

Selective Weak Supervision for Neural Information Retrieval

Kaitao Zhang[♡], Chenyan Xiong[♣], Zhenghao Liu[♡], and Zhiyuan Liu[♡]
Tsinghua University[♡], Microsoft Research AI[♣]

{zkt18, liu-zh16}@mails.tsinghua.edu.cn; chenyan.xiong@microsoft.com; liuzy@tsinghua.edu.cn

ABSTRACT

This paper democratizes neural information retrieval to scenarios where large scale relevance training signals are not available. We revisit the classic IR intuition that anchor-document relations approximate query-document relevance and propose a reinforcement weak supervision selection method, ReInfoSelect, which learns to select anchor-document pairs that best weakly supervise the neural ranker (*action*), using the ranking performance on a handful of relevance labels as the *reward*. Iteratively, for a batch of anchor-document pairs, ReInfoSelect back propagates the gradients through the neural ranker, gathers its NDCG reward, and optimizes the data selection network using policy gradients, until the neural ranker’s performance peaks on target relevance metrics (*convergence*). In our experiments on three TREC benchmarks, neural rankers trained by ReInfoSelect, with only publicly available anchor data, significantly outperform feature-based learning to rank methods and match the effectiveness of neural rankers trained with private commercial search logs. Our analyses show that ReInfoSelect effectively selects weak supervision signals based on the stage of the neural ranker training, and intuitively picks anchor-document pairs similar to query-document pairs.

KEYWORDS

Neural IR, Weak Supervision, Pre-Training Data Selection

ACM Reference Format:

Kaitao Zhang[♡], Chenyan Xiong[♣], Zhenghao Liu[♡], and Zhiyuan Liu[♡]. 2020. Selective Weak Supervision for Neural Information Retrieval. In *Proceedings of The Web Conference 2020 (WWW ’20)*, April 20–24, 2020, Taipei, Taiwan, China. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3366423.3380131>

1 INTRODUCTION

Neural information retrieval (Neu-IR) methods learn distributed representations of query and documents and conduct soft-matches between them in the embedding space [4, 5, 13, 25, 33, 45]. In scenarios with sufficient training signals, for example, in commercial search engines with large amounts of user clicks, and on benchmarks with millions of relevance labels, end-to-end Neu-IR methods have significantly improved their ranking accuracy [5, 23, 29, 45].

Without large scale relevance labels, the effectiveness of Neu-IR is more ambivalent [47]. A main challenge is that the language modeling style weak supervision, i.e. word2vec style word co-occurrence [27] and BERT style mask language model [9, 35], does

not provide as effective distributed representations for search relevance modeling [45, 52]. This discrepancy is a significant bottleneck for Neu-IR’s impact in scenarios without the luxury of large amounts of relevance-specific supervision signals, for example, in many academic settings and none-web search domains.

This work addresses the discrepancy between weak supervision methods and the needs of relevance matching, to liberate Neu-IR from the necessity of large scale relevance-specific supervision. Inspired by the classic “Anchor” intuition: *anchor texts are similar to query texts and the anchor-document relations approximate relevance matches between query and documents* [3], we propose ReInfoSelect, “Reinforcement Information retrieval weak supervision Selector”, which conducts selective weak supervision training specifically designed for Neu-IR models. Given a handful of relevance labels in the target ranking task, for example, a TREC benchmark, a large amount of anchor-document pairs, and a Neu-IR model. ReInfoSelect uses REINFORCE [42] to learn to select anchor-document pairs that better optimize the neural ranker’s performance in the target ranking task.

The weak supervision data selection is conducted by a state network that represents the anchor, the document, and the anchor-document relation, and an action network to determine whether to select each pair. This *data selector* is connected to the target *neural ranker* using policy gradients—as the tool to overcome the non-differentiability of the data selection and weakly supervised training process. The learning of the two parties is conducted iteratively in ReInfoSelect’s stochastic process. For a batch of anchor-document pairs, ReInfoSelect 1) selects the weak supervision pairs using its data selector, 2) conducts several back-propagation steps through the neural ranker using the selected pairs, 3) evaluates the neural ranker on the target scenario to collect rewards, and 4) updates the data selector using policy gradients. This process continues through batches of anchor-document pairs until the neural ranker’s performance on the target task converges.

In our experiments on three widely studied TREC benchmarks, ClueWeb09-B, ClueWeb12-B13, and Robust04, ReInfoSelect provides state-of-the-art weak supervision method for two commonly used neural rankers: Conv-KNRM [5] and BERT [29]. Only when guided by ReInfoSelect’s selective weak supervision, these neural rankers robustly outperform feature-based learning to rank methods. ReInfoSelect also matches the effectiveness of Bing User Clicks [4]—The latter is only available in commercial search environments, while everything in ReInfoSelect is publicly available¹.

Our in-depth studies demonstrate the raw anchor-document pairs are too noisy and may hurt neural rankers’ accuracy when used directly [8]. In comparison, ReInfoSelect intuitively selects weak supervision signals based on the status of the trained neural ranker: it is lenient to noisy anchors when the neural ranker is just initialized, but, as the model converging, quickly becomes selective

¹All our codes, data, and results are available at <https://github.com/thunlp/ReInfoSelect>.

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW ’20, April 20–24, 2020, Taipei, Taiwan, China

© 2020 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-7023-3/20/04.

<https://doi.org/10.1145/3366423.3380131>

and only picks weak supervision signals that can further elevate the ranker’s effectiveness. This is also revealed in our human evaluation: earlier in the training process, ReInfoSelect selects 80%+ anchor-document pairs, most not that similar to query-relevant documents; later—as the neural ranker becomes better—ReInfoSelect becomes more selective and picks anchor-document pairs rated as better approximations for query-relevant document pairs.

The next section discusses related work. Section 3 presents the ReInfoSelect framework. Section 4 and Section 5 describe our experiments and results. Section 6 concludes.

2 RELATED WORK

Neu-IR models can be categorized as representation based and interaction based [13]. Representation based methods encode the query and the document separately as two distributed representations, for example, using feed-forward neural networks [19, 40], convolutional networks [18], or transformers [22], and match them using encoding distances. The encoding-and-match nature makes them more efficient for retrieval [1].

Interaction based methods model the fine-grained interactions between query and documents, often by the translation matrices between all query and document term pairs [2], which in Neu-IR are calculated using term embeddings [18]. The ranking scores are then calculated using the translation matrices, for example, by convolutional neural networks [18, 32], recurrent neural networks [33], density estimation kernels [45], and position-aware networks [20]. BERT-based rankers, with the strong interactions between query and document terms in the transformer’s multi-head attentions, are also interaction based [37].

When large scale relevance training signals are available, the interaction based Neu-IR methods have shown strong effectiveness over previous feature-based learning to rank methods. Conv-KNRM, which captures soft matches between query-document n-grams using kernels [5], significantly outperforms feature-based methods in the Chinese Sogou-T search log [23, 53, 54], the Bing click log [5], and MS MARCO which includes millions of expert relevance label [17, 37]. The Bing log pre-trained Conv-KNRM can also generalize to ClueWeb benchmarks [4, 5].

Nevertheless, previous research finds that much of Neu-IR’s effectiveness relies on embeddings specifically trained for relevance, because the word co-occurrence trained word embeddings do not align well with the needs of ad hoc retrieval [45, 48]. This necessitates large scale relevance training data for Neu-IR methods’ effectiveness. In reality, large scale relevance training data either require search logs from commercial search engines or expensive human labeling; this luxury is not often available, i.e. in academic settings or in none-web search domains where neither a large amount of human labels nor large search traffic exists. This significantly limits the impact of Neu-IR: in special domain search and TREC benchmarks, ambivalent performances have been observed from many Neu-IR methods [47].

