# Another Look at Information Retrieval as Statistical Translation

Yuqi Liu,* Chengcheng Hu,* and Jimmy Lin
David R. Cheriton School of Computer Science
University of Waterloo, Ontario, Canada

## ABSTRACT

Over two decades ago, Berger and Lafferty proposed "information retrieval as statistical translation" (IRST), a simple and elegant method for ad hoc retrieval based on the noisy channel model. At the time, they lacked the large-scale human-annotated datasets necessary to properly train their models. In this paper, we ask the simple question: What if Berger and Lafferty had access to datasets such as the MS MARCO passage ranking dataset that we take for granted today? The answer to this question tells us how much of recent improvements in ranking can be solely attributed to having more data available, as opposed to improvements in models (e.g., pretrained transformers) and optimization techniques (e.g., contrastive loss). In fact, Boytsov and Kolter recently began to answer this question with a replication of Berger and Lafferty's model, and this work can be viewed as another independent replication effort, with generalizations to additional conditions not previously explored, including replacing the sum of translation probabilities with ColBERT's MaxSim operator. We confirm that while neural models (particularly pretrained transformers) have indeed led to great advances in retrieval effectiveness, the IRST model proposed decades ago is quite effective if provided sufficient training data.

## CCS CONCEPTS

• **Information systems → Language models**.

## KEYWORDS

MS MARCO, noisy channel model

**ACM Reference Format:**
Yuqi Liu, Chengcheng Hu, and Jimmy Lin. 2022. Another Look at Information Retrieval as Statistical Translation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531717

## 1 INTRODUCTION

There are three crucial ingredients to any solution based on machine learning: the model itself (linear, tree-based, neural, etc.), the training data (quality, quantity, domain, as well as many other considerations), and the optimization strategy (the loss function, the optimizer itself, hyperparameters, etc.). With the advent and dominance of neural approaches to information retrieval, we have seen significant advances in all three. There is no dispute that neural models today—particularly pretrained transformers—are tremendously powerful and excel at a wide range of retrieval tasks [17, 19], including in specialized domains such as medicine [26]. We have seen corresponding advances in different optimization strategies: most recently, a body of research has shown that different loss functions contribute significantly to effectiveness gains [13, 28]. In the realm of data, the MS MARCO datasets have been a huge boon for the community, driving gains in effectiveness by feeding data-hungry neural models.

Studies of neural ranking models today mostly take for granted training on large datasets with human-labeled relevance judgments such as MS MARCO and instead primarily focus on model and optimization advances [19]. In this paper, we focus on the issue of data availability and try to answer the following question: How much of recent effectiveness gains can be attributed *solely to the availability of more data*?

To this end, we revisit the "information retrieval as statistical translation" (IRST) model proposed by Berger and Lafferty [5] from 1999—over two decades ago (and from the previous millennium)! We ask: How effective would their model be if given access to the training data available today? That is, let us pretend that modeling and optimization advances (e.g., the "neural revolution") in the intervening two decades never occurred. The answer to this question allows us to more appropriately assign "credit" to the role of simply having more data in driving advances in retrieval effectiveness. For this exercise, we set aside the issue of computational power, which is obviously an important consideration [14]. It is not clear if translation models at the scale described in this paper can be trained on circa 1999 commodity hardware, but let us ignore this potential limitation here.

Our research question has been asked and previously answered by Boytsov and Kolter [6] (building on the work in Boytsov and Nyberg [7]), and thus our work here can be viewed as an independent replication of that work, which itself re-implemented the ideas of Berger and Lafferty. We confirm that, indeed, the decades-old "information retrieval as statistical translation" (IRST) model can be quite effective if provided sufficient training data.

Going beyond the work of Boytsov and Kolter [6], we generalized their findings to experimental conditions not previously explored, including replacing the sum of translation probabilities with ColBERT's MaxSim operator. Evaluations on test collections from the TREC Deep Learning Tracks show that implementations of the IRST model are on par with or better than the best "traditional" (i.e., non-neural) methods today in terms of effectiveness, based on results from TREC 2019 and 2020 [10, 11]. Finally, all of our models are integrated into the Pyserini IR toolkit [18], which provides state-of-the-art, easily reproducible non-neural baselines for

* Both authors contributed equally.


