Generalized Weak Supervision for Neural Information Retrieval

YEN-CHIEH LIEN, University of Massachusetts Amherst, Amherst, MA
HAMED ZAMANI, University of Massachusetts Amherst, Amherst, MA
W. BRUCE CROFT, University of Massachusetts Amherst, Amherst, MA

Neural ranking models (NRMs) have demonstrated effective performance in several information retrieval (IR) tasks. However, training NRMs often requires large-scale training data, which is difficult and expensive to obtain. To address this issue, one can train NRMs via weak supervision, where a large dataset is automatically generated using an existing ranking model (called the weak labeler) for training NRMs. Weakly supervised NRMs can generalize from the observed data and significantly outperform the weak labeler. This paper generalizes this idea through an iterative re-labeling process, demonstrating that weakly supervised models can iteratively play the role of weak labeler and significantly improve ranking performance without using manually labeled data. The proposed Generalized Weak Supervision (GWS) solution is generic and orthogonal to the ranking model architecture. This paper offers four implementations of GWS: self-labeling, cross-labeling, joint cross- and self-labeling, and greedy multi-labeling. GWS also benefits from a query importance weighting mechanism based on query performance prediction methods to reduce noise in the generated training data. We further draw a theoretical connection between self-labeling and Expectation-Maximization. Our experiments on four retrieval benchmarks suggest that our implementations of GWS lead to substantial improvements compared to weak supervision if the weak labeler is sufficiently reliable.

Additional Key Words and Phrases: weak supervision, distant supervision, neural ranking models, zero-shot learning, unsupervised learning.

ACM Reference Format:
Yen-Chieh Lien, Hamed Zamani, and W. Bruce Croft. 2024. Generalized Weak Supervision for Neural Information Retrieval. 1, 1 (April 2024), 26 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Deep neural networks have shown promising results in many retrieval tasks, including ad-hoc retrieval [10, 25, 28, 54], conversational search [13, 33, 48], and cross-modal retrieval [9, 16]. Training existing neural ranking models (NRMs) often requires a large amount of training data. However, obtaining such a large training set is often difficult and expensive.

This paper focuses on training NRMs when no manually labeled data is available for training. A straightforward solution to tackle this problem is to use large-scale pre-trained language models, e.g., BERT [8], as zero-shot ranking models. However, since these models are not optimized for retrieval tasks, their zero-shot performance for retrieval tasks is limited. They even perform poorer than term-matching models, such as BM25 [37]. This is why these models are often fine-tuned using labeled training data.

Authors' addresses: Yen-Chieh Lien, University of Massachusetts Amherst, Amherst, MA, ylien@cs.umass.edu; Hamed Zamani, University of Massachusetts Amherst, Amherst, MA, zamani@cs.umass.edu; W. Bruce Croft, University of Massachusetts Amherst, Amherst, MA, croft@cs.umass.edu.

XXXX-XXXX/2024/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

An alternative solution to this problem is to train NRMs using noisy training signals produced by existing (unsupervised) retrieval models. This teacher-student learning approach is called *weak supervision* [7, 51], and the teacher model is often called the weak labeler. Weak supervision addresses the data scarcity issue by leveraging unsupervised methods to infer a noisy ranked list and uses that signal as ground truth for training a neural ranking model. In this line of research, classical IR methods, such as BM25, are usually selected as the weak labeler [7, 53]. There exists numerous theoretical and empirical evidence that weakly supervised models can significantly outperform their weak labelers [7, 51, 53]. This paper generalizes the weak supervision formulation such that a weakly supervised model iteratively becomes the weak labeler. We hypothesize that such an approach should lead to performance improvement since the quality of weakly supervised training data is iteratively improved. Based on this hypothesis, we propose *Generalized Weak Supervision (GWS)*, a generic framework for training neural ranking models with no labeled data. We offer four implementations of this framework. The first implementation is called *self-labeling*, in which one weakly supervised model iteratively produces the training data for the next iteration. The second implementation is called *cross-labeling*. In this case, we use two NRMs M_1 and M_2 exchanging information by playing the roles of teacher models for one another in weak supervision. In other words, each model is optimized using the training data produced by the other model, and the role of the teacher model alternates between them. The third implementation is called *joint cross- and self-labeling*. As the combination of the previous two implementations, we also have two NRMs M_1 and M_2 as teacher and student models. Different from *cross-labeling*, which exchanges weak signals in each iteration, after each model alternation (i.e., cross-labeling), we apply *self-labeling* to train the student model thoroughly and then repeat the cross-labeling process. The last implementation is called *greedy multi-labeling*, in which we train several model checkpoints based on weak supervision signals generated from all ranking models and pick the best one to represent this structure as the signal provider (i.e., the teacher) for the next iteration. In other words, the best-performing students at every iteration become the teachers for the next iteration.

