-to-end pipeline, Birch[1], that applies BERT to document retrieval over large document collections via integration with the open-source Anserini information retrieval toolkit. An accompanying Docker image is also included to ensure that anyone can easily deploy and test our system. We elaborate on the technical challenges in the integration of NLP and IR capabilities, and the rationale behind design decisions.

1.2 Thesis Organization

Add link to actual chapters The remainder of this thesis is organized in the following order: Chapter 2 reviews related work in neural document retrieval and transfer learning, particularly applications of BERT to document retrieval. Chapter 3 motivates the approach with some background information on the task, and introduces the datasets used for both training and evaluation as well as metrics. Chapter 4 proposes an end-to-end pipeline for document retrieval with BERT by elaborating on the design decisions and challenges. Chapter 5 describes the experimental setup, and presents the results on three newswire collections – Robust04, Core17 and Core18. Chapter 6 concludes the thesis by summarizing the contributions and discussing future work.

[1] https://github.com/castorini/birch

Chapter 2

Background and Related Work

2.1 Unsupervised Language Pretraining

2.1.1 Word Representations

Word representations are an integral building block of natural language processing applications with a significant influence on their performance. The underlying problem concerns how to convert arbitrary alphanumeric representations into a format that can be represented on and processed by computers. The simplest approach to attack this problem may be to pre-process words, e.g: with lemmatization, and to map them to unique word IDs, thereby retrieving words with a dictionary lookup. This approach especially does not fare well with deep learning models because it introduces a misleading notion of ordering among the words as indicated by their IDs. This shortcoming can be overcome with a one-hot encoding scheme where a word is represented as a vocabulary-sized vector with only the index corresponding to the word ID is set to 1 and the rest to 0. This massive sparse vector representation comes at the cost of a large space requirement.

More importantly, neither approach captures an intrinsic notion of similarity between different words given the orthogonal representation space. Intuitively, words are treated as independent entities, as in exact matching in information retrieval, therefore failing to leverage the context. For this reason, learning word representations automatically has been an active research effort in NLP for decades, spanning both non-neural cite and neural cite methods. This paradigm frames words as continuous feature vectors that embody hidden semantic or syntactic dependencies. In order to incorporate context, word2vec

[14] represents each word based on neighbouring words in a given window with a skip-gram model. GloVe [19] instead exploits the co-occurrence of words rather than individual context windows. These feature vectors can be learned from scratch for a specific corpus or pretrained over large corpora and reused, leading to significant improvements over the former [28]. For example, popular pretrained word2vec vectors are trained on Google News, and GloVe on Common Crawl, Wikipeda and Twitter. Such word embeddings enable many of the neural information retrieval methods discussed in Section X.

BERT tokenization!

Used in models discussed in Section...