The rise of large pre-trained transformers, e.g., BERT [9], has significantly influenced the landscape of Neu-IR. By concatenating the query and document, as a sequence to sequence pair, and fine-tuning on relevance labels, BERT based ranker outperforms previous (shallow) neural ranking models by large margins on

the MS MARCO passage ranking task [29, 30]. The strong effectiveness, especially considering BERT is also pre-trained on word co-occurrence signals, has raised many investigations of its source of effectiveness in ranking [4, 31, 37]. Though definitive conclusions remain to be studied, recent research still observed a significant defect of BERT’s weak supervision in relevance matching: Dai and Callan show that when BERT is further fine-tuned on Bing user clicks, its accuracy on TREC Web Tracks improved by 16% compared to only using Mask-LM pretraining [4]. This indicates that there is still a significant gap between the weak supervision of BERT’s pretraining and the needs of search relevance matching.

There are several attempts to train Neu-IR models using weak supervision [7, 24, 49, 50]. One explored weak supervision signal is Pseudo Relevance Feedback [3]. The top retrieved documents have been used to train individual word2vec for each query and then used for query expansion [10]. The unsupervised retrieval scores, e.g., from BM25, have been used as relevance labels to train neural ranking models [8]. The PRF weak supervision signals in general can promote neural models to similar effectiveness as PRF-based query expansion methods [8, 10, 47]. The other explored weak supervision signal is the title-body relation in web documents, where MacAvaney et al. treat titles as an approximation of queries and build a discriminator to find most proper titles for weak supervision [26].

Recently, learning to select higher quality weak supervision data has received much attention in deep learning, partly because of the “data hungry” property of deep neural networks [15, 16, 39]. A relatively new technique, it has not yet been utilized in ad hoc retrieval nor weak supervision settings; recent studies were mainly in the domain adaptation setting of natural language processing tasks, i.e., paraphrase identification and natural language inference [38, 41].

Anchor texts have been used in various IR tasks [12]: query refinement [21], query suggestion [6], and document expansion [11], to name a few. Previous usage of anchor texts in Neu-IR is mainly as an additional document field to provide additional ranking signals [50], not as weak supervision. The application of reinforcement learning in IR is also mainly on ranking models [43, 51].

3 METHODOLOGY

This section first describes some preliminaries of neural rankers and then our ReInfoSelect method.

3.1 Preliminary

Given a query q and document d , neural ranking models calculate ranking score $f(q, d)$ using query words $q = \{t_1^q, \dots, t_i^q, \dots, t_m^q\}$ and document words $d = \{t_1^d, \dots, t_j^d, \dots, t_n^d\}$. There are many neural architectures developed for $f(q, d)$. In this work, we choose Conv-KNRM [5], as the representative of neural rankers with shallow word embeddings, and BERT Ranker [29], as the representative of pre-trained transformers.

Conv-KNRM matches the query and document in the n-gram embedding space using matching kernels [45]. It first uses Convolutional Neural Network (CNN) to calculate h -gram embeddings \vec{g}_i^h from word embedding \vec{t} [5]:

$$\vec{g}_i^h = \text{CNN}^h(\vec{t}_{i:i+h}). \quad (1)$$

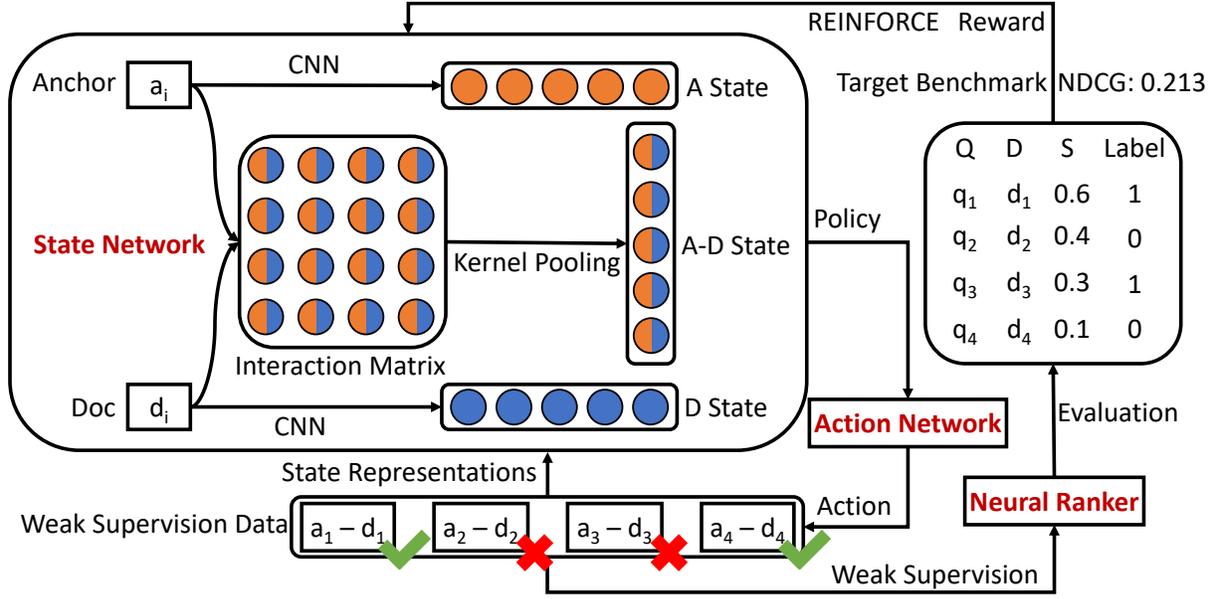


Figure 1: The Architecture of ReInfoSelect.

The h_q -gram and h_d -gram embeddings of query and document are used to construct a translation matrix M^{h_q, h_d} , whose item is the cosine similarity of corresponding h -gram pairs:

$$M_{ij}^{h_q, h_d} = \cos(\vec{g}_i^{h_q}, \vec{g}_j^{h_d}). \quad (2)$$

It then uses K Gaussian kernels to extract the matching feature $\phi(M^{h_q, h_d}) = \{K_1(M^{h_q, h_d}), \dots, K_K(M^{h_q, h_d})\}$ from M^{h_q, h_d} . Each kernel K_k summarizes the translation scores as soft-TF counts:

$$K_k(M^{h_q, h_d}) = \sum_i (\log \sum_j \exp(-\frac{(M_{ij}^{h_q, h_d} - \mu_k)^2}{2\delta_k^2})), \quad (3)$$

where μ_k and δ_k are the mean and width for the k -th kernel. The h -gram soft match kernels are concatenated to the final features:

$$\Phi(M) = \phi(M^{1,1}) \dots \phi(M^{h_q, h_d}) \dots \phi(M^{h_{\max}, h_{d_{\max}}}), \quad (4)$$

where \circ is the concatenate operation.

The soft-TF features are combined by a standard ranking layer:

$$f(q, d) = \tanh(\omega_r \cdot \Phi(M) + b_r), \quad (5)$$

with parameters ω_r and b_r and tanh activation.

BERT is a pre-trained deep transformer and performs well on many text related tasks [9]. To leverage the pre-trained BERT’s sequence to sequence modeling capability, BERT first concatenates the query and document into one text sequence and feed it into pre-trained BERT [29]:

$$\text{BERT}(q, d) = \text{Transformer}([\text{CLS}] \circ q \circ [\text{SEP}] \circ d \circ [\text{SEP}]). \quad (6)$$

The last layer’s “[CLS]” token representation is used as the “matching” feature $\text{BERT}(q, d)$. Then a ranking layer combines the representation to the ranking score:

$$f(q, d) = \tanh((\omega_r \cdot \text{BERT}(q, d) + b_r), \quad (7)$$

where ω_r and b_r are the learning to rank parameters.

