Chapter 1

Experimental Results

Tables 1.1, 1.2 and 1.3 displays our main results on Robust04, Core17 and Core18. The top row (BM25+RM3) corresponds to the BM25 runs with RM3 query expansion using default Anserini parameters. Although higher scores could be obtained on Robust04 with tuned parameters, we present untuned runs for the sake of fairness as no careful tuning has been performed for Core17 or Core18. The remaining five blocks show the retrieval

Model	MAP	P@20	NDCG@20
BM25+RM3	0.2903	0.3821	0.4407
1S: BERT(MB)	0.3408†	0.4335†	0.4900†
2S: BERT(MB)	0.3435†	0.4386†	0.4964†
3S: BERT(MB)	0.3434†	0.4422†	0.4998†
1S: BERT(CAR)	0.3025†	0.3970†	0.4509
2S: BERT(CAR)	0.3025†	0.3970†	0.4509
3S: BERT(CAR)	0.3025†	0.3970†	0.4509
1S: BERT(MS MARCO)	0.3028†	0.3964†	0.4512
2S: BERT(MS MARCO)	0.3028†	0.3964†	0.4512
3S: BERT(MS MARCO)	0.3028†	0.3964†	0.4512
1S: BERT(CAR $\rightarrow$ MB)	0.3476†	0.4380†	0.4988†
2S: BERT(CAR $\rightarrow$ MB)	0.3470†	0.4400†	0.5015†
3S: BERT(CAR $\rightarrow$ MB)	0.3466†	0.4398†	0.5014†
1S: BERT(MS MARCO $\rightarrow$ MB)	0.3676†	0.4610†	0.5239†
2S: BERT(MS MARCO $\rightarrow$ MB)	**0.3697**†	0.4657†	0.5324†
3S: BERT(MS MARCO $\rightarrow$ MB)	0.3691†	**0.4669**†	**0.5325**†

Table 1.1: Ranking effectiveness on Robust04.

Model	MAP	P@20	NDCG@20
BM25+RM3	0.2823	0.5500	0.4467
1S: BERT(MB)	0.3091†	0.5620	0.4628
2S: BERT(MB)	0.3137†	0.5770	0.4781
3S: BERT(MB)	0.3154†	0.5880	0.4852†
1S: BERT(CAR)	0.2814†	0.5500	0.4470
2S: BERT(CAR)	0.2814†	0.5500	0.4470
3S: BERT(CAR)	0.2814†	0.5500	0.4470
1S: BERT(MS MARCO)	0.2817†	0.5500	0.4468
2S: BERT(MS MARCO)	0.2817†	0.5500	0.4468
3S: BERT(MS MARCO)	0.2817†	0.5500	0.4468
1S: BERT(CAR $\rightarrow$ MB)	0.3103†	0.5830	0.4758
2S: BERT(CAR $\rightarrow$ MB)	0.3140†	0.5830	0.4817†
3S: BERT(CAR $\rightarrow$ MB)	0.3143†	0.5830	0.4807
1S: BERT(MS MARCO $\rightarrow$ MB)	0.3292†	0.6080†	0.5061†
2S: BERT(MS MARCO $\rightarrow$ MB)	**0.3323**†	0.6170†	**0.5092**†
3S: BERT(MS MARCO $\rightarrow$ MB)	0.3314†	**0.6200**†	0.5070†

Table 1.2: Ranking effectiveness on Core17.

Model	MAP	P@20	NDCG@20
BM25+RM3	0.3135	0.4700	0.4604
1S: BERT(MB)	0.3393†	0.4930	0.4848†
2S: BERT(MB)	0.3421†	0.4910	0.4857†
3S: BERT(MB)	0.3419†	0.4950†	0.4878†
1S: BERT(CAR)	0.3120	0.4680	0.4586
2S: BERT(CAR)	0.3116	0.4670	0.4585
3S: BERT(CAR)	0.3113	0.4670	0.4584
1S: BERT(MS MARCO)	0.3121	0.4670	0.4594
2S: BERT(MS MARCO)	0.3121	0.4670	0.4594
3S: BERT(MS MARCO)	0.3121	0.4670	0.4594
1S: BERT(CAR $\rightarrow$ MB)	0.3385†	0.4860	0.4785
2S: BERT(CAR $\rightarrow$ MB)	0.3386†	0.4810	0.4755
3S: BERT(CAR $\rightarrow$ MB)	0.3382†	0.4830	0.4731
1S: BERT(MS MARCO $\rightarrow$ MB)	0.3486†	**0.4920**	**0.4953**†
2S: BERT(MS MARCO $\rightarrow$ MB)	0.3496†	0.4830	0.4899†
3S: BERT(MS MARCO $\rightarrow$ MB)	**0.3522**†	0.4850	0.4899†