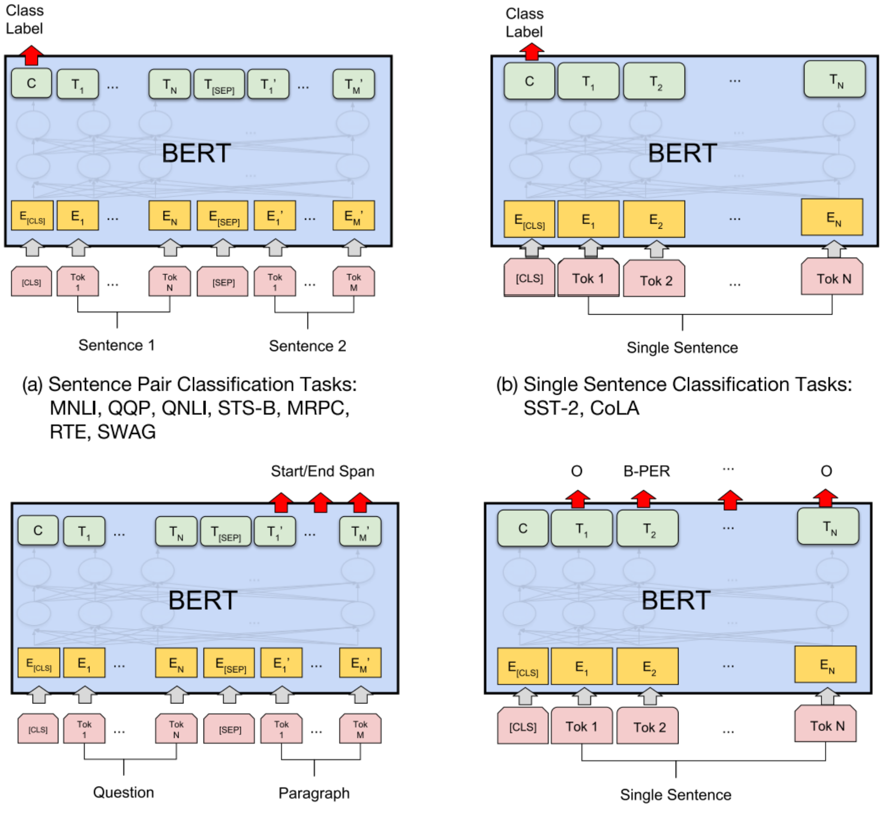
2.1.2 Pretrained Language Models

With the increasing availability of large corpora, pretrained deep language models have been rapidly gaining traction among NLP researchers. Recent work in NLP has demonstrated that language model pretraining proves extremely effective for many natural language processing tasks ranging from machine translation to reading comprehension. In general, these models work by LSTM-based or transformer-based language models on large corpora, and then are applied to downstream NLP tasks in one of two ways.

Feature-based approach, such as ELMo [20], employs deep pretrained representations learned with language modeling as additional features in task-specific architectures. This approach has the advantage of being easily incorporated into existing models to significantly improve performance. Advantages? ELMo [20] extends the traditional word embeddings introduced in Section X to learn context-sensitive features with a deep language model. These deep contextualized word representations are learned by X the internal states of a deep bidirectional language model trained on a large corpus What corpus? Peters at al. [20] show that integrating the embeddings learned with ELMo into task-specific architectures significantly improves over the state of the art at the time in six NLP tasks, including question answering on SQuAD [23] and sentiment analysis on the Stanford Sentiment Treebank (SST-5) [27].

The more popular fine-tuning approach, such as the GPT-2 [22], is inspired by the recent trend in transfer learning. These models are first pretrained with respect to a language modeling objective, and then applied to downstream NLP tasks by “freezing” their last layer, and “fine-tuning” on external data for the specific task with minimal task-specific parameters. Not only does this approach introduce only a few task-specific parameters to optimize, but it also greatly boosts the performance of many NLP tasks given the rich semantic expressiveness introduced by the pretrained language model. Radford et al. [22] claim that language models inherently capture many NLP tasks without explicit supervision. They propose Generative Pretrained Transformers (GPT) to perform zero-shot task transfer on multiple sentence-level tasks from the GLUE benchmark [30] with impressive results.

One of the latest and most sophisticated pretrained deep language models is undoubtedly the Bidirectional Encoder Representations from Transformers (BERT), which has already enjoyed widespread popularity across the NLP community. [5] Unlike previous language models, such as OpenAI's GPT [22], BERT produces deep bidirectional representations by conditioning on both left and right context in all layers by employing a new pretraining objective called "masked language model" (MLM). Conceptually, MLM ran-



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Figure 2.1: BERT Models.

domly masks some of the tokens from the input with the goal of predicting the original token based only on its left and right context. This objective allows the representation to fuse both left and right context, which is crucial for token-level tasks such as question answering according to the authors.

Comparison image?

In additional to the novel language modeling approach, Devlin et al. [5] also propose a "next sentence prediction" task for applications that require an understanding of the relationship between two sentences, such as question answering or language language inference. Details? May even add example in the paper? Figure 2.1 visualizes the input and output for fine-tuning BERT for a "sentence pair classification" model.