This paper also draws theoretical connections between the simplest implementation of the proposed GWS framework (i.e., self-labeling) and the Expectation–maximization (EM) algorithm, a well-known framework for unsupervised learning which has been successfully used for a wide range of tasks, including semi-supervised text classification [27], transfer learning [21], language model estimation [14], and pseudo-relevance feedback [55].

We further survey techniques for enhancing the effectiveness of GWS training. To this aim, we study query importance for the weak supervision training process. Intuitively, we would like to train NRMs by emphasizing the queries for which the weak labeler produces high-quality results. This will reduce the level of noise in the training set. Based on this intuition, we leverage existing query performance prediction (QPP) models that have been studied in the information retrieval literature for decades [5, 39, 52], and propose an in-batch weighting method of training instances to modify the importance of queries based on the prediction of QPP models.

In our experiments, we evaluate the proposed methods using four publicly available datasets: (1) WikiPassageQA [4], a passage retrieval dataset based on Wikipedia articles; (2) ANTIQUE [12] a passage retrieval dataset for non-factoid questions submitted by real users to community question answering websites; (3) NQ [20], an open domain question answering dataset for real users' questions and corresponding answering based on Wikipedia; and (4) MSMARCO, a large scale passage ranking dataset. In our experiments, we follow Dehghani et al. [7] and adopt BM25 [36] as the initial weak labeler. We train two pre-trained language models, BERT [8] and RoBERTa [22], using (generalized) weak supervision. For QPP, we use Normalized Query Commitment (NQC) [39], a popular unsupervised QPP method for predicting the performance of the initial weak labeler

for each training query. Further, we adopt an in-batch re-weighting training process to incorporate NQC into the loss function.

To summarize, the contributions of this paper include:

- Proposing the Generalized Weak Supervision framework.
- Introducing four implementations of the GWS framework.
- Drawing connections between GWS and the EM algorithm.
- Enhancing GWS training by leveraging query performance prediction models for query importance.
- Demonstrating substantial improvements in ranking quality compared to zero-shot and weakly supervised baselines. Note that these improvements are solely observed using automatic relabeling of training data with no manually labeled data.

2 RELATED WORK

In this section, we first review two of the most relevant lines of research to this paper: neural ranking models and weak supervision. We further provide a brief review of prior work on self-labeling, knowledge distillation, and domain adaptation. Even though these three topics are not directly related to the contributions of this work, there are some connections that are worth exploring.

2.1 Neural Ranking Model

In recent years, several neural ranking models have been proposed for retrieval tasks. DSSM [17] and C-DSSM [38] adopted a method to learn the representation of query and document individually and use a matching function to score. The deep relevance matching model [10] exploits histogram feature to represent the interaction between query and document as the input of neural ranking architecture. DUET [25] uses two networks to learn local interaction and distributed matching between query and document respectively.

After BERT [8] is proposed, large-scale pre-trained language models are widely applied to ranking problems. For instance, Nogueira and Cho [28] used BERT for passage ranking and demonstrated significant improvement. Han et al. [11] combined learning-to-rank and the ensemble of BERT [8], RoBERTa [22] and ELECTRA [2] for passage ranking. Qu et al. [32] apply BERT for the conversational question answering task.

The mentioned neural ranking models focus on re-ranking problems, where an efficient first-stage retrieval model, such as BM25, provides a small list of documents for re-ranking. Zamani et al. [53] demonstrated for the first time that neural models can be used for document retrieval from a large collection without the need to a multi-stage cascaded retrieval architecture. This phenomenon was later adopted and applied to dense query and document embedding with the use of approximate nearest-neighbor search algorithms. Such dense retrieval approaches, such as DPR [18], use a dual encoder architecture to encode queries and documents separately and compute their similarity using simple matching functions, such as dot product or cosine similarity. Several works [19, 34, 46, 54] were proposed based on the dense retrieval setting to move from re-ranking to ranking.