Table 1.3: Ranking effectiveness on Core18.

effectiveness of our models trained as described in Chapter **??**. The models are labeled to reflect the fine-tuning procedure where the datasets that BERT was trained on are listed in order inside parentheses. For example, MS MARCO $\rightarrow$ MB refers to a model that was first fine-tuned on MS MARCO and then on MB. The nS preceding the model name indicates that the top n sentences in each document were interpolate to compute an overall document score. The main result tables also highlights statistically significant results based on paired t-tests compared to the BM25+RM3 baseline with a †. We report significance at the $p < 0.01$ level, with appropriate Bonferroni corrections for multiple hypothesis testing.

1.1 Effect of Training Data

By fine-tuning BERT on three different datasets alone and in combination, we hope to study the effect of nature and amount of training data on the power of our learned relevance matching model. As seen in Tables 1.1, 1.2 and 1.3, the particular source of relevance judgements that we train BERT on substantially influences retrieval effectiveness across all three test collections.

First of all, we find that fine-tuning BERT on MB alone (BERT(MB)) significantly outperforms the BM25+RM3 baseline for all metrics on Robust04. We also observe significant increases in AP on Core17 and Core18, as well as in P@20 and NDCG@20 in some cases. These results confirm that relevance models learned from tweets can be successfully transferred to news articles in spite of the considerable differences in domain. This surprising finding may be attributed to the relevance matching power enabled with deep semantic information learned by BERT.

Contrary to the large gains brought by fine-tuning on MB, fine-tuning on CAR or MS MARCO alone results in marginal gains over the baseline on Robust04. Re-ranking with these models in fact hurts effectiveness on Core17 and Core18 for most metrics. The synthetic nature of CAR data especially does not appear to be useful for relevance modeling on newswire collections. For instance, using BERT(CAR) gives 0.3120 AP on Core18 compared to the 0.3135 AP of the baseline, which means that the model actually fails to score relevant sentences highly. As the 2S and 3S results for the same model show, the effectiveness on Core18 in fact progressively degrades the more sentences are considered in final score aggregation. Intuitively, these results indicate that using these models incorrectly disrupts the better order imposed by the baseline.

Results for BERT(MS MARCO) are more surprising since the web passages in the MS MARCO dataset are "closer" to the news articles in the test collections than MB. Given the

proximity of the domains, it would be reasonable to expect relevance transfer between MS MARCO and newswire collections to be more effective. However, our results reveal that this is not necessarily the case, and fine-tuning on MS MARCO alone is far less effective for relevance transfer than fine-tuning on MB alone.

Although fine-tuning on only CAR or MS MARCO does not yield large improvements, we actually obtain considerably higher results by fine-tuning these models further on MB. With BERT(CAR $\rightarrow$ MB) we achieve effectiveness that is slightly better than fine-tuning on MB alone in some cases. We hypothesize that CAR might have a similar effect to language pre-training in that it doesn't directly apply to the downstream document retrieval task, but provides a better representation that can benefit from fine-tuning on MB. More surprisingly, fine-tuning on MS MARCO first and then on MB represents our best model (BERT(MS MARCO $\rightarrow$ MB)) as shown in the final block of the table. This model is able to exploit data from both MS MARCO and MB, with a score that is higher than fine-tuning on each dataset alone.

1.2 Number of Sentences

We consider up to three top scoring sentences in each document to re-rank documents. Our main results suggest that the top scoring sentence by itself is often a good indicator of overall document relevance; in fact we achieve the best AP in about half of the experiments by only considering the most relevant sentence of the document. This finding is consistent with the results of Zhang et al. [**?**] who found through user studies that the most relevant sentence or paragraph in a document provides a reliable proxy for document relevance.