The underlying model architecture of BERT is a multi-layer bidirectional transformer [29]. The larger BERT model has 24 layers each with 1024 hidden nodes, and 16 self-attention heads in total. As expected, optimizing this objective requires a very complex model: for example, the larger BERT model requires around 340 million parameters be optimized. In fact, training this model end-to-end takes four days to complete even on 16 high-end tensor processing units (TPUs) [5]. The input representation for BERT is formed by concatenating the token with segment and position embeddings. The words are represented with WordPiece embeddings [] with a vocabulary of 30,000 tokens. Positional embeddings are learned for a length up to 512, which is the maximum input size allowed. The input to BERT may contain a single sentence or a sentence pair separated by the meta-token $[SEP]$, i.e: separator. Each sequence is prepended with $[CLS]$, corresponding to the "class" meta-token, whose final hidden state can be used for classification tasks.

To fine-tune BERT for classification tasks, a single-layer neural network is added to the end of this network with the class label as the input, and label probabilities are computed with softmax. The parameters of this layer and BERT are fine-tuned jointly to maximize the log-probability of the correct label. BERT has been applied to a broad range of NLP tasks from ... with impressive results. Examples Most relevant to the task of document retrieval, these include BERTserini by Yang et al. [34] which integrated BERT with Anserini for question answering over Wikipedia by fine-tuning BERT on SQuAD, and Nogueira et al. [16] who adopted BERT for passage reranking over MS MARCO.

2.2 Document Retrieval

2.2.1 Unsupervised Document Retrieval

Exact matching

Okapi BM25

Binary and its extension to BM25... shortcomings semantic

Okapi BM25 (commonly dubbed BM25) is a bag-of-words ranking function that was developed to accommodate documents of variable lengths. BM25 ranks documents based on the occurrence of query terms in each document, paying more attention to the rarer terms in the query. The goal of this approach is to take into account term frequency and document frequency while estimating the relevance of a document for a given query without introducing too many additional parameters. (Sparck) To achieve this goal, BM25 implementations define two parameters for term frequency saturation and field-length normalization, respectively. Tuned for most common datasets?

RM3

RM3 is a pseudo-relevance feedback mechanism where the original query is expanded by adding terms found in the contents of relevant BM25 documents. Why is it effective, why does it improve performance, how common is it

Previous approaches and results on these datasets Cormack et al and other stuff

2.2.2 Neural Document Retrieval

Rephrase to highlight semantic matching

concerns Neural models developed to address the deep matching problem can be divided into two broad categories based on their underlying architecture: representation-based and interaction-based.

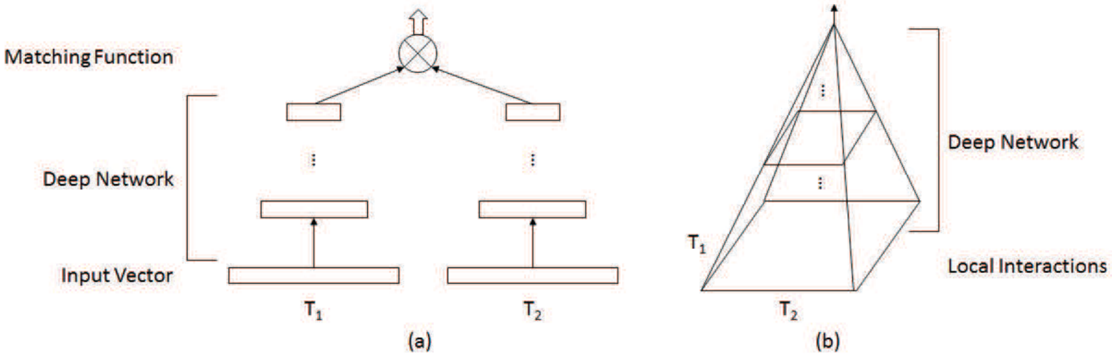


Figure 2.2: [8]