*SIGIR '22, July 11–15, 2022, Madrid, Spain*

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531717

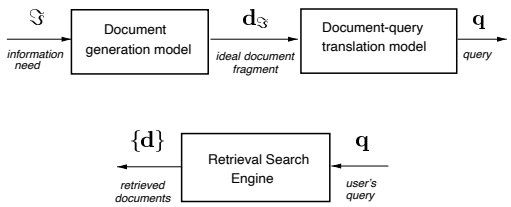


**Figure 1: Model of query generation and retrieval, taken from Berger and Lafferty [5].**

the community. Our work provides a foundation for further explorations that can help researchers more clearly understand the impact of neural methods.

## 2 BACKGROUND

In their classic paper "Information Retrieval as Statistical Translation", published in SIGIR 1999, Berger and Lafferty [5] formulated a noisy channel model of ad hoc retrieval. In their own words:

> In formulating a query to a retrieval system, we imagine that a user begins with an information need. This information need is then represented as a fragment of an "ideal document"—a portion of the type of document that the user hopes to receive from the system.

Query formulation can be thought of as a "corruption" of this ideal document fragment, i.e., that has been passed through a noisy channel. Thus:

> Given **q** and a model of the channel—how an information need gets corrupted into a query—the retrieval system's task is to identify those documents **d** that best satisfy the information need of the user.

Berger and Lafferty's model of query generation and retrieval is shown in Figure 1, taken directly from their paper. They argued that "translations" between query and document terms are no different from translations between different languages, hence the idea of "information retrieval as statistical translation". Recognizing this connection allowed the authors to exploit statistical translation models that had been developed for machine translation, namely the IBM models [8].

Berger and Lafferty then went on to derive two variant ranking methods in the language modeling framework of Ponte and Croft [24]. However, though, they soon reached an impasse:

> The key ingredient in these models is the collection of translation probabilities $t(q|w)$. But how are we to obtain these probabilities? The statistical translation strategy is to learn these probabilities from an aligned bilingual corpus of translated sentences, using the likelihood criterion. Ideally, we should have a collection of query/document pairs to learn from, obtained by human relevance judgments. But we know of no publicly-available collection of data sufficiently large to estimate parameters for general queries.

While the final statement may have been accurate in 1999, exactly such data exist today in the MS MARCO passage ranking dataset! Without access to training data, Berger and Lafferty had to make do with synthetic data, but all the major ingredients of the approach we explore here—e.g., learning translation probabilities with the IBM Model 1 translation model, ranking by query likelihood—have been in existence for over two decades.

Boytsov and Kolter [6] to our knowledge was the first to observe that with the availability of modern large-scale datasets, the approach of Berger and Lafferty was worth re-examining. The experiments in this paper can be viewed as an independent replication (i.e., an effort to recreate the experimental conditions with an independent implementation). Our results align with the original findings, confirming the effectiveness of the model. In addition, we explore more experimental conditions to generalize their findings.

## 3 METHODS

In this section, we describe the datasets, the models, and detailed configurations of our experiments.

### 3.1 Datasets

The MS MARCO passage ranking test collection [4], released in 2018, provides the crucial ingredient necessary to realize the noisy channel retrieval model of Berger and Lafferty. Microsoft released the data to help advance information retrieval research in the large-data regime [9], and it is difficult to overstate the impact that the dataset has had in driving recent progress in information retrieval by fueling data-hungry neural models.

The training set of the MS MARCO passage ranking test collection contains approximately 503k queries taken from Bing query logs. Each query is associated with around one relevant passage from a corpus of 8.8m passages that are typical of "answers" shown at the top of a search engine's result page. It is from this dataset that we learn translation probabilities.

To evaluate our models on passage ranking, we used the 6980 queries from the development set of the MS MARCO passage dataset. Additionally, we evaluated on topics from the TREC 2019 and 2020 Deep Learning Tracks [10, 11], which also used the same passage corpus. The development queries, much like the training data, are associated with only one relevant passage per query on average, and hence are usually referred to as "sparse" judgments. This contrasts with the TREC test collections, which were created via traditional approaches based on pooling.