Training. Conv-KNRM and BERT often require large scale relevance supervision [14, 47]. The main capacity of Conv-KNRM is its relevance specific n-gram embeddings, which need to be trained by a large amount of relevance labels, for example, user clicks in search logs [4, 5] or human labels from MS MARCO (1 Million Labels) [37]. BERT is already pre-trained by the Mask-LM task [9]. Still, its advantage in ranking is more observed when fine-tuned by a large amount of supervision signals from user clicks or MS MARCO [4, 29, 30, 47].

3.2 Reinforcement Data Selection

ReInfoSelect overcomes the dependency of a large amount of relevance labels by weakly supervising neural rankers with the widely available anchor data. The challenge is, the anchor data are inevitably noisy: Many anchors are “functional” rather than “informational”, for example, “homepage” and “contact us”; some anchors are too general and may not even retrieve its linked document, e.g., “customer support”. Directly using all anchor-document pairs to train Neu-IR models is unlikely effective.

To address this challenge, ReInfoSelect learns to select more suitable anchor-document pairs directly by their ability to optimize neural rankers. As shown in Figure 1, this is achieved by several components: the *State* network that represents the a-d pair, the *Action* network which decides whether to keep the pair, and the training *Reward* gathered from the trained ranker.

Specifically, for the i -th weak supervision pair $b_i = (a_i, d_i)$, ReInfoSelect decides if the pair should be used as a weak supervision signal (Action_i) by using the state representations s_i of the a-d pair. The selected weak supervision pairs are then used to train the neural ranker to obtain the reward R , which goes back to train the action and state networks. The rest of this section describes the *State*, *Action*, *Reward*, and the learning via *Policy Gradient*.

State. The state represents whether an anchor-document pair is good training data for neural rankers. The state s_i for the i -th pair include three continuous vectors: anchor state s_i^a , document state s_i^d , and anchor-document interaction state s_i^{ad} .

The anchor state and document state representations use standard convolutional neural networks on their word embeddings:

$$s_i^a = \text{CNN}_a(a_i); \quad (8)$$

$$s_i^d = \text{CNN}_d(d_i). \quad (9)$$

The anchor-document state representation uses the ranking feature $\Phi(\mathcal{M}_{a_i, d_i})$ from Conv-KNRM (Eqn. 4):

$$s_i^{ad} = \Phi(\mathcal{M}_{a_i, d_i}). \quad (10)$$

The three vectors are concatenated to the final state:

$$s_i = s_i^a \circ s_i^d \circ s_i^{ad}. \quad (11)$$

Note that the parameters are not shared with the neural ranker.

Action. The action decides whether to use the anchor-document pair (1) or not (0) as a weak supervision signal. The action on the i -th a-d pair is calculated as

$$\text{Action}_i = \text{argmax}_{0,1} \pi(s_i), \quad (12)$$

$$\pi(s_i) = \text{softmax}(\text{Linear}(s_i)), \quad (13)$$

a simple linear layer on the state to predict the action probability.

Reward. The state and action networks are trained using the ultimate goal of ReInfoSelect: the accuracy of the neural ranker when trained by the selected pairs, as the reward.

Let $\hat{B}^t = \{\hat{b}_1^t, \dots, \hat{b}_i^t, \dots, \hat{b}_{k^*}^t\}$ be a set of selected anchor-document pairs in the t -th batch B^t , the reward of this selected batch is:

$$r^t = \text{NDCG}(f^{\hat{B}^t}(q, d)) - \text{NDCG}(f^{\hat{B}^{t-1}}(q, d)), \quad (14)$$

where $f^{\hat{B}^t}$ and $f^{\hat{B}^{t-1}}$ are the neural ranker, e.g., Conv-KNRM or BERT, trained using the weak supervision pairs selected from the (1:t) and (1:t-1) batches. NDCG evaluates the neural ranker's accuracy on the *validation* part of the target ranking benchmarks. The reward is the NDCG change of the neural ranker when trained with additional weak supervision signals from \hat{B}^t .

Policy Gradient. The NDCG metric and the discrete action are not differentiable. We use the standard policy gradient and REINFORCE to “propagate” the reward to the training of the state and action networks [42].

At the T -th reinforcement step, REINFORCE first calculates the accumulate reward R^t for each of the t -th step:

$$R^t = \sum_{j=t}^T c^j r^j. \quad (15)$$

It first calculates the accumulated influence of the action taken at the t -th step in the future batches ($t:T$). The influence is discounted by the hyperparameter c^t . This leads to the expected reward \bar{R} for the T -th step:

$$\bar{R} = \frac{1}{T} \sum_{t=1}^T R^t. \quad (16)$$

The expectation reward is then used to optimize the parameters of the state and action networks using standard policy gradient [42]:

$$\theta_{\text{state, action}}^* \leftarrow \theta_{\text{state, action}} + \alpha \sum_T \sum_{i=1}^k \bar{R} \nabla_{\theta} \log \pi_{\theta}(s_i), \quad (17)$$

where α is the learning rate and k is the total number of weak supervision pairs of batch B^t . The expectation reward guides the updates of the parameters θ of the data selection policy $\pi(s_i)$.

ReInfoSelect uses policy gradients as the tool to connect its action and state networks to the neural ranker's performance, and thus learns to select more suitable weak supervision signals from the anchor-document pairs.

3.3 Neural Ranker Training with ReInfoSelect

As shown in Figure 1, ReInfoSelect learns interactively with the neural ranker. The two stochastically go through batches of anchor-document pairs. In each batch, ReInfoSelect first selects the anchor-document pairs using its state and action networks, and then stochastically trains the neural ranker using the selected pairs. After that, the updated neural ranker is evaluated on the validation query-document relevance labels to obtain reward, which updates ReInfoSelect's policy networks via policy gradient.

Specifically, for the batch B , the neural ranker f is trained using standard pairwise learning to rank:

$$l = \sum_{a_i} \sum_{d_i^+, d_i^-} \max(0, 1 - f(a_i, d_i^+) + f(a_i, d_i^-)), \quad (18)$$

where a_i, d_i^+ is the anchor-document pair selected by ReInfoSelect. The document d_i^+ linked by a_i approximates the relevant document for the pseudo query a_i . The negative document d_i^- is from the documents retrieved by a base retrieval model, i.e., BM25, using a_i as the query, following standards in learning to rank [3].

The neural ranker is updated per batch B as it is used to provide reward for the actions taken per batch. ReInfoSelect's state and action networks are updated per T batches (an episode in REINFORCE) to capture the action's delayed influences. The two circle through the entire anchor-document pairs and stochastically update their parameters until ranking performance converges.

4 EXPERIMENTAL METHODOLOGY

This section describes the weak supervision dataset, evaluation datasets, baselines, training and implementation details.

Weak Supervision Dataset. The weak supervision dataset is constructed from English corpora of ClueWeb09, which consists of 504 million web pages. The anchor texts and their linked web pages are regarded as pseudo queries and potential documents for weak supervision. All web pages are parsed by “KeepEverythingExtractor” in Boilerpipe. Anchor texts are collected using warc-clueweb².

About 100K anchors (from total 6 million collected) and their linked documents are randomly sampled as the weak supervision dataset. Pseudo negative documents are the top retrieved ones by BM25 in Elastic Search³. The influences of number of anchors and ratio of pseudo positive and negative documents are studied in Section 5.5.

²<https://github.com/cdegroc/warc-clueweb>

³<https://www.elastic.co/cn/downloads/elasticsearch>

Evaluation Datasets. Three ad hoc retrieval benchmarks are used in evaluation: ClueWeb09-B, Robust04, and ClueWeb12-B13. ClueWeb09-B consists of 200 queries with relevance labels from TREC Web Track 2009-2012. Robust04 consists of 249 queries with relevance labels. ClueWeb12-B13 includes 100 queries from TREC Web Track 2013-2014. Title queries are used.