The vast majority of recent neural ranking models are trained on large data collections, such as MS MARCO, and do not focus on the issue of data volume. In this work, we aim to propose a general framework for training neural ranking models without a need to ground truth labels. Therefore, the proposed approach can potentially be applied to any of the existing neural ranking model architectures listed above.

2.2 Weak Supervision for IR

As the motivation of this paper, weak supervision tries to solve the problem of data volume for neural ranking models. Dehghani et al. [7] first proposed weak supervision to train a neural ranking model based on the labels generated by existing retrieval methods, e.g., BM25 or heuristics. They empirically showed that weakly supervised neural ranking models can significantly improve their weak labeler, solving an important problem in optimizing large deep learning models without labeled data. Later, Zamani and Croft [51] provided theoretical insights into weak supervision for information retrieval.

Several works [26, 43, 52] exploited weak supervision on specific IR tasks. Voskarides et al. [43] used automatically generated data for fact ranking in a knowledge graph. Zamani et al. [52] leveraged multiple weak signals for query performance prediction (QPP). Nie et al. [26] used weak supervision to train the retrieval model with multi-level matching. Zamani and Croft [50] used weak supervision for learning relevance-based word embeddings. Given the success of weak supervision in IR, a number of approaches focused on strengthening the effectiveness of weak supervision. Zhang et al. [57] applied reinforcement learning to select anchor-document pairs for training weakly supervised neural ranking models. Some methods [24, 47] adopt pre-trained language models like BERT as the weakly supervised ranking model. In this paper, we also follow this setting and use pre-train language models as the retrieval model. Different from enhancing weak supervision by switching models, improving data selection, or broadening the application, our method aims to generalize the whole weak supervision framework. That is, GWS can cooperate with all previous related works and further improve them.

Previous works solely rely on one or more weak labeler to train their model. In this paper, we generalize this approach such that the weakly supervised models in each step become weak labelers in the next step. The proposed framework is sufficiently generic to be applied to any weakly supervised model.

2.3 Self-Labeling

Self-labeling is widely used for semi-supervised learning problems. By directly imputing ground truth labels for unlabeled instances, self-labeling propagates labels to unknown target data. Nigam et al. [27] applied self-labeling to semi-supervised text classification using an Expectation-Maximization (EM) algorithm. Chen et al. [1] designed an algorithm for semi-supervised sentiment classification by iterative imputing sentiment labels for unlabeled reviews according to the current model's confidence score on the data.

Among all, we found the one conducted by Li et al. [21] the most relevant to ours. The authors trained an initial ranker from ground truth labels on a source domain and used self-labeling to label the target domain's data. Then, they re-trained the ranker on the target data from self-labeling and repeated the above operation until convergence. The steps for the target domain are similar to ours. However, their task is transfer learning, which needs large-scale ground truth labels on the source domain. In our setting, we do not include any labeled data at any stage of our training.

2.4 Knowledge Distillation for IR

To achieve the performance of neural models with lower computational cost, a common approach is to distill knowledge from large teacher neural models into smaller student models. For pre-trained language models like BERT, DistilBERT, and TinyBERT are proposed to create light-weight models when maintaining the performance on various tasks using distillation.

Due to the success of pre-trained language models on IR tasks, there are several works on applying knowledge distillation on IR. Zeng et al. [54] proposed a curriculum learning framework

to optimize student dense retrieval models from teacher re-ranking models. Vakili Tahami et al. [42] proposed a new cross-encoder architecture to transfer its knowledge to a low-cost bi-encoder for the response retrieval task. Hofstätter et al. [15] proposed a cross-architecture training procedure to adapt knowledge distillation to the varying output score distributions from different neural models.

Although the relationship between teacher and student models is similar to weak signals and weakly supervised models, there are two main differences between the two tasks. First, in the setting of knowledge distillation, label information is available especially for the teacher models' training, and the goal is to create low-cost inference when having a supervised model. Second, knowledge distillation uses a smaller student model to approximate the performance of a larger one, while this is not the case in weak supervision, and some labels can even come from simple non-ML models.