We also evaluated our models for document ranking, using the MS MARCO document corpus, which comprises 3.2m documents from which the passage corpus was extracted. Mirroring the passage ranking conditions, we evaluated our models on the 5193 queries in the development set as well as topics from the TREC 2019 and TREC 2020 Deep Learning Tracks.

### 3.2 Estimating Translation Probabilities

Having described the data, we turn our attention to the translation model itself, which is IBM Model 1, a specific variant from

a series of models proposed by Brown et al. [8]. One of the key insights of Berger and Lafferty was to draw an analogy between machine translation (e.g., from English to French) and information retrieval. In machine translation, the IBM models define $p(\mathrm{f}; \mathrm{a}|\mathrm{e})$ for generating (say) a French sentence $\mathrm{f} = \{f_1, f_2, \ldots, f_m\}$ from (say) an English sentence $\mathrm{e} = \{e_1, e_2, \ldots e_n\}$ in terms of a hidden alignment (denoted a) between the words in the two sentences. This conceptual connection allows us to reuse much of the "machinery" developed for machine translation. Using IBM Model 1, we are able to learn the translations probabilities $T(q|d)$, which relate query terms and document terms based on the noisy channel model as outlined in Section 2.

All of these translation probabilities are learned from the training set of the MS MARCO passage ranking test collection using the FlexNeuART toolkit [7]. This part of our work is largely a reproduction of the paper by Boytsov and Kolter [6]. As the passages in the MS MARCO passage corpus are much longer than queries, we split each relevant query–passage pair $(Q, D)$ into small contiguous chunks that are around the same length as the queries. In the original work, BERT tokenization was applied to parsed text, which comprises whitespace-separated terms, retaining stopwords. In this paper, we chose to generate BERT subwords from the raw passages but with stopwords removed.

### 3.3 Ranking Documents

The translation probabilities learned from the MS MARCO passage ranking dataset comprise the heart of our "modern take" on IRST. In this section, we describe two different approaches of how these probabilities are applied to ranking. Our starting points are Equations (2) and (3) of the document ranking model based on IBM Model 1 from Boytsov and Kolter [6], which derive ultimately from Berger and Lafferty [5]. To be clear, we use "documents" here in a generic sense to refer to the unit of indexing and retrieval; in the case of passage ranking, "documents" refer to the passages that comprise the corpus.

The IRST model begins with a standard query-likelihood formulation of ranking [24], where we rank documents from a corpus based on the probability that document $D$ generated query $Q$, assuming that each query term (i.e., $q \in Q$) is generated independently (i.e., a unigram language model):

$$P(Q|D) = \prod_{q \in Q} P(q|D) \tag{1}$$

The probability of generating an individual query term $q$ can then be computed as follows:

$$P(q|D) = \sum_{d \in D} T(q|d) P(d|D) \tag{2}$$

where $T(q|d)$ comes from the probability of translating the document term $d$ into query term $q$. $P(d|D)$ is the probability of term $d$ in document $D$, which can be computed from raw counts of terms in each document $D$ (i.e., the maximum likelihood estimate). Note that the above equation sums over *unique* terms in the document, since we are working with a bag-of-words model.

It is customary to add a bit of smoothing, and with the standard approach of linear interpolation, we arrive at the following:

$$P(q|D) = (1 - \lambda) \left[ \sum_{d \in D} T(q|d) P(d|D) \right] + \lambda \cdot P(q|C) \tag{3}$$

Here, $P(q|C)$ represents the probability of query term $q$ in the entire collection (corpus). The value of the $\lambda$ parameter is determined by tuning on training data.

The second approach is motivated by the late interaction architecture of ColBERT [15]. Note that this is a novel formulation that was not originally explored by Boytsov and Kolter [6]. Instead of converting query $Q$ and document $D$ into two contextual embedding sequences (as in the full ColBERT model), we instead asked: What is the effectiveness of directly applying the MaxSim operator on translation probabilities of query–document pairs? That is, for each query term, we find the document term $d$ from $D$ that produces the highest translation score, $T(q|d)$. Note that here we are not using any of the "neural machinery" proposed in ColBERT, and thus such an implementation would have been possible in 1999.