On ClueWeb09-B and Robust04, we use the exact same re-ranking setup with Dai and Callan [4], which presents the state-of-the-art neural ranking accuracy. All our ranking models re-ranked their released top 100 SDM retrieved results [4]. We use the same title concatenated with the first paragraph as the document representation to fit in BERT’s max sequence length [4]. On ClueWeb12-B13, we follow the re-ranking setup from Xiong et al. [44], as Dai and Callan [4] does not include ClueWeb12.

All experiment settings are kept consistent with Dai and Callan [4] (Xiong et al. [44] on ClueWeb12). All the candidate documents to rerank are from their base retrieval methods. Conv-KNRM and BERT Ranker use open source implementations [5, 37]. The evaluation scores with the corresponding paper are thus directly comparable. This is crucial for reproducible IR and to compare ReInfoSelect with the supervision from private Bing search log [4].

TREC official metrics, NDCG@20 and ERR@20, are used. Statistic significance is tested by permutation test with $p < 0.05$.

Baselines. Our main baselines are other training methodologies in Neu-IR. We also compare with standard ranking baselines.

No weak Supervision uses no additional ranking labels beyond the existing small scale relevance labels in the evaluation datasets. This is the vanilla baseline. Conv-KNRM’s word embeddings are initialized by Glove [34]; BERT uses Google’s released pre-trained parameters [9].

Anchor+BM25 Labels is our implementation of BM25 weak supervision [8]. We use the same anchors in our weak supervision dataset as pseudo queries, their BM25 retrieved documents as documents, and the BM25 scores as the weak supervision labels. The difference is that we use Conv-KNRM and BERT, stronger ranking models than their feedforward neural networks [8].

Title Discriminator is our implementation of MacAvaney et al. [26]. We use the title-document relation in ClueWeb09-B as the pseudo label and Conv-KNRM as their discriminator. It differs from ReInfoSelect that the discriminator is not trained by reward from neural ranker’s NDCG. Instead, it is a classifier trained to classify query-document pairs from title-document pairs; then the title-document pairs most similar to query-document are used.

All Anchor uses all randomly sampled anchor-document pairs without any filtering or weighting.

MS MARCO Human Label uses the passage ranking labels from MS MARCO as relevance supervision [28]. It includes human labels for one million Bing queries.

Bing User Clicks is the results from Dai and Callan [4], where they used user clicks in Bing as the supervision signal, which includes 5M query-document pairs. As the commercial search log is not publicly available, we use their reported numbers, which are directly comparable as our experimental setting are kept consistent.

All these training methods are applied to Conv-KNRM and BERT using the exact same setup except different training strategies. All neural rankers are adapted to the target ranking benchmark (ClueWeb and Robust) the same with previous research [4, 5]. They

are first trained with the weak supervision or source domain supervision signals; then their ranking features (kernels or [CLS] embeddings) are combined with the base retrieval (SDM) score using Coordinate-Ascent, through standard five-fold cross validation on the target benchmark.

Standard ranking baselines include SDM and two learning to rank methods, RankSVM and Coor-Ascent, with standard IR features. We found that the NDCG scores on ClueWeb09-B and Robust04 in Dai and Callan [4] are much higher than our implementations and previous research [5, 45]. We choose to compare with their stronger baselines, though not all metrics were provided. ClueWeb12-B13 baselines are those released by Xiong et al. [44].

Training Details of ReInfoSelect. There are three steps for training with the ReInfoSelect: warm up, reinforce training with anchor data, and adapting to the ranking benchmark. For all steps, the ranking benchmark labels (ClueWeb09-B or Robust04) are partitioned to five folds for cross validation.

The warm up stage first trains the state and action network using the discriminator setup [26]. Then in the reinforce stage, ReInfoSelect’s networks are initialized (warmed up) by the learned discriminator weights. The reinforce and adaptation are all conducted via five-fold cross validation. In each of the five runs, ReInfoSelect only uses the four training folds to calculate the rewards and to train the neural rankers. No testing information is used in any of its training stages. Then the same training splits are used to fine-tune the weakly supervised neural rankers, the same with baselines. The testing fold is only used for final evaluation.

Implementation Details. This part describes the implementation details of ReInfoSelect and all baselines.

Standard Indri stopword removal and KrovetzStemmer are used to process queries and documents for Conv-KNRM. The BERT based models use BERT’s sub-word tokenizer.

Conv-KNRM uses 21 kernels, one exact match and the rest soft match [37]; the uni-gram, bi-gram, and tri-gram of query and document texts are considered, the same as the previous work [4, 5]. The word embedding dimension is 300 and initialed with Glove [34]; the learning rate is $1e - 3$. The BERT models inherit pytorch-transformers⁴. The max sequence length is 384. Adam with learning rate = $5e - 5$ and warm up proportion 0.1 is used.

For the data selector of ReInfoSelect, the discount factor c^t is 0.99. State networks use their own 300-dimensional word embeddings initialized with Glove [34]. Their CNNs use window sizes 3, 4, and 5. Its learning rate is $1e - 3$.

The neural ranking models are updated with one gradient step per batch, while the data selector is updated once every 4 batches ($T=4$). All our neural models are implemented with PyTorch. All models are trained with a single GeForce GTX TITAN GPU and trained about 40 hours for one epoch. More details of our implementation can be found in our code repository⁵.

5 EVALUATION RESULTS

Six experiments are conducted to evaluate ReInfoSelect’s effectiveness. We also provide human evaluations and case studies on the selected weak supervision data.

⁴<https://github.com/huggingface/pytorch-transformers>

⁵<https://github.com/thunlp/ReInfoSelect>

Table 1: Ranking results of ReInfoSelect and baselines. †, ‡, §, ¶, * indicate statistically significant improvements over No Weak Supervision[†], Anchor+BM25 Labels[‡], Title Discriminator[§], All Anchor[¶], and MS MARCO Human Label^{*}. None-neural baselines on ClueWeb09-B and Robust04 are from Dai and Callan [4] and those on ClueWeb12-B13 are from Xiong et al. [44].

Method	ClueWeb09-B		Robust04		ClueWeb12-B13	
	NDCG@20	ERR@20	NDCG@20	ERR@20	NDCG@20	ERR@20
SDM (From [4] [44])	0.2774	0.1380	0.4272	0.1172	0.1083	0.0905
RankSVM (From [4] [44])	0.289	n.a.	0.420	n.a.	0.1205	0.0924
Coor-Ascent (From [4] [44])	0.295	n.a.	0.427	n.a.	0.1206	0.0947
Conv-KNRM as the Neural Ranker						
No Weak Supervision (From [4])	0.270	n.a.	0.416	n.a.	n.a.	n.a.
No Weak Supervision (Ours)	0.2873	0.1597	0.4267	0.1168	0.1123	0.0915
Anchor+BM25 Labels [8]	0.2910	0.1585	0.4322	0.1179	0.1181	0.0978
Title Discriminator [26]	0.2927	0.1606	0.4318	0.1193	0.1176	0.0975
All Anchor	0.2839	0.1464	0.4305	0.1190	0.1119	0.0906
MS MARCO Human Label	0.2903	0.1542	0.4337	0.1194	0.1183	0.0981
Bing User Clicks (From [4])	0.314	n.a.	n.a.	n.a.	n.a.	n.a.
ReInfoSelect	0.3094 ^{†‡§¶*}	0.1611 [¶]	0.4423 ^{†‡§¶*}	0.1202 [†]	0.1225 ^{†¶}	0.1044 ^{†‡§¶*}
BERT as the Neural Ranker						
No Weak Supervision (From [4])	0.286	n.a.	0.444	n.a.	n.a.	n.a.
No Weak Supervision (Ours)	0.2999	0.1631	0.4258	0.1163	0.1190	0.0963
Anchor+BM25 Labels [8]	0.3068	0.1618	0.4375 [†]	0.1233 [†]	0.1160	0.0990
Title Discriminator [26]	0.3021	0.1513	0.4379 [†]	0.1202 [†]	0.1162	0.0981
All Anchor	0.3072	0.1609	0.4446 [†]	0.1206 [†]	0.1208	0.0965
MS MARCO Human Label	0.3085	0.1652	0.4415 [†]	0.1213 [†]	0.1207	0.1024
Bing User Clicks (From [4])	0.333	n.a.	n.a.	n.a.	n.a.	n.a.
ReInfoSelect	0.3261 ^{†‡§¶*}	0.1669	0.4500 ^{†‡§¶*}	0.1220 [†]	0.1276 ^{†‡§}	0.0998