Representation-based Models

Representation-based approaches first construct the representation from the input vectors for each text, e.g: documents, with a deep neural network, and then perform matching between the representations. Earlier work on neural information retrieval focused on representation-based approaches, such as DSSM [9] and C-DSSM [26]. DSSM (short for Deep Structured Semantic Models) [9] extended previously dominant latent semantic models to deep semantic matching for web search by projecting query and documents into a common low-dimensional space. In order to accommodate a large vocabulary required by the task, the text sequences were mapped into character-level trigrams with a word hashing layer before computing a similarity matrix through dot product and softmax layers. While shown effective on a private? datasset comprised of log files of a commercial search engine, DSSM was criticized by Guo et al. [8] for requiring too much training data. Moreover, DSSM cannot match synonyms because it is based on the specific composition of words. C-DSSM [26] extended DSSM by introducing a convolutional layer to devise semantic vectors for search queries and Web document. By performing a max pooling operation to extract local contextual information at the n-gram level, a global vector representation is formed from the local features. They demonstrated that both local and global contextual features were necessary for semantic matching for Web search. While C-DSSM improves over DSSM by exploiting the context of each trigram, it still suffers from the same issues listed above.

Interaction-based Models

Interaction-based approaches instead capture local matching signals, and directly compute the similarity of the query and document representations. In contrast to more shallow representation-based approaches, in this approach deep neural network learns more complex hierarchical matching patterns. Some notable examples include DRMM [8], KNRM [31] and DUET [15]. DRMM (stands for Deep Relevance Matching Model) [8] maps the variable-length local interactions of each query term with the document? into a fixed-length matching histogram. A feed forward matching network is used to learn hierarchical matching patterns and compute a matching score is computed for each term. An overall matching score is obtained by aggregating the scores from each query term with a term gating network. Similar to other models, KNRM [31] calculates the word-word similarities between query and document embeddings. They propose a novel kernel-pooling technique to convert word-level interactions into ranking features. Details? Finally they combine the ranking features into a final ranking score through a learning-to-rank layer. Private dataset and stuff? Benefits? Unlike DRMM and KNRM, the goal of DUET [15] is to employ both local and distributed representations to leverage both exact matching and semantic matching signals. DUET is composed of two separate deep neural networks, one to match the query and the document using a one-hot representation, and another using learned distributed representations, which are trained jointly. The former estimates document relevance based on exact matches of the query terms in the document by computing an interaction matrix from one-hot encodings. The latter instead performs semantic matching by computing the element-wise product between the query and document embeddings. Their approach was shown to significantly outperform traditional baselines for web search with lots of clickthrough logs. Any other major models in the tutorial?

In fact, despite growing interest in neural models for document ranking, researchers have recently voiced concern as to whether or not they have truly contributed to progress [10], at least in the absence of large amounts of behavioral log data only available to search engine companies. This opinion piece echoes the general skepticism concerning the empirical rigor and contributions of machine learning applications in Lipton et al. [11] and Sculley at al. [25].

To rigorously test this claim, Yang et al. [33] recently conducted a thorough meta-analysis of over 100 papers that report results on the TREC Robust 2004 Track. Their findings are illustrated in Figure 2.3 where the solid black line represents the best submitted run at 0.333, and the dotted black line the median TREC run at 0.258. The other line is a RM3 baseline run with default parameters from the Anserini open-source information retrieval toolkit [32] at AP 0.3903. The untuned RM3 baseline is more effective than 60%

Condition	**AP**	**NDCG@20**
BM25 [7]	0.255	0.418
DRMM [7]	0.279	0.431
BM25+RM3	0.3033	0.4514
+ DSSM	0.3026	0.4491
+ CDSSM	0.2995	0.4468
+ DRMM	0.3152†	0.4718†
+ KNRM	0.3036	0.4441
+ DUET	0.3051	0.4502

Table 2.1: Experimental results applying neural models to rerank a strong baseline; † indicates statistical significance.

of all studied papers, and 20% of them report results below the TREC median. More surprisingly, only six of them report AP scores higher than the TREC best, with the highest being by Cormack et al. [2] in 2009 at AP 0.3686. Among the neural models, the highest encountered score is by Zamani et al. [36] in 2018 at AP 0.2971.