2.5 Domain Adaptation for Neural IR

Because there exist some massive IR datasets like MS MARCO as a rich source domain, domain adaptation is also a crucial solution for neural IR to solve the high dependency on in-domain data. Cohen et al. [3] did early work on domain adaptation for neural retrieval by cross-domain adversarial learning, but it did not include pre-trained models from a source domain.

Recent works exploited pre-trained IR models from existing data on other retrieval tasks. Wang et al. [44] trained doc2query T5 model and retrieval models on the source domain, used T5 to generate pseudo-queries on the target domain, and then applied a pre-trained dense retrieval model and a cross-encoder model to build pseudo pairwise data for training a new retrieval model on the target domain. Sun et al. [40] not only built pseudo-labeled data on the target domain by pre-train models on the source domain but also added a meta-learning method to learn meta weighting on synthetic data to exploit weak supervision signals better. Different from weak supervision, Zhan et al. [56] split a retrieval component into two modules, Relevance Estimation Module (REM) and Domain adaptation Module (DAM), to deal with general relevance matching and adaptation to the target domains. Even though domain adaptation focuses on solving data scarcity and sometimes includes weak supervision, the general assumption of domain domain adaptation is to have an initial rich data source for transferring. We deal with a different problem, which does not assume the existence of a rich source domain with large-scale labeled data, and aim to train a feasible model without any relevance information in all domains. However, for the works adopting weak supervision as a part of the solution, the proposed GWS can potentially be incorporated.

3 FORMULATING WEAK SUPERVISION FOR IR

Given a query q and a document collection C, the task of ad-hoc information retrieval is to develop a retrieval model M_θ parameterized by θ for retrieving documents from C with respect to their relevance to the query q in descending order. Unsupervised approaches for ad-hoc retrieval mostly focus on term matching between the query and document content, such as TF-IDF [35] , BM25 [37], and query likelihood [30]. There also exist supervised ranking models that learn from a manually labeled training set. Weak supervision is an approach for training retrieval models without any manually labeled data. It uses unsupervised retrieval models (called the weak labeler), e.g., BM25, to automatically annotate queries and documents for training learning to rank models.

For every training query $q \in Q$, weak supervision uses a weak labeler $\widehat{M}$ to retrieve a list of documents D from C and creates a set of triplets $T_{\widehat{M}} = \{(q, d, \widehat{M}(q, d)) : \forall q \in Q, \forall d \in D\}$. This training set can be considered as noisy ground truth and thus can be used for training weak

supervision models as follows:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(M_\theta, T_{\widehat{M}})$$

where Q and $\mathcal{L}$ denote the training query set and the ranking loss function, respectively. The query set Q can be sampled from a search engine's query logs or questions in community question answering forums. It can also be automatically generated using autoregressive query generation models or even by random n-gram selection from a corpus. The loss function $\mathcal{L}$ can be implemented using any of the pointwise, pairwise, and listwise ranking loss functions. Zamani and Croft [51] proved that the weak supervision loss function $\mathcal{L}$ should be symmetric in order to be robust to the weak supervision noise. They demonstrated that Hinge loss satisfies this property.

4 GENERALIZED WEAK SUPERVISION

In this section, we introduce the generalized weak supervision (GWS) framework for information retrieval. GWS is a general framework for training ranking models. The model parameters in GWS are first initialized using typical weak supervision approaches. Next, GWS runs an iterative process. In each iteration, it re-labels the training data and uses the new training set for training another ranking model. GWS repeats this process until a stopping criterion is met.

GWS can work with one single ranking model or multiple ranking models by changing the re-labeling settings. In this work, we provide four different settings.

(1) Algorithm 1 introduces GWS with self-labeling, in which a single ranking model iteratively re-labels the dataset and reuses it for optimization.
(2) Algorithm 2, on the other hand, introduces the weak labeling alternation implementation of GWS, in which the relabeling process alternates between k weakly supervised rankers.
(3) Algorithm 3 is the combination of the above two. Ranking models also provide weak signals to the other model in the manner of weak supervision but apply Algorithm 1 to train a model thoroughly before exchanging.
(4) Finally, Algorithm 4 also adopts a multi-model setting, but it considers all teacher-student combinations in each iteration and chooses the best one for a model structure as the teacher model for the next round.