5.1 Overall Results

The overall ranking results are presented in Table 1. Note that though our implementation of No Weak Supervision performs slightly better than Dai and Callan’s [4], with only the several hundreds of labeled queries, neither Conv-KNRM nor BERT convincingly outperforms the classic feature-based learning to rank methods, RankSVM or Coor-Ascent. Similar ambivalent effectiveness on Neu-IR systems when relevance training data are limited has been observed in multiple previous studies [4, 5, 46, 47].

With both neural ranking models, ReInfoSelect outperforms all baselines except Bing User Clicks on both datasets. The improvements on NDCG are robust across the table, while the ERR metric is a little more brittle, especially on ClueWeb12-B13, the same as observed by previous research [5, 25, 46]. ReInfoSelect and Bing User Clicks are the only two methods that show stable improvements over the feature-based Coor-Ascent. Note again that Bing User Clicks are not publicly available, while ReInfoSelect only uses widely available anchor information.

Adapting from MS MARCO Human Label does not lead to much improvement, though using one million expert relevance labels. The MS MARCO ranking task is a passage ranking for more natural language queries. The domain differences limit the generalization ability of human relevance labels. Weak supervision and transfer learning sometimes are necessary: One million human labels are not time and cost effective, if ever feasible, in many ranking scenarios.

ReInfoSelect significantly outperforms All Anchor on both datasets. The latter uses the same anchor-document relation as weak

supervision signals but without any selection. Random anchors are noisy and not always similar to search queries. Section 5.7 further studies the quality of selected anchors.

Our implementation of Anchor+BM25 Labels uses better neural rankers and also combines it with base retrieval, compared to its vanilla form in previous research [8]. Still, it does not yet outperform No Weak Supervision. BM25 scores can be used as pseudo relevance feedback (PRF) or to find stronger negative documents [3, 26]. More in line with the later, ReInfoSelect uses BM25 to find negative documents for anchors.

There are two differences between Title Discriminator [26] and ReInfoSelect. The first is the weak supervision signals: Title VS Anchor. The second is that ReInfoSelect learns to select weak supervision pairs using reward from the neural ranker, while Title Discriminator conducts the selection of training data and the training of neural rankers independently [26]. Section 5.2 and Section 5.3 further study the effectiveness of the data selection.

We have also experimented with other neural rankers, including EDRM, which integrates external knowledge [23], and Transformer-Kernel, which uses sub-word and transformer with kernels⁶. Similar effectiveness and trends were observed. We also explored the ensemble of the ReInfoSelect supervised neural rankers; similar gains from previous research are observed [36] and the ensemble models outperform the single model trained by Bing user Clicks, though it is not a fair comparison. These additional results are listed in our open source repository due to space limitations.

⁶<https://github.com/sebastian-hofstaetter/transformer-kernel-ranking>

Table 2: Classification Accuracy of different state networks when used as the data discriminator [25]: Anchor only, Anchor-Document pair only, and All state together.

Method	ClueWeb09-B		Robust04		ClueWeb12-B13	
Discriminator	A	A-D	A	A-D	A	A-D
A State	0.775	–	0.850	–	0.786	–
A-D State	0.723	0.702	0.890	0.928	0.726	0.712
All State	0.743	0.728	0.863	0.899	0.740	0.732

5.2 State Networks As Discriminators

This experiment studies the effectiveness of the state representations, using an intermediate evaluation—how well they can classify the actual query-documents from anchor-documents pairs, i.e., as the discriminator role in Title Discriminator [26].

We use the state network and the action network to directly learn a binary classifier, using the query-documents from ClueWeb09-B, Robust04 or ClueWeb12-B13 as positive instances and the anchor-documents as negative instances. We train the model the same as we train the Title Discriminator and evaluate their classification accuracy in five-fold cross-validation. The results are shown in Table 2.

Overall, the ClueWeb queries and documents are harder to distinguish from the anchor and documents; the accuracy on ClueWeb in general is lower than on Robust. This is expected as the weak supervision documents are also from ClueWeb and the ClueWeb queries are in the web domain. There is less domain difference between our weak supervision and ClueWeb’s labels. This also correlates with the greater relative improvements of ReInfoSelect on ClueWeb compared to Robust.

All three state networks have decent accuracy on both datasets, showing their good representation ability of the query/anchor-document pairs. The Anchor State easily distinguishes anchors from queries, and the Anchor-Document states find query-document pairs from anchor-document pairs, especially on Robust. However, these intermediate results only show whether the state networks represent the data well. It is unclear, especially when used in the discriminator setup [26], whether a highly accurate classifier will lead to better trained neural models. A perfect discriminator discards all anchor-document data, which is not useful as it leaves no information to use for weak supervision. We care the most whether these state networks help us pick weak supervision pairs that lead to more effective neural rankers, which is studied in the next experiment.

5.3 Effectiveness of ReInfoSelect Selection

Table 3 shows the effectiveness of ReInfoSelect’s data selection with different strategies. Three different policy strategies are experimented. The first is the two-step approach, Discriminator [26]. It selects anchor-document pairs that like real query-document pairs and then train the neural ranker with selected data. The next two are from ReInfoSelect: Scratch initializes its state and action network from scratch, and Warm Up initializes its state and action network from the results of the corresponding Discriminator. Three state combinations are evaluated with the three strategies: A(nchor) State only, A(nchor)-D(ocument) State only, and All State.

Table 3: Conv-KNRM Results on ClueWeb09-B in different classifiers or ReInfoSelect states. Relative performances % and statistical significance[†] are compared with the corresponding Discriminator using the same states.

Method	NDCG@20		ERR@20	
All Anchor	0.2839	–	0.1464	–
A Discriminator	0.2893	–	0.1521	–
A State (Scratch)	0.3002 [†]	+3.77%	0.1604	+5.46%
A State (Warm Up)	0.3050 [†]	+5.43%	0.1632	+7.30%
A-D Discriminator	0.2974	–	0.1556	–
A-D State (Scratch)	0.3033	+1.98%	0.1653	+6.23%
A-D State (Warm Up)	0.3083 [†]	+3.66%	0.1646	+5.78%
All Discriminator	0.3021	–	0.1576	–
All State (Scratch)	0.3078	+1.89%	0.1670	+5.96%
All State (Warm Up)	0.3094 [†]	+2.42%	0.1611	+2.22%

All data selection methods outperform the ALL Anchor, which illustrates the anchor data is informative but noisy; data selection is necessary to filter out noisy anchor data. The reinforce based data selection methods outperform all Discriminator models, demonstrating the effectiveness of ReInfoSelect. Connecting the performances of weakly supervised neural ranker to data selector using the policy network provides significant accuracy boosts.