Yang et al. also implemented five recent neural retrieval models discussed above to evaluate their effectiveness on the "ad hoc" document retrieval task on the Robust04 dataset: DSSM, CDSSM, DRMM, KNRM and DUET. These models were selected because they were specifically designed for "ad hoc" document retrieval unlike some others designed to handle shorter texts? All the models were trained on the documents in the baseline RM3 runs... details CV etc Table 2.1 displays the AP and NDCG@20 values of each run on the Robust04 dataset. The first two rows are taken from the original DRMM paper [8] and show their reported baseline and results; the other models do not report results on Robust04. The next row refers to the untuned RM3 baseline from Anserini. The following results refer to results from the neural models that were used to rerank the strong baseline, BM25+RM3, to gauge how much they actually contribute. Of the five models, only one – DRMM – is found to significantly improve over the baseline.

Contextualized Language Models

While the neural ranking models introduced in the previous section successfully leverage contextual information to improve retrieval effectiveness, they are limited by the size and variability of the available training data. Ideally, these models would be trained on a large number of semantically varied yet relevant query-document pairs; however, it is impractical to automatically gather a sufficient number of such training samples. Massively pretrained unsupervised language models hold promises for obtaining better representations for the query and document, and therefore, achieving unprecedented effectiveness at semantic matching without the need for more relevance information. Language models are pretrained on massive amounts of external data in an unsupervised fashion. ELMo, GPT, BERT...

Document retrieval requires an understanding of the relationship between two text sequences – the query and the document. However, language modeling does not suffice to capture such a relationship. BERT facilitates such relevance classification by pre-training a binary next sentence prediction task based on its masking language model approach. This mechanism has been adopted for "passage" reranking by ... Notably, Nogueira et al. [16] proposed to re-rank MS MARCO passages based on a simple re-implementation of BERT, outperforming the previous state of the art by 27% in MRR@10 and replacing the top entry in the leaderboard of the MS MARCO passage retrieval task at the time of publication. Our neural model is inspired by the BERT re-implementation described in this paper. What about others? MS MARCO leaderboard?

To our knowledge, Yang et al. [35] were the first to successfully apply BERT to "ad hoc" document retrieval. They demonstrated that BERT can be fine-tuned to capture relevance matching by applying the above approach on the TREC Microblog Tracks where document length does not pose an issue. They further proposed overcoming the challenge of long documents by applying inference on each individual sentence and combining the top scores to compute a final document score. Their approach was motivated by user studies? by Zhang et al. [37] who suggested that the most relevant sentence or paragraph in a document provides a good proxy for overall document relevance. Their work paved the way for future work that culminated in this thesis.

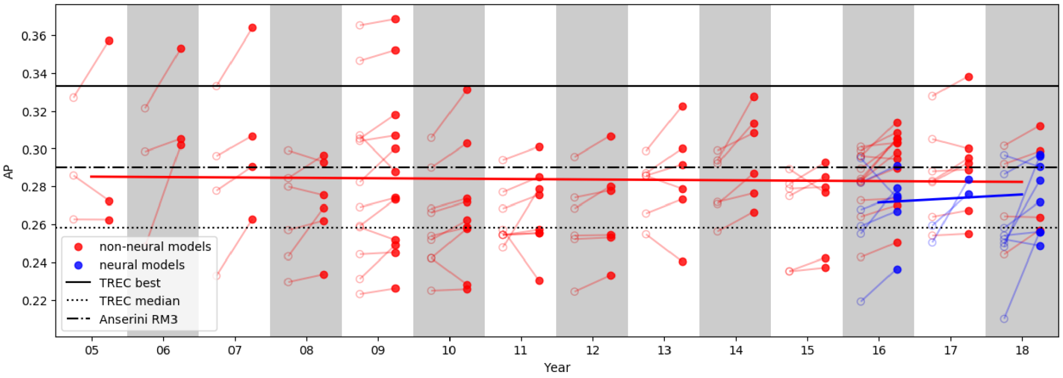


Figure 2.3: ...