Operationally, using MaxSim has the effect of replacing sum in Equation (3) with a max:

$$P(q|D) = (1 - \lambda) \left[ \max_{d \in D} T(q|d) P(d|D) \right] + \lambda \cdot P(q|C) \tag{4}$$

A higher translation score means that the two terms have a higher degree of 'match' under a statistical translation model. Equation (3) serves as a drop-in replacement in the query likelihood calculation in Equation (1).

### 3.4 Reranking Implementation

Since there is no straightforward way to implement IRST directly with existing query evaluation techniques on inverted indexes (because the need to score terms that do not appear in the query cannot be efficiently captured with existing index traversal algorithms), we adopted the standard "retrieve-then-rerank" approach to implement IRST. In the context of BERT-based cross-encoders, Nogueira and Cho [21] were the first to adopt this approach, but multi-stage ranking dates back well over a decade [3, 20, 23].

In our implementation, $k$ candidates in the first stage are retrieved using the Pyserini IR toolkit [18], which is built on the open-source Lucene search library. Ranking is performed with bag-of-words queries using BM25, with parameters $k_1 = 0.82$, $b = 0.68$, tuned by a grid search to optimize recall@1000 based on the authors' recommendation.[1]

The retrieved candidates are then passed to our IRST reranker, also implemented in Pyserini. As the translation model only contains translation probabilities between different terms, we also add a self-translation parameter, set to 0.35. This parameter was tuned using grid search over the range $[0.1, 0.9]$, with a step size of 0.05 on five subsets of the training set of the MS MARCO passage ranking data. Two additional details: The minimal collection probability $P(q|C)$, which denotes the probability of a query term $q$ appearing

[1] https://github.com/castorini/anserini/blob/master/docs/experiments-msmarco-passage.md

| | | MS MARCO dev | TREC 2019 | | TREC 2020 | |
|---|---|---|---|---|---|---|
| | | MRR@10 | nDCG@10 | MAP | nDCG@10 | MAP |
| | | | comparisons | | | |
| (1a) | BM25 ($k_1 = 0.82, b = 0.68$) | 0.188 | 0.497 | 0.290 | 0.488 | 0.288 |
| (1b) | BM25 + Model 1, by Boytsov and Kolter [6] | - | 0.517 | - | 0.509 | - |
| (1c) | Best "traditional" | - | 0.556 | 0.318 | 0.567 | 0.338 |
| (1d) | BM25 + monoBERT | 0.365 | 0.698 | 0.468 | - | - |
| | | | our configurations | | | |
| (2a) | BM25 + IRST (Sum) | $0.221^{1a}$ | 0.526 | $0.328^{1a}$ | 0.558 | $0.352^{1a}$ |
| (2b) | BM25 + IRST (Max) | $0.215^{1a}$ | 0.537 | $0.329^{1a}$ | 0.547 | $0.336^{1a}$ |

**Table 1: Experimental results on the MS MARCO passage corpus for the development queries and TREC topics. Superscripts on the IRST conditions indicate significant improvements over the BM25 baseline, based on paired $t$-tests ($p < 0.01$).**

in the entire collection, is set to $10^{-9}$. We set the minimum translation probability to 0.0025. Both parameters are taken directly from Boytsov and Kolter [6] without further tuning.

Based on the above setup, we use the following notation to represent our two retrieve-then-rerank configurations:

- **BM25 + IRST (Sum)**: This represents our design of applying the IRST reranker on BM25 first-stage retrieval, where reranking is performed according to Equation (3). This variant captures the original formulation of Berger and Lafferty.
- **BM25 + IRST (Max)**: This represents our variant based on ColBERT's MaxSim operator, where BM25 candidates are reranked according to Equation (4).

In both cases, we performed additional linear interpolation between (normalized) BM25 and IRST scores to generate the final ranking. We tuned the interpolation weight on five subsets of the MS MARCO passage ranking training set; this yielded a weight of 0.9 on the translation score for IRST (Sum) and 0.7 for IRST (Max). To avoid the risk of overfitting, we decided not to tune the weight parameter for document ranking.