All state shows better performance than other state representations. It helps ReInfoSelect select better anchor-document pairs for weak supervision, though it might not be the best classifier to distinguish anchor from the query when used as a data discriminator. We also find Warming Up the state and action networks slightly more effective than training from Scratch. We observe that the REINFORCE is slow and unstable in training and a better initial state may improve its stability. This is further studied in Section 5.6.

5.4 Fine-Tuning Strategies

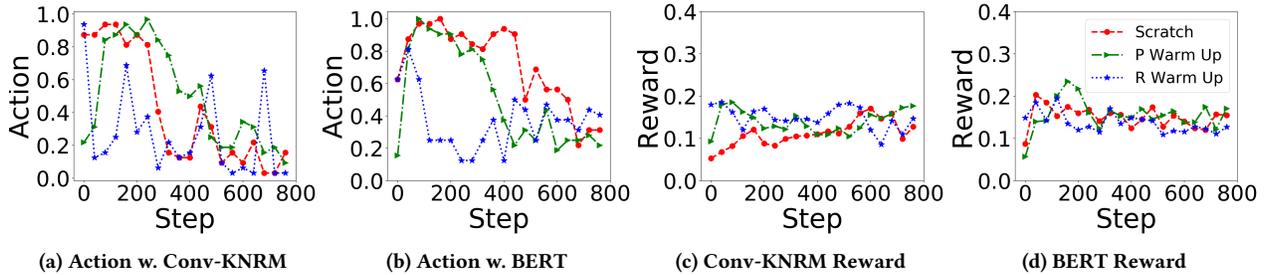
This experiment studies different fine-tuning strategies when adapting the weakly supervised neural rankers to target ranking scenarios. We focus on Conv-KNRM as the ranker and ClueWeb09-B as the target ranking scenario, and experiment with several different adaptation approaches. The first directly fine-tunes the Dense Layer of Conv-KNRM on ClueWeb09-B, with the embeddings frozen (which we find more effective). The second feeds the kernel scores (soft match features) to Coor-Ascent instead of a Dense layer. The last is the one used in previous research and this work, which combines the neural ranker’s features with SDM scores in the standard learning to rank set up [4, 5]. We also combine Conv-KNRM and BERT with SDM, following previous research [25].

Adding SDM score significantly improves the ranking accuracy. Even Conv-KNRM has the soft n-gram match function. The core IR intuitions, e.g., proximity, smoothing, and normalization, in SDM are still necessary for neural rankers to perform well. The current Neu-IR models enhance classic IR approaches but have not yet replaced them. Coor-Ascent is also much more effective learning to rank model compared to the simple Dense (Linear) Layer. The listwise ranker is as effective in combining neural features as in combining classic features.

Combining Conv-KNRM and BERT also provides further improvements over either one individually [25]. We also observed better

Table 4: Conv-KNRM Results on ClueWeb09-B in different fine-tuning strategies. Percentages indicate relative performance over No Weak Supervision.

Method	Feature	LeToR	NDCG@20		ERR@20	
No Weak Supervision (Ours)	Conv-KNRM+SDM	Coor-Ascent	0.2873	–	0.1597	–
Discriminator	Conv-KNRM	Dense Layer	0.2523	–12.18%	0.1344	–15.84%
	Conv-KNRM	Coor-Ascent	0.2787	–2.99%	0.1429	–10.52%
	Conv-KNRM+SDM	Coor-Ascent	0.2980	+3.72%	0.1592	–0.31%
	Conv-KNRM+BERT+SDM	Coor-Ascent	0.3170	+10.34%	0.1747	+9.39%
ReInfoSelect	Conv-KNRM	Dense Layer	0.2694	–6.23%	0.1523	–4.63%
	Conv-KNRM	Coor-Ascent	0.2896	+0.80%	0.1615	+1.13%
	Conv-KNRM+SDM	Coor-Ascent	0.3094	+7.69%	0.1611	+0.88%
	Conv-KNRM+BERT+SDM	Coor-Ascent	0.3222	+12.15%	0.1796	+12.46%

**Figure 2: The behavior of ReInfoSelect’s on ClueWeb09-B when trained from Scratch, with its policy networks warmed up (P Warm Up), and the neural ranker warmed up (R Warm Up), both from All Discriminator. X-axes show the training step (batches) before convergence; Y-axes show the fraction of pairs being selected (action) and the rewards from neural rankers.****Table 5: Data Strategies Results of ReInfoSelect on ClueWeb09-B, with different number of anchors (#a), different number of positive/negative documents per anchor (#d+/a and #d-/a), and different number of total anchor-document pairs (#pair). Percentages indicate relative performance compared with 10K anchors.**

#a	#d ⁺ /a	#d ⁻ /a	#pair	NDCG@20		ERR@20	
10K	35.4	35.5	0.7M	0.3025	–	0.1611	–
50K	7.2	7.2	0.7M	0.3084	+1.95%	0.1651	+2.48%
100K	5.5	5.5	1.1M	0.3094	+2.28%	0.1611	+0.00%
100K	3.6	7.1	1.1M	0.3073	+1.59%	0.1645	+2.11%
100K	2.0	8.0	1.0M	0.3042	+0.56%	0.1621	+0.62%
500K	1.2	1.2	1.2M	0.3085	+1.98%	0.1664	+3.29%

performances of ReInfoSelect over Discriminator across all settings, showing the robust effectiveness of ReInfoSelect in selecting more effective weak supervision signals.

5.5 Influence of Training Data Strategies

Recent advancements of Neu-IR models are mainly trained on data with a vast amount of queries but only a handful of document labels per query, e.g. search log clicks [4, 5] and MS MARCO labels [30]. Both have less than five relevance documents per query, much fewer than typical TREC benchmarks.

This experiment studies the influence of different data combinations in training neural ranking models. Specifically, we keep

the number of total anchor-document pairs roughly the same and vary the different combinations of the number of anchors (#a) and documents per anchor (#d/a). We also try some different balances of positive VS negative documents. The results of ReInfoSelect trained Conv-KNRM on ClueWeb09-B are shown in Table 5.

The ranking accuracy does vary, to some degree, with different data combinations, although using similar amounts of training labels. We observe that Conv-KNRM prefers more variations on the query side. It performs better when with more anchors but fewer documents per anchor. This correlates with the setup in MS MARCO and in Sogou-QCL [28, 53]. We also observe that the neural ranker prefers more positive labels per query. It is expected as the positive labels provide more information than negative ones. Nonetheless, in this experiment, ReInfoSelect is not very sensitive and its accuracy does not vary much across different data strategies.

5.6 Stochastic Training Analysis

This experiment analyzes the stochastic training of ReInfoSelect. We plot the action (fraction of selected pairs) and the reward of each training batch in Figure 2. Three variations of ReInfoSelect are evaluated: with all parameters in the policy network and the neural rankers from Scratch (BERT ranker is initialized from pretraining without IR continuous training), with the policy network parameters warmed up using parameters from All Discriminator, and with the neural rankers warmed up using their correspondence from All Discriminator.

Table 6: Method Agreements on ClueWeb09-B: the fraction of anchor-document pairs where different runs/methods choosing the same action (select/not select). w/ represents ReInfoSelect uses the following model as neural ranker.

Method	Discriminator	w/ Conv-KNRM	w/ BERT
Discriminator	0.924	0.356	0.109
w/ Conv-KNRM	0.356	0.585	0.323
w/ BERT	0.109	0.323	0.623

Scratch starts by selecting a large fraction of weak supervision pairs, and quickly boosts the reward from the neural rankers. At the beginning of ReInfoSelect training, the neural rankers are nearly random, and raw anchor-document pairs can help boost their performance. As the training goes on, the neural rankers get better and better, thus being more selective in their training data. ReInfoSelect then discards more of the anchor-document pairs as adding them may lead to negative rewards.