Considering the next most relevant sentence in addition to the top sentence yields a noticeable increase in some of the experiments, such as BERT(MS MARCO $\rightarrow$ MB) on Robust04 and Core18. However, we find that adding a third in fact causes effectiveness to drop in some cases. Preliminary experiments show that in general looking beyond the top three sentence doesn't help effectiveness. These results suggest that document ranking may be distilled into relevance prediction primarily at the sentence level.

1.3 Comparison to Other Ranking Models

In this section we assess our results in the broader context of document re-ranking literature. The meta-analysis[1] of over 100 papers up until 2019 by [?] currently provides the most thorough overview of related work on Robust04. Following the same cross-validation settings as in this thesis to re-rank a strong BM25 baseline, they report the most effective neural model to be DRMM [?] at 0.3152 AP and 0.4718 NDCG@20. In comparison, with our best model BERT(MS MARCO → MB) we report the highest AP that we are aware of at 0.3697. Furthermore, our results also exceed the previous highest known score of 0.3686, which is a non-neural method based on ensembles [?].

More recently, MacAvaney et al. [?] reported 0.5381 NDCG@20 on Robust04 by integrating contextualized word embeddings into existing neural ranking models. Our best NDCG@20 on Robust04 at 0.5325 approaches their results even though we optimize for AP instead of NDCG@20. Furthermore, since we are only using Robust04 data for hyperparameter tuning in Eq (??), and not for fine-tuning BERT itself, it is less likely that we are overfitting.

Our best model also achieves a higher AP on Core17 than the best TREC submission that does not make use of past labels or human intervention (`umass_baselnrm`, 0.275 AP) [?]. Under similar conditions, we beat every TREC submission in Core18 as well (with the best run being `uwmrg`, 0.276 AP) [?]. Core17 and Core18 are relatively new and thus have yet to receive much attention from researchers, but to our knowledge, these figures represent the state of the art.

1.4 Per-Query Analysis

Our results in Table 1.1 serve as an overview of the effect of training data on cross-domain relevance transfer. However, they do not reveal much with respect to the particular strengths and weaknesses of each model compared to the baseline. To gain further insight into the characteristics of our models, we analyze the per-query retrieval effectiveness of each model compared to the baseline on Robust04, Core17 and Core18.

Figure 1.1 plot the Δ AP per query between our best model, BERT(MS MARCO → MB), and the baseline sorted in descending order. BERT(MS MARCO → MB) performs better than the baseline for 83%, 88% and 84% of the queries on Robust04, Core17

[1] https://github.com/lintool/robust04-analysis

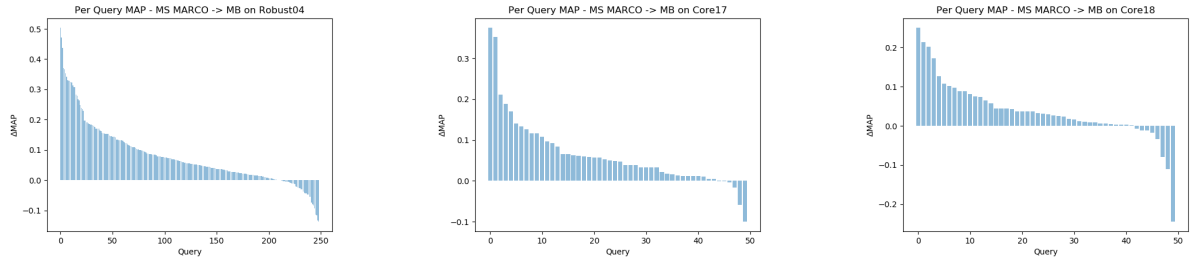

Figure 1.1: Per-query difference in AP between BERT(MS MARCO → MB) and the BM25+RM3 baseline on Robust04, Core17 and Core18.

and Core18 respectively, highlighting the usefulness of relevance signals learned from MS MARCO and MB with BERT. One of the best performing queries across all three collections is `human stampede` while one of the worst is `flavr savr tomato`. Manual inspection of top scoring sentences for `flavr savr tomato` show that our model matches terms semantically close to "tomato" but not relevant to the query phrase itself while the BM25+RM3 baseline looks for direct query matches, therefore ranking documents correctly.