More recently, MacAvaney et al. [12] shifted focus from incorporating BERT as a reranker to using its representation capabilities to improve existing neural architectures. By computing a relevance matrix between the query and each candidate document? at each layer of a contextualized language model – ELMo or BERT – they established state-of-the-art effectiveness on Robust04 and Webtrack 2012–2014 at the time of writing. Not MAP... They also proposed a joint model that combines the aforementioned classification mechanism of BERT into existing neural architectures. They claim that this approach benefits from both deep semantic matching with BERT *and* relevance batching with traditional ranking architectures.

Qiao et al. [21] study in an arXiv preprint the performance and behavior of BERT when used as a reranker for passage ranking on MS MARCO and for document ranking on the TREC Web Track. Their findings are consistent with those of Nogueira et al. [16] in that BERT outperforms previous neural models on the passage reranking task on MS MARCO. For ad hoc document ranking, they explore using BERT as representation-based and interaction-based rankers and in combination with some of the neural models discussed in Section X. However, they find that BERT, even when pretrained on MS MARCO ranking labels, performs worse than feature-based learning to rank and neural retrieval models pretrained user clicks in Bing log. They also compare their BERT-based reranker to Conv-KNRM [3]. Compared to interaction-based neural models that focus on term pairs, they show that BERT propagates information across larger spans through its attention mechanism.

Padigela et al. [17] also focus on understanding the reasons behind the gains on passage reranking on MS MARCO. They also demonstrate that fine-tuning BERT is more effective than feature based learning to rank models such as RankSVM [**?**] and a number of neural kernel matching models such as KNRM [31] and Conv-KNRM [3]. They test four hypotheses regarding the behavior of semantic matching with BERT compared to BM25; specifically, with respect to term frequency and document length. More?

2.3 Evaluation Metrics

The standard approach to evaluation in information retrieval relies on the distinction between "relevant" and "irrelevant" documents with respect to an information need as expressed by a query. A number of automatic evaluation metrics has been formalized for ranking tasks such as document retrieval. These metrics rely on The size of most document collections makes it infeasible for humans to manually judge the relevance of all documents. All relevant documents need to be labelled to prevent false negatives, i.e: treating documents which are in fact relevant as irrelevant.

2.3.1 Mean Average Precision (MAP)

Precision specifies what fraction of a set of retrieved documents is in fact relevant for a given query q. Precision can easily be extended to evaluate ranked retrieval results by ... Average precision (AP) expresses the average of the precision values obtained for the set of top k documents for a query. Support that $D = \{d_1, ..., d_{m_j}\}$ is the set of all relevant documents for a query q_j, then AP can be formulated as:

$$AP = \frac{1}{m_j} \sum_{k=1}^{m_j} P(R_{jk}) \tag{2.1}$$

where R_{jk} represents the set of top k tanked retrieval results.

The respective AP for each query can be aggregated to obtain mean average precision (MAP) for the overall retrieval effectiveness in the form of a single-figure measure of quality across various recall levels:

$$MAP = \frac{\sum_{j=1}^{|Q|} AP}{Q} = \frac{1}{Q} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} P(R_{jk}) \tag{2.2}$$

It has been show to have especially good discrimination and stability compared to other metrics, which makes it the ideal choice for large text collections [13]. It is hence one of the standard metrics among the TREC community.

2.3.2 Precision at k (P@k)

Unlike MAP which factors in precision at all recall levels, certain applications have a distinctly different notion for ranking quality. Particularly in the case of web search, the user often only cares about the results on the first page or two, but not all of them. This restriction essentially leads to measuring precision at fixed low levels of retrieved results, i.e: top k documents – hence the name for metric "precision at k". On the one hand, it eliminates the need for any estimate of the size of the set of relevant documents. However, it also produces the least stable out of all measures. Moreover, precision at k does not average well because the total number of relevant documents for a query has a very strong influence on its value.