The same intuitions are reflected by the two warm up versions as well. In P Warm Up, the model starts with a low selection rate, as the policy network is initialized by All Discriminator, which is trained to consider all anchors as negatives. However, it quickly learns to relax the selection rate, as the neural ranker is just initialized and most anchor-document pairs can help. After that, P Warm Up behaves rather similarly to Scratch. In R Warm Up, as the neural ranker is warmed up with meaningful weights, the data selection rates remain low and ReInfoSelect only picks those can further boosted the warmed up ranker’s accuracy.

The behaviors of ReInfoSelect with Conv-KNRM and BERT share similar trends in the stochastic training process. The main difference is that BERT uses more relaxed selection rate for more epochs. Its deep transformer networks are hard to optimize than Conv-KNRM’s embedding and CNN layers. On the reward side, all model variations converge to similar reward scores, showing the robustness of ReInfoSelect.

5.7 Data Selection Behaviors

This set of experiments analyze the data selection behaviors of ReInfoSelect, include the agreements between different models and human evaluations.

Method Agreements. The first experiment studies whether ReInfoSelect chooses different weak supervision signals for different neural rankers. Table 6 shows the agreement between the “All Discriminator” and ReInfoSelect when used with the two neural rankers. The agreements between the same model (diagonal elements) are evaluated on the five cross-validation runs. The rests are the average of between their five runs. All models read the anchor-document pairs in the same order.

All methods agree the most with themselves across different runs. Discriminator is more deterministic than ReInfoSelect, as expected. ReInfoSelect picks different data for Conv-KNRM and for BERT. The latter has already been pre-trained and has rather different, and much deeper, architectures than Conv-KNRM. ReInfoSelect disagrees with Discriminator the most. The data selection in Discriminator is disconnected from the target ranker.

Table 7: Human Evaluation on ClueWeb09-B. The numbers are the fraction of anchors (A) or anchor-document pairs (A-D) labeled as proper search queries or relevant query-document pairs. Selected and Discarded are the actions taken by the models. Percentages (%) are the fraction of Selected (the same on A and A-D). Early and Late refer to the training stage in ReInfoSelect: before and after 400 batches.

	Selected		Discarded		Selected % A & A-D
	A	A-D	A	A-D	
Discriminator	0.900	0.750	0.923	0.479	5%
With Conv-KNRM					
ReInfoSelect-Early	0.926	0.489	0.861	0.528	91%
ReInfoSelect-Late	0.942	0.540	0.892	0.432	56%
With BERT					
ReInfoSelect-Early	0.912	0.510	0.946	0.435	77%
ReInfoSelect-Late	0.929	0.560	0.912	0.378	63%

Human Evaluation. This experiment examines the anchor intuition and evaluates whether ReInfoSelect selects anchors that are considered reasonable web queries and whether the documents are considered as relevant for their anchors.

We recruit four graduate students to label anchor-document pairs. We present them 100 anchors, each associated with one linked document (pseudo positive) and one retrieved (pseudo negative) document, all randomly sampled and ordered. The judges are asked to provide two binary labels. The first is whether the anchor could be a reasonable web search query, and the second is which of the two documents is more relevant to the anchor. The majority of votes are used as the final label. Our judges agree well: their Cohen’s Kappa is 0.526 on anchor and 0.544 on anchor-document.

We then mix the labeled pairs into all pairs and feed them to Discriminator and ReInfoSelect. When feeding ReInfoSelect, we either mix the labeled pairs in the early part of the epoch, when the neural ranker is just initialized, and the late part, when the neural ranker is close to converge, to evaluate the behavior on the two stages. The results are shown in Table 7.

Aligned with the Anchor intuition, 90%+ anchors are rated as reasonable search queries. The relevance between anchors and linked documents is more ambivalent; many anchors are functional (e.g. “homepage”) than informational. ReInfoSelect shows significantly different behavior in Early and Late. As the neural ranker converging, ReInfoSelect selects pairs more and more similar to query-relevant documents, while in the beginning, especially with the less pre-trained Conv-KNRM, 91% data is selected. Its “selectiveness” is customized for the target’s training status.

Discriminator does a better job in picking a-d pairs similar to query and relevant documents, which is what it is trained for. However, it is too strict and only picks 5% pairs. The challenge is that it is only trained with limited target query and documents, thus may not generalize well to anchors that are good but different from the handful target queries. Another challenge is that the data selector is isolated with the target ranker, while ReInfoSelect provides some final push that elevates the target neural rankers another 3-5% compared to Discriminator.

Table 8: Case study for anchor-document pairs that only selected by Discriminator or ReInfoSelect. Document snippets are manually picked. Manual labels on whether the anchor or pair is relevant (+) or not (-) are shown in brackets.

Method	Anchor	Linked Document (Manual Snippet)
Discriminator	oregon short line (+)	...a rail line owned and operated by the union pacific railroad in the u.s. state of utah (-)...
	new york state police (+)	...2007 pictures published by the new york state police carefully (-)...
	greenwich time (+)	...search for : greenwich time greenwich sponsored links (-)...
	pojoaque pueblo (+)	...contact pojoaque pueblo here for your perusal is (+)...
ReInfoSelect	banner ads (-)	...online banner advertising blog - rupiz ads home about buzz (+)...
	bmw 325i (+)	...as soon as your order is finalized, your bmw 325i aftermarket clutch pivot pin is dispatched (+)...
	bible bee (+)	...general overview bible bee basics bee-attitudes statement of faith philosophy (+)...
	kansas city zoo (+)	...yahoo experience kansas city - kansas city's new zoo take a safari through africa (+)...
	invasive alien species (+)	...the north european and baltic network on invasive alien species (nobanis) is a gateway to (+)...
power load (-)	...in the aliens movie, to combat the queen alien, ripley stepped into her a power loader (+)...	

Table 9: Examples of selected anchors and manually picked similar queries.

ClueWeb09-B Query	Anchor
dieting	crash dieting
french lick resort and casino	tropicana casino & resort atlantic city
diabetes education	vegan menu for people with diabetes
income tax return online	personal income taxes
orange county convention center	orange county convention center
Robust04 Query	Anchor
most dangerous vehicles	vehicle injury cases
international art crime	art crimes
mexican air pollution	outdoor air pollution
commercial cyanide uses	cyanide pills
el nino	el nino
ClueWeb12-B13 Query	Anchor
wind power	wind power in pa
nba records	nba records
teddy bears	beanie baby teddy bears
benefits of yoga	benefits of yoga
balding cure	balding treatment

5.8 Case Study

We first study instances where ReInfoSelect and Discriminator chose different actions. The example anchor-document pairs are listed in Table 8. Some anchor and document pairs are quite similar to search queries and relevant documents, for example, “kansas city zoo” leads to the Yahoo! page about it. There are anchors that are more functional, e.g., “banner ads”, and too general, e.g., “power load”. The linked documents may also be irrelevant. For example, the pictures published by New York State Police are as irrelevant to “New York State Police”. Discriminator and ReInfoSelect behave differently and as shown in last experiment, the difference was mainly from the (dis)connection with the target Neu-IR model.

We also find many anchors selected by ReInfoSelect very similar to the actual queries in the TREC benchmarks. Table 9 lists some examples. These anchors reflect very similar information needs with the actual queries. For example, “vegan menu for people with diabetes” is a legit web search query; one can imagine it forming a search session with “diabetes education”. These echo the classic IR intuition that many anchors are similar to search queries, which are effectively selected by ReInfoSelect.