Repeating the same analysis for our worst model BERT(CAR), we present the results in Figure 1.2. This model outperforms the baseline in 57%, 80% and 34% of the Robust04, Core17 and Core18 queries respectively. Not only does this model perform less accurately for much fewer queries, but it is also less stable than BERT(MS MARCO → MB). The queries `polygamy polyandry polygyny` and `women driving in Saudi Arabia` appear to be particularly difficult for BERT based models to correctly match.

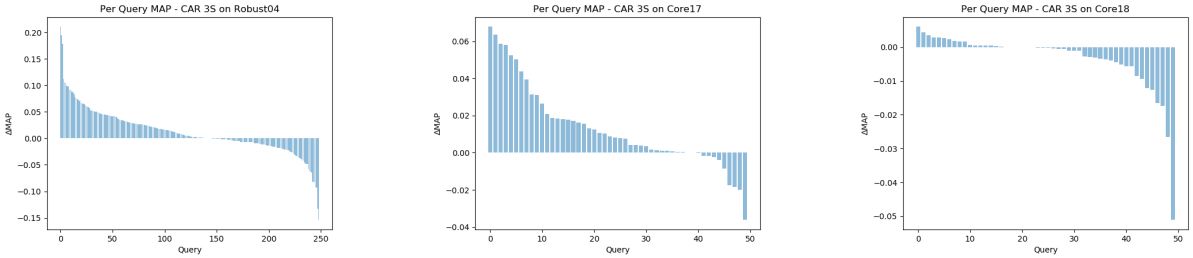

Figure 1.2: Per-query difference in AP between BERT(CAR) and the BM25+RM3 baseline on Robust04, Core17 and Core18.

1.5 Effect of Length

1.5.1 Query Length

To investigate the ability of BERT to exploit context-aware representations of the queries, we study the trend of effectiveness across increasing query lengths on Robust04. We conjecture that longer queries would give our models richer context to work with, therefore increasing retrieval effectiveness. To this end, we categorize the 250 queries from the Robust Track by the number of tokens, and calculate the average AP per query length, i.e: from 1 to 5. The majority of the queries (56%) are composed of three tokens, and only a small fraction (5%) have either only a single token or five tokens. This approach is somewhat limited by the narrow range of query lengths, but still gives insight into the effect of query length on each model's effectiveness.

The average AP per query length for each model is shown in Table 1.4. We only report the best results for each model considering any number of top sentences. The average AP for the BM25+RM3 baseline drops by 24% from a single token query to a query with five tokens, indicating that exact term-matching is more effective for fewer number of terms. Furthermore, the improvements introduced by our models follow a similar pattern to when they are evaluated over all queries as shown in Table **??**, with BERT(MS MARCO → MB) being the best model across all query lengths. Similar to the baseline, BERT(CAR) and BERT(MS MARCO) experience a gradual decline in AP as the query length increases from 1 to 5, leading to an overall 29% and 28% decrease respectively. Note, however, that the decrease in AP is not continuous at each step but reaches a minimum at query length 3 and a maximum at query length 4.

More interestingly, while increasing query length results in a decrease in AP across

	Query Length				
Model	**1**	**2**	**3**	**4**	**5**
BM25+RM3	0.3889	0.2975	0.2760	0.3089	0.2972
BERT(MB)	0.4145	0.3624	0.3230	0.3681	0.4172
BERT(CAR)	0.3978	0.3112	0.2871	0.3288	0.2844
BERT(MS MARCO)	0.3997	0.3117	0.2872	0.3289	0.2871
BERT(CAR → MB)	0.4202	0.3603	0.3296	0.3757	0.4926
BERT(MS MARCO → MB)	0.4253	0.3885	0.3480	0.4077	0.5627

Table 1.4: Average AP with respect to query length on Robust04.

	Robust04		
Model	**AP**	**P@20**	**NDCG@20**
BERT(CAR*)	0.3030	0.3980	0.4520
BERT(MS MARCO*)	0.3300	0.4309	0.4906

Table 1.5: Ranking effectiveness on shortened MS MARCO and CAR evaluated on Robust04.

all the other models, BERT(MB), BERT(CAR → MB) and BERT(MS MARCO → MB) in fact perform better at query length 5 than at 1. However, due to the fairly small sample size for query length 5 it is not possible to draw generalizable conclusions from this observation. Nevertheless, the effectiveness of the best model BERT(MS MARCO → MB) degrades much less where the minimum usually occurs, i.e: query length 3, compared to the baseline, BERT(CAR) and BERT(MS MARCO), showing that they are indeed able to use contextualized query representations most effectively.