The methods described above can rerank spans of text that are arbitrarily long, and so they can be directly applied to both passage and document ranking. However, for comparison purposes, we also evaluated IRST on document segments for document ranking. In the context of reranking with neural models—primarily to work around input length restrictions with transformers—researchers typically adopt a sliding window strategy that segments each document into multiple passages. This segmentation can be applied *prior* to indexing, where each segment becomes a unit of retrieval, or applied to the documents retrieved during first-stage retrieval. At search time, the highest score among the segments determines the document score. In the case where segmentation is applied to document candidates generated by the first stage, this is known as the MaxP approach [12] and is widely used [2, 16].

In this work, we adopt the alternative approach, where the documents are segmented prior to indexing. In our implementation, we constructed segments based on a 10-sentence sliding window with a 5-sentence stride, following Pradeep et al. [25]. Thus, our IRST reranker processes retrieved document segments from the first stage to construct the final ranking.

### 3.5 Experimental Setup

We evaluated the two configurations of IRST (Sum and Max) on the MS MARCO passage and document corpora, with queries from the development sets of the passage and document test collections as well as topics from the TREC 2019 and 2020 Deep Learning Tracks. In all cases, our final runs contained 1000 hits for each query. For the segmented document condition, we reranked the top 10000 segments from first-stage retrieval to generate the top 1000 hits. For the passage and (full) document conditions, we reranked the top 1000 hits from first-stage retrieval.

Our evaluations used standard IR metrics. For the MS MARCO development queries, we report the official metrics: MRR@10 for passage ranking and MRR@100 for document ranking. For the TREC topics, we report nDCG at rank cutoff 10 and MAP.

All experiments described in this paper can be reproduced with our open-source Pyserini IR toolkit. Detailed documentation can be found linked off the landing page of the Pyserini GitHub repository, at http://pyserini.io/.

## 4 RESULTS

We separately present results on passage ranking and document ranking experiments in this section. The high-level summary is that we were able to successfully replicate the results of Boytsov and Kolter [6] and thereby demonstrate the effectiveness of the IRST model for both passages and documents. For passages, the "Sum" and "Max" formulations achieve comparable effectiveness, but for documents, the "Sum" formulation appears to be more effective.

### 4.1 Passage Ranking

Experimental results on the MS MARCO passage corpus are shown in Table 1. The BM25 baseline is shown in row (1a), which reports the effectiveness of first-stage retrieval using bag-of-words queries, without any reranking. Here we use the setting of $k_1 = 0.82, b = 0.68$ for BM25. The results from Boytsov and Kolter [6] are shown in row (1b), which is our replication target.

IRST (Sum) and IRST (Max) refer to the two different reranking approaches discussed in Section 3.4; these results are presented in rows (2a) and (2b). On the MS MARCO development queries, MRR@10 is significantly better than the BM25 baseline in row (1a)

| | | MS MARCO dev | TREC 2019 | | TREC 2020 | |
|---|---|---|---|---|---|---|
| | | MRR@100 | nDCG@10 | MAP | nDCG@10 | MAP |
| | | | baselines | | | |
| (1a) | BM25 + Model 1, by Boytsov and Kolter [6] | - | 0.557 | - | 0.525 | - |
| (1b) | Best "traditional" | - | 0.561 | 0.265 | 0.563 | 0.383 |
| (1c) | BM25 w/ RM3 + monoBERT variant | - | 0.640 | 0.323 | - | - |
| | | | doc (full) | | | |
| (2a) | BM25 ($k_1 = 0.82, b = 0.68$) | 0.249 | 0.510 | 0.241 | 0.528 | 0.378 |
| (2b) | BM25 + IRST (Sum) | $0.302^{2a}$ | 0.549 | 0.252 | 0.556 | 0.383 |
| (2c) | BM25 + IRST (Max) | 0.252 | 0.491 | $0.220^{2a}$ | 0.502 | $0.337^{2a}$ |
| | | | doc (segmented) | | | |
| (3a) | BM25 ($k_1 = 0.82, b = 0.68$) | 0.269 | 0.529 | 0.240 | 0.531 | 0.362 |
| (3b) | BM25 + IRST (Sum) | $0.296^{3a}$ | 0.560 | $0.271^{3a}$ | 0.534 | $0.376^{3a}$ |
| (3c) | BM25 + IRST (Max) | $0.259^{3a}$ | 0.520 | 0.243 | 0.509 | 0.350 |

**Table 2: Experimental results on the MS MARCO document corpus for the development queries and TREC topics, with the full document and segmented document conditions shown in separate blocks. Superscripts on the IRST conditions indicate significant improvements over the corresponding BM25 baselines, based on paired $t$-tests ($p < 0.01$).**

for both approaches, based on paired $t$-tests ($p < 0.01$). For TREC 2019 and TREC 2020, the improvements in MAP are statistically significant. It appears that we have successfully replicated the results of Boytsov and Kolter [6]: Our scores are comparable for the TREC 2019 topics and higher for the TREC 2020 topics. Finally, we observe that both IRST formulations, "Sum" and "Max", achieve comparable effectiveness.