2.3.3 Normalized Discounted Cumulative Gain (NDCG@20)

Cumulative gain (CG) simply computes the sum of relevance labels for all the retrieved documents. It views the search results as an unordered set, and disregards the ordering of the documents. Since a highly relevant document is inherently more useful when it appears higher up in the search results, CG has been extended to discounted cumulative gain (DCG). Discounted cumulative gain (DCG) estimates the relevance of a document based on its rank among the retrieved documents. The relevance measure is accumulated from top to bottom, discounting the value of documents at lower ranks. NDCG measures DCG for the top k documents, normalizing by the highest possible value for a query.

NDCG is uniquely useful in applications with a non-binary notion of relevance, e.g: a spectrum of relevance. For this reason, NDCG is a popular choice for systems with machine learning approaches? Like precision at k, it is evaluated as a weighted sum over the top k search results, and normalized so that a perfect ranking yields NDCG equals 1. This makes NDCG comparable across different queries: The NDCG values for all queries can be averaged to reliably evaluate the effectiveness of a ranking algorithm for various information needs across a collection. However, the use of NDCG is dependent on the availability of ground truth relevance labels?

Chapter 3

Cross-Domain Relevance Transfer with BERT

3.1 Datasets

Add statistics and examples

Elaborate on splits

Add motivation, gist of each dataset

Add length distribution

3.1.1 Fine-Tuning

As discussed in Section 1, applying BERT to document retrieval requires leveraging passage- or sentence-level relevance judgements fortuitously available in large text collections. Since no such newswire collection currently exists, we train the BERT relevance classifier on three out-of-domain collections.

TREC Microblog (MB)

TREC Microblog datasets draw from the Microblog Tracks at TREC from 2011 to 2014, with topics (i.e., queries) and relevance judgments over tweets. We use the data prepared by Rao et al. [24]. We extract the queries, tweets and relevance judgements from the dataset,

Type	**Training Set**	**Validation Set**	**Total**
Number of queries	166	59	225
Number of tweets	asd	asd	asd
Percentage of relevant tweets	asd	asd	asd

Table 3.1: ...

excluding metadata such as query time and URLs of the tweets. Relevance judgements in Microblog are in fact reported in a three-point scale where ("irrelevant", "relevant" and "highly relevant"); however, for the purposes of this work we treat both degrees of relevance as equal. We tokenize both query and tweets...

We sample 25% of the data for the validation set, and use the rest for fine-tuning BERT. What else?

Add total

MicroSoft MAchine Reading Comprehension (MS MARCO)

MS MARCO [1] features user queries sampled from Bing's search logs and passages extracted from web documents. The dataset is composed of tuples of a query with relevant and non-relevant passages. Each query is associated with sparse relevance judgments by human editors. In other words, some queries may have no relevant passage. Because the dataset was initially built from top 10 passages from the Bing search engine and then annotated, some relevant passages might not have been retrieved with BM25. On average, however, each query is associated with one relevant passage.

The models in Section ref were trained on less than 2% of the total training set due to the size of the dataset and time required to train on it even on TPUs. According to

Query: bbc world service staff cuts
Text: irish times : bbc world service confirms cuts : the bbc world service will shed around 650 jobs or more
Relevance: 1 ("relevant")

Figure 3.1: An example of... MB

Nogueira et al. [16], training for up to 12.5% of the total data does not improve MRR@10 on the validation set.

TODO

TREC Complex Answer Retrieval (CAR)

TREC CAR [6] uses queries and paragraphs extracted from English Wikipedia: each query is formed by concatenating an article title and a section heading, and passages in that section are considered relevant. This makes CAR, essentially, a synthetic dataset. ...

TODO

Type	**Training Set**	**Validation Set**	**Total**
Number of queries	12.8M	asd	asd
Number of ?	asd	asd	asd
Percentage of relevant passages	asd	asd	asd

Table 3.2: ...