1.5.2 Document Length

It is interesting to note that fine-tuning on MS MARCO or CAR alone results in marginal improvements over the baseline, if any, as shown in the second and third blocks of Table **??**. Considering any number of top scoring sentences, BERT(CAR) and BERT(MS MARCO) both outperform the BM25+RM3 baseline by 1.2 AP on Robust04 with similar gains in P@20 and NDCG@20. Although still statistically significant, these improvements are much lower than those gained with fine-tuning on MB (5 AP). Moreover, both models in fact perform worse than the baseline on both Core17 and Core18.

We attribute this observation to the length mismatch between the training and evaluation text lengths. After splitting documents into chunks that fit the maximum input that BERT can handle, the average number of tokens in Robust04, Core17 and Core17 sentences is 19, which is fairly close to the the average number of tokens in tweets (15). However, MS MARCO and CAR passages are much longer. It is possible that the difference in document length may be causing issues with relevance transfer across domains.

To validate this hypothesis, we divide MS MARCO and CAR passages into chunks the same size as Robust04 sentences, and train BERT for relevance prediction. The ranking effectiveness of BERT(CAR*) and BERT(MS MARCO*) trained on the shortened data is shown in Table 1.5. For BERT(CAR*), AP only increases by 0.5 upon fine-tuning on

	Pruned Robust04		
Model	**AP**	**P@20**	**NDCG@20**
BM25+RM3	0.2903	0.3821	0.4407
BERT(MB)	0.3031	0.4014	0.4580
BERT(CAR)	0.2959	0.3936	0.4480
BERT(MS MARCO)	0.2962	0.3936	0.4483
BERT(CAR → MB)	0.3037	0.3998	0.4527
BERT(MS MARCO → MB)	0.3101	0.4102	0.4639

Table 1.6: Retrieval effectiveness on pruned Robust04.

the shortened data, and is still not comparable to BERT(MB). However, the change in document length results in an increase of almost 3 AP for BERT(MS MARCO*), thus approaching BERT(MB) in all metrics. From this finding we infer that while comparable document length is an important consideration for cross-domain relevance transfer, there may be other factors at play that need to be investigated.

1.6 Semantic Matching

To isolate the contribution of BERT, we filter sentences in Robust04 that don't contain any of the query terms. This essentially eliminates the impact of exact matching on the sentence relevance scores, allowing us to verify whether BERT successfully leverages semantic cues in the documents. Table 1.6 displays the retrieval effectiveness of all models on the "pruned" Robust04 dataset. Similar to Table 1.4 we only report the best results for each model.

As expected, filtering Robust04 sentences leads to a decrease in all metrics across all models, which indicates that exact matching signals are still valuable in relevance predictions over long documents. However, notice that the all models still perform better than the baseline; in other words, they indeed successfully perform semantic matching with notable gains. The improvements over the baseline follow the same trend as the results in Table **??** for all models. While BERT(CAR) and BERT(MS MARCO) yield minor improvements over the baseline when sentences that contain exact query matches are removed, the best performing model BERT(MS MARCO → MB) still significantly outperforms the baseline. Furthermore, the drop in AP caused by filtering sentences is also the highest for the best