For comparison, in row (1c) we report results from the best "traditional" runs (in terms of nDCG@10), taken from TREC 2019 and 2020. By "traditional", the organizers of the tracks mean those runs that did not use any neural methods [10, 11], so these are perhaps better characterized as "pre-neural". In particular, "traditional" runs may have taken advantage of techniques developed after 1999 (e.g., learning-to-rank methods). Unlike with TREC 2019 and 2020, where participants self-categorized their submissions, similar metadata does not exist for runs on the MS MARCO leaderboard. Thus, we are unable to ascertain the best comparable "traditional" run on the MS MARCO development queries. From this comparison, IRST appears to be competitive with the best "traditional" approaches deployed in TREC.

As a final comparison, we report results of monoBERT [22], a transformer-based reranker, taken from Nogueira and Cho [21] and Akkalyoncu Yilmaz et al. [1]. While it is no surprise that transformers vastly outperform our IRST model, more importantly, this comparison allows us to quantify the impact of modeling advances vs. data availability. On the MS MARCO development queries, starting from the BM25 baseline, monoBERT improves effectiveness by 0.177, and IRST (Sum) by 0.033. This means that roughly 20% of the gain can be attributed to simply having more data. In terms of nDCG@10 on TREC 2019, we reach a similar conclusion.

## 4.2 Document Ranking

To examine the robustness and generality of our models, we report ranking experiments on the MS MARCO document corpus. Results are presented in Table 2, where ranking full documents and segmented documents are shown in separate blocks. Row (1b) reports results from Boytsov and Kolter [6], our replication target.

The results here do not appear to be as clear as with the passage retrieval experiments. On TREC 2019 topics, our results are comparable to Boytsov and Kolter [6], comparing rows (2b) and (3b) to row (1a). However, for TREC 2020 topics, our BM25 baselines already beat the results reported by Boytsov and Kolter [6], so it is not clear what conclusions can be drawn. Nevertheless, our gains appear to be more modest.

Examining the doc (full) vs. doc (segmented) conditions, the BM25 baselines shown in rows (2a) and (3a) appear to exhibit comparable effectiveness. With reranking based on IRST (Sum), we observe a statistically significant increase in MRR@100 for the development queries, but at best only modest gains on the TREC topics. However, we do observe statistically significant gains in MAP over the baseline for the IRST (Sum) variant on segmented documents, row (3b). For IRST (Sum), both doc (full) and doc (segmented) conditions perform comparably.

Unlike for passage ranking, it is clear that the "Sum" variant is more effective than the "Max" variant for document ranking. In fact, for the doc (segmented) condition, IRST (Max) is significantly *worse* than the BM25 baseline, shown in row (3c) vs. row (3a), and we observe some significant decreases on TREC topics for doc (full) as well, shown in row (2c) vs. row (2a). We interpret these results as follows: since "Max" only considers the highest matching translation probability, it is likely to be affected by noise, and reranking longer segments of text increases noise.

Just as with the passage retrieval experiments, we compare against the best "traditional" methods from TREC, reported in row (1b). Our results appear to be competitive, which once again highlights the impact of simply having more data. How do our methods compare to reranking with transformers? Row (1c) reports results from team `h2oloo` [1] in the TREC 2019 Deep Learning Track,

which used a model similar to monoBERT to rerank BM25 with RM3 query expansion. However, the addition of RM3 means that the reranker is starting from a higher quality set of candidates, which makes the comparison unfair.

Nevertheless, on the whole it appears that the gains we observe with IRST on document ranking are more modest than the gains for passage ranking. For this we have two explanations: First, we are applying the models learned from passage retrieval in a zero-shot manner, without any customization to document ranking. Second, it is perhaps the case that document ranking is inherently more challenging since we are working with longer (and hence noisier) text. Further explorations of these two points would represent interesting future work.