6 CONCLUSION

ReInfoSelect leverages the widely available anchor data to weakly supervise neural rankers and mitigates their dependency on large amounts of relevance labels. To handle the noises in anchor data, ReInfoSelect uses policy gradient to connect the demand—the needs of training signals from neural ranker—and the supply—anchor-document pairs, to select more effective weak supervision signals. On three widely studied TREC benchmarks, ReInfoSelect is the only weak supervision method that guides neural rankers stably outperform feature-based learning to rank methods. Using only publicly available data, it also nearly matches the effectiveness of training signals from private commercial search logs.

In our experiments, ReInfoSelect robustly selects better anchor-document pairs than previous weak supervision approaches disconnected from target neural models. We also found that ReInfoSelect tailors the weak supervision for each individual Neu-IR model as well as its convergence status. Intuitively, ReInfoSelect starts with providing as much supervision as possible when the ranker is random; then it further elevates the neural ranker’s performance using anchor-document pairs that well approximate query and relevant documents.

ReInfoSelect also provides a handy experiment ground to analyze the advantages and disadvantages of Neu-IR. We show results on how neural rankers work with feature-based learning to rank methods, what is the effective fine-tuning strategies on target ranking tasks, and how different training data amounts and query-document fractions influence Neu-IR models. These analyses were hard to do as Neu-IR models were limited by the lack of large scale relevance-specific supervision. ReInfoSelect provides a simple way to lessen this dependency thus will facilitate and broaden the impact of deep learning research in information retrieval.

ACKNOWLEDGMENTS

Kaitao Zhang and Zhiyuan Liu are supported by the National Natural Science Foundation of China (NSFC61661146007, 61772302). Si Sun, Houyu Zhang, Kaitao Zhang, and Zhenghao Liu conducted human evaluations. We thank Guoqing Zheng, Paul Bennett, and Susan Dumais for discussions in weak supervision methodologies, the Anchor intuition, and the iterative view of pretraining-application. We thank Zhuyun Dai and Jamie Callan for sharing the base retrieval results on ClueWeb09-B and Robust04.

REFERENCES

- [1] Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. 2019. ReQA: An Evaluation for End-to-End Answer Retrieval Models. CoRR abs/1907.04780 (2019).
- [2] Adam L. Berger and John D. Lafferty. 1999. Information Retrieval as Statistical Translation. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*. 222–229.
- [3] W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines - Information Retrieval in Practice*. Pearson Education.
- [4] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. 985–988.
- [5] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. 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, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*. 126–134.
- [6] Van Dang and W. Bruce Croft. 2010. Query reformulation using anchor text. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*. 41–50.
- [7] Mostafa Dehghani, Aliaksei Severyn, Sascha Rothe, and Jaap Kamps. 2017. Learning to Learn from Weak Supervision by Full Supervision. CoRR abs/1711.11383 (2017).
- [8] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. 65–74.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. 4171–4186.
- [10] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query Expansion with Locally-Trained Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.
- [11] Zhicheng Dou, Ruihua Song, Jian-Yun Nie, and Ji-Rong Wen. 2009. Using anchor texts with their hyperlink structure for web search. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*. 227–234.
- [12] Nadav Eiron and Kevin S. McCurley. 2003. Analysis of anchor text for web search. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*. 459–460.
- [13] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*. 55–64.
- [14] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2019. A Deep Look into Neural Ranking Models for Information Retrieval. CoRR abs/1903.06902 (2019).
- [15] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. 8536–8546.
- [16] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. 10477–10486.
- [17] Sebastian Hofstätter, Navid Rekasaz, Carsten Eickhoff, and Allan Hanbury. 2019. On the Effect of Low-Frequency Terms on Neural-IR Models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. 1137–1140.
- [18] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2042–2050.
- [19] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM '13, San Francisco, CA, USA, October 27 - November 1, 2013*. 2333–2338.
- [20] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A Position-Aware Neural IR Model for Relevance Matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. 1049–1058.
- [21] Reiner Kraft and Jason Y. Zien. 2004. Mining anchor text for query refinement. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*. 666–674.
- [22] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. 6086–6096.
- [23] Zheng-Hao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 2395–2405.
- [24] Cheng Luo, Yukun Zheng, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2017. Training Deep Ranking Model with Weak Relevance Labels. In *Databases Theory and Applications - 28th Australasian Database Conference, ADC 2017, Brisbane, QLD, Australia, September 25-28, 2017, Proceedings*. 205–216.
- [25] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. 1101–1104.
- [26] Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder. 2019. Content-Based Weak Supervision for Ad-Hoc Re-Ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. 993–996.
- [27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. 3111–3119.
- [28] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*.
- [29] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. CoRR abs/1901.04085 (2019).
- [30] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. CoRR abs/1904.08375 (2019).
- [31] Harshith Padigela, Hamed Zamani, and W. Bruce Croft. 2019. Investigating the Successes and Failures of BERT for Passage Re-Ranking. CoRR abs/1905.01758 (2019).
- [32] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching as Image Recognition. , 2793–2799 pages.
- [33] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*. 257–266.
- [34] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. 1532–1543.
- [35] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. 2227–2237.
- [36] Mary Arpita Pyreddy, Varshini Ramaseshan, Narendra Nath Joshi, Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Consistency and Variation in Kernel Neural Ranking Model. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 961–964.
- [37] Yifan Qiao, Chenyan Xiong, Zheng-Hao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. CoRR abs/1904.07531 (2019).
- [38] Chen Qu, Feng Ji, Minghui Qiu, Liu Yang, Zhiyu Min, Haiqing Chen, Jun Huang, and W. Bruce Croft. 2019. Learning to Selectively Transfer: Reinforced Transfer Learning for Deep Text Matching. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*. 699–707.
- [39] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to Reweight Examples for Robust Deep Learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*. 4331–4340.

- [40] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*. 101–110.
- [41] Bo Wang, Minghui Qiu, Xisen Wang, Yaliang Li, Yu Gong, Xiaoyi Zeng, Jun Huang, Bo Zheng, Deng Cai, and Jingren Zhou. 2019. A Minimax Game for Instance based Selective Transfer Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 34–43.
- [42] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8 (1992), 229–256.
- [43] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2017. Adapting Markov Decision Process for Search Result Diversification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. 535–544.
- [44] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-Entity Duet Representations for Document Ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. 763–772.
- [45] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. 55–64.
- [46] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. 2018. Towards Better Text Understanding and Retrieval through Kernel Entity Saliency Modeling. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 575–584.
- [47] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. 2019. Critically Examining the "Neural Hype": Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. 1129–1132.
- [48] Hamed Zamani and W. Bruce Croft. 2017. Relevance-based Word Embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. 505–514.
- [49] Hamed Zamani and W. Bruce Croft. 2018. On the Theory of Weak Supervision for Information Retrieval. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2018, Tianjin, China, September 14-17, 2018*. 147–154.
- [50] Hamed Zamani, W. Bruce Croft, and J. Shane Culpepper. 2018. Neural Query Performance Prediction using Weak Supervision from Multiple Signals. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 105–114.
- [51] Wei Zeng, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2018. Multi Page Search with Reinforcement Learning to Rank. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2018, Tianjin, China, September 14-17, 2018*. 175–178.
- [52] Hongfei Zhang, Xia Song, Chenyan Xiong, Corby Rosset, Paul N. Bennett, Nick Craswell, and Saurabh Tiwary. 2019. Generic Intent Representation in Web Search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. 65–74.
- [53] Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Sogou-QCL: A New Dataset with Click Relevance Label. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 1117–1120.
- [54] Yukun Zheng, Yiqun Liu, Zhi-Qiang Fan, Cheng Luo, Qingyao Ai, Min Zhang, and Shaoping Ma. 2019. Investigating Weak Supervision in Deep Ranking. *Data and Information Management* 3 (2019), 155 – 164.