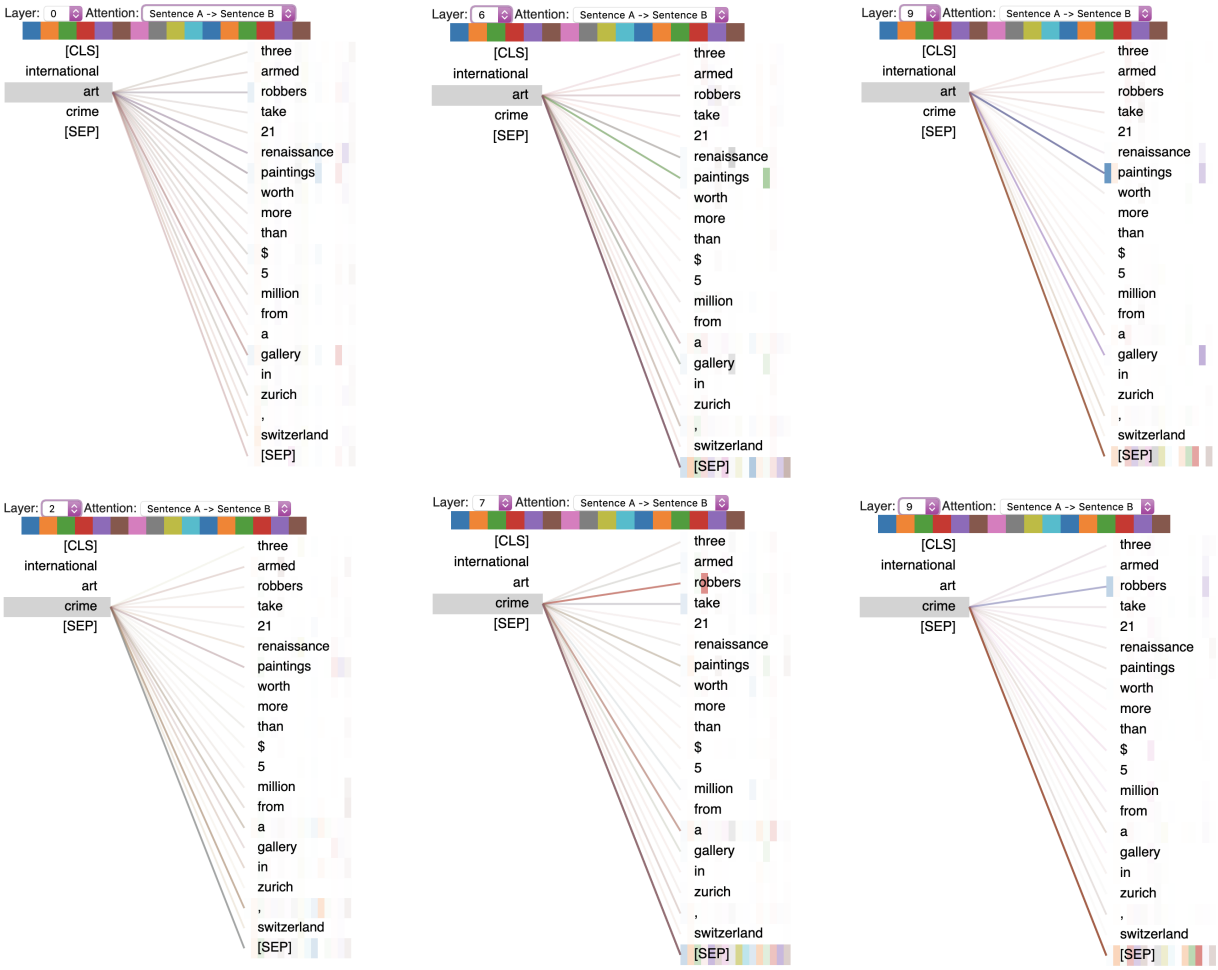

Figure 1.3: Attention visualizations of BERT(MS MARCO → MB) for a sentence with a high BERT score for the query `international art crime`.

performing model, indicating that this model is able to exploit both exact and semantic matching signals. The overall effectiveness of this model on the original Robust04 dataset may be owing to the joint relevance matching power demonstrated in this experiment.

To further investigate the semantic matching power of BERT, we visualize the attention of our models at different layers. Having been pretrained on massive amounts of data for language modeling, we expect BERT to capture various semantic relationships helpful for relevance prediction. Figure 1.3 visualizes the attention of our best model BERT(MS MARCO $\rightarrow$ MB) for one of the top scoring sentences for the query 322 `international art crime`. Note that the sentence does not contain any exact matches with the query terms and is nonetheless relevant to the query. The top three images illustrate the attention for the query term "art" at increasingly deeper layers (0, 6 and 9) and the bottom three for the query term "crime". We see that the model attends to related terms such as "renaissance" and "paintings" for the term "art", and "robbers" for crime, therefore giving this particular sentence a high relevance score. On the other hand, these terms are not recognized by term-matching techniques like BM25 as relevant. This highlights how the semantic knowledge captured by BERT directly help with relevance modeling on newswire articles.