## 5 CONCLUSIONS

In this paper, we successfully implemented the "information retrieval as statistical translation" model of Berger and Lafferty [5] and replicated the more recent work of Boytsov and Kolter [6]. These results show that conceptual innovations from over two decades ago are truly insightful, but the lack of data precluded empirical validation at the time. However, data availability has finally caught up to the ideas!

Moving forward, we highlight one interesting direction for future work. Any model based on supervised machine learning immediately raises the question of how it generalizes to inputs that are not from the same distribution as the training data. For information retrieval, this relates to questions of domain adaption. Recent work in the context of the BEIR benchmark [27] has shown that learned transformer-based representations may not be effective for retrieval across multiple domains. It would be interesting to explore whether there are similar challenges with IRST as well.

## ACKNOWLEDGEMENTS

This research was supported in part by the Canada First Research Excellence Fund, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Waterloo–Huawei Joint Innovation Laboratory. Computational resources were provided in part by Compute Ontario and Compute Canada.

## REFERENCES

[1] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, and Jimmy Lin. 2019. $H_2$oloo at TREC 2019: Combining Sentence and Document Evidence in the Deep Learning Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference (TREC 2019)*.

[2] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China, 3490–3496.

[3] Nima Asadi and Jimmy Lin. 2013. Effectiveness/Efficiency Tradeoffs for Candidate Generation in Multi-Stage Retrieval Architectures. In *Proceedings of the 36th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2013)*. Dublin, Ireland, 997–1000.

[4] 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. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268v3* (2018).

[5] Adam Berger and John Lafferty. 1999. Information Retrieval as Statistical Translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*. Berkeley, California, 222–229.

[6] Leonid Boytsov and Zico Kolter. 2021. Exploring Classic and Neural Lexical Translation Models for Information Retrieval: Interpretability, Effectiveness, and Efficiency Benefits. In *Proceedings of the 43rd European Conference on Information Retrieval (ECIR 2021), Part I*. 63–78.

[7] Leonid Boytsov and Eric Nyberg. 2020. Flexible retrieval with NMSLIB and FlexNeuART. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*. Online, 32–43.

[8] Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics* 16, 2 (1990), 79–85.

[9] Nick Craswell, Bhaskar Mitra, Daniel Campos, Emine Yilmaz, and Jimmy Lin. 2021. MS MARCO: Benchmarking Ranking Models in the Large-Data Regime. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 1566–1576.

[10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference Proceedings (TREC 2020)*.

[11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2019. Overview of the TREC 2019 Deep Learning Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference Proceedings (TREC 2019)*.

[12] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. Paris, France, 985–988.

[13] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink Training of BERT Rerankers in Multi-Stage Retrieval Pipeline. In *Proceedings of the 43rd European Conference on Information Retrieval (ECIR 2021), Part II*. 280–286.

[14] Sara Hooker. 2020. The Hardware Lottery. *arXiv:2009.06489* (2020).

[15] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020)*. 39–48.

[16] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage Representation Aggregation for Document Reranking. *arXiv:2008.09093* (2020).

[17] Jimmy Lin. 2019. The Neural Hype, Justified! A Recantation. *SIGIR Forum* 53, 2 (2019), 88–93.

[18] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2356–2362.

[19] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Morgan & Claypool Publishers.

[20] Irina Matveeva, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. 2006. High Accuracy Retrieval with Multiple Nested Ranker. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*. Seattle, Washington, 437–444.

[21] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv:1901.04085* (2019).

[22] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. *arXiv:1910.14424* (2019).

[23] Jan Pedersen. 2010. Query Understanding at Bing. In *Industry Track Keynote at the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*. Geneva, Switzerland.

[24] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*. Melbourne, Australia, 275–281.

[25] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *arXiv:2101.05667* (2021).

[26] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R. Hersh. 2020. TREC-COVID: Rationale and Structure of an Information Retrieval Shared Task for COVID-19. *Journal of the American Medical Informatics Association* 27, 9 (2020), 1431–1436.

[27] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *arXiv:2104.08663* (2021).

[28] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.