Query: bbc world service staff cuts
Text: irish times : bbc world service confirms cuts : the bbc world service will shed around 650 jobs or more
Relevance: 1 (“relevant”)

Figure 3.2: An example of... MS MARCO

Query: bbc world service staff cuts
Text: irish times : bbc world service confirms cuts : the bbc world service will shed around 650 jobs or more
Relevance: 1 (“relevant”)

Figure 3.3: An example of... CAR

3.1.2 Evaluation

We conduct end-to-end document ranking experiments on three TREC newswire collections: the Robust Track from 2004 (Robust04) and the Common Core Tracks from 2017 and 2018 (Core17 and Core18).

Robust04

Robust04 comprises 250 topics, with relevance judgments on a collection of 500K documents (TREC Disks 4 and 5 except the Congressional Record). We apply five-fold cross-validation [] on...

Core17 & Core18

Core17 and Core18 have only 50 topics each; the former uses 1.8M articles from the New York Times Annotated Corpus while the latter uses around 600K articles from the TREC Washington Post Corpus. Motivation for this dataset? Similar to Robust04, we split the 50 topics into ...

Type	**Training Set**	**Validation Set**	**Total**
Number of queries	12.8M	asd	asd
Number of ?	asd	asd	asd
Percentage of relevant passages	asd	asd	asd

Table 3.3: ...

References

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv:1611.09268v3*, 2018.

[2] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 758–759, New York, NY, USA, 2009. ACM.

[3] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 126–134. ACM, 2018.

[4] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019.

[6] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. TREC Complex Answer Retrieval overview. In *Proceedings of the Twenty-Sixth Text REtrieval Conference (TREC 2017)*, Gaithersburg, Maryland, 2017.

[7] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64. ACM, 2016.

[8] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. *CoRR*, abs/1711.08611, 2017.

[9] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338. ACM, 2013.

[10] Jimmy Lin. The neural hype and comparisons against weak baselines. In *ACM SIGIR Forum*, volume 52, pages 40–51. ACM, 2019.

[11] Zachary C Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship. *arXiv preprint arXiv:1807.03341*, 2018.

[12] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. CEDR: contextualized embeddings for document ranking. *CoRR*, abs/1904.07094, 2019.

[13] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[15] Bhaskar Mitra and Nick Craswell. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*, 2017.

[16] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv:1901.04085*, 2019.

[17] Harshith Padigela, Hamed Zamani, and W. Bruce Croft. Investigating the successes and failures of BERT for passage re-ranking. *arXiv:1905.01758*, 2019.

[18] Harshith Padigela, Hamed Zamani, and W. Bruce Croft. Investigating the successes and failures of BERT for passage re-ranking. *CoRR*, abs/1905.01758, 2019.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[20] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[21] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. Understanding the behaviors of BERT in ranking. *arXiv:1904.07531*, 2019.

[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

[23] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[24] Jinfeng Rao, Wei Yang, Yuhao Zhang, Ferhan Türe, and Jimmy Lin. Multi-perspective relevance matching with hierarchical convnets for social media search. *CoRR*, abs/1805.08159, 2018.

[25] D Sculley, Jasper Snoek, Alex Wiltschko, and Ali Rahimi. Winner's curse? on pace, progress, and empirical rigor. 2018.

[26] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 373–374. ACM, 2014.

[27] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[28] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[30] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[31] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. *CoRR*, abs/1706.06613, 2017.

[32] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Reproducible ranking baselines using lucene. *J. Data and Information Quality*, 10(4):16:1–16:20, October 2018.

[33] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. Critically examining the "neural hype": Weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*, pages 1129–1132, Paris, France, 2019.

[34] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*, 2019.

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

[36] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 497–506. ACM, 2018.

[37] Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Mark D Smucker, Gordon V Cormack, and Maura R Grossman. Effective user interaction for high-recall retrieval: Less is more. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 187–196. ACM, 2018.

[38] Le Zhao and Jamie Callan. Term necessity prediction. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 259–268. ACM, 2010.