We provide a conceptual demonstration with two ranking model for all four implementations n Figure 1. Note that the red circle in Figure 1d means the best checkpoint in the iteration. To simplify the understanding, we only show the route starting from model 1 in Figure 1b and 1c, but the route starting from model 2 is also conducted in parallel.

The following subsections provide in-depth details and justification for all of these implementations of GWS. We first explain the initialization of these models, which is similar in all these four implementations. We then explain different re-labeling implementations. We also discuss the relationship between GWS and Expectation-Maximization. We show the notations used for the explanation in Table 1.

4.1 Initialization in GWS

The first step in GWS is to train the initial weakly supervised model using the typical weak supervision setup introduced by Dehghani et al. [7]. All the four re-labeling algorithms demonstrate that the initial weak labeling model is initialized by $\widehat{M}$, an existing unsupervised retrieval model, such as BM25 [36].

Even though algorithms provide a general implementation of weak supervision, we only use the top k retrieved documents by $\widehat{M}$ instead of all documents in the collection. This has been done for efficiency considerations. Thus, for every query $q_i \in Q$, let $\{d_{i1}, d_{i2}, ..., d_{ik}\}$ be the top k documents

Table 1. Notations for GWS framework

Notation	Definition
Q	A query set
D	A document set
θ	Model parameters
$\theta^{(i)}$	Model parameters of the i^{th} neural structure
$\theta^{(i,j)}$	Model parameters of the i^{th} neural structure trained on the triplets generated from the j^{th} neural structure
M	A ranking model
M_θ	A ranking model parameterized by θ
T_M	A set of triplets generated by a model M
$\theta^{(I)}$ / $M^{(i)}$ / $T^{(i)}$	Model parameters / A model / A set of triplets from the i^{th} neural structure.
V	A validation error function

retrieved by $\widehat{M}$. Therefore, the training triplets for this query include $\{(q_i, d_{i1}, \widehat{M}(q_i, d_{i1})), \cdots, (q_i, d_{ik}, \widehat{M}(q_i, d_{ik}))\}$. Therefore, the initial training set $T_{\widehat{M}}$ consists of $|Q| \times k$ query-document pairs. Again, for efficiency reasons, we only re-score these documents in the following iterations of the GWS framework. In the following subsections, the training data used in iteration t is denoted as $T_{M_{\theta_t}}$, which is generated from a model M_{θ_t} parameterized by θ_t.

4.2 Iterative Re-Labeling and Training in GWS

After building an initial model on weak supervision signals, we need to re-label the data by the current checkpoint, i.e., re-scoring all the triplets in the training data, and train a new model on these updated weak supervision signals.

In the t^{th} iteration, we optimize the model parameter θ_t based on the weakly supervised data $T_{M_{\theta_{t-1}}}$ generated from re-labeling in the previous iteration. Let the loss function be $\mathcal{L}(M, T)$ for the model M and the data T, we have the following update in the training phase:

$$\theta_t = \arg\min_\theta \mathcal{L}(M_\theta, T_{M_{\theta_{t-1}}})$$

Note that the update is not related to θ_{t-1} since the operation is re-training a new model instead of fine-tuning the last parameter. We empirically found that starting from the initial model would lead to higher performance. The reason is that fine-tuning the last iteration is likely to overfit the produced data.

In practice, when the training is done after n iterations, we will pick the final model from all intermediate models based on the validation error function V:

$$\theta_{\text{final}} = \arg\min_{\theta \in \{\theta_1, \theta_2, \ldots \theta_n\}} V(\theta)$$

Note that we still need a small validation set for V to judge which checkpoints are the best ones for our tasks. We leave the fully unsupervised judgment to the future extension.

In the following, we introduce four re-labeling algorithms described below.

4.2.1 Self-Labeling. In Algorithm 1, we exploit the intermediate model as a new weak labeler to build new data for the next training iteration. Assume in the t^{th} training iteration, after training on weak supervision data T_{t-1}, we get a ranking model M_{θ_t}. For the next iteration, we aim to build new weak supervision data D_t by M_{θ_t}. Regarding M_{θ_t} as the next weak labeler, we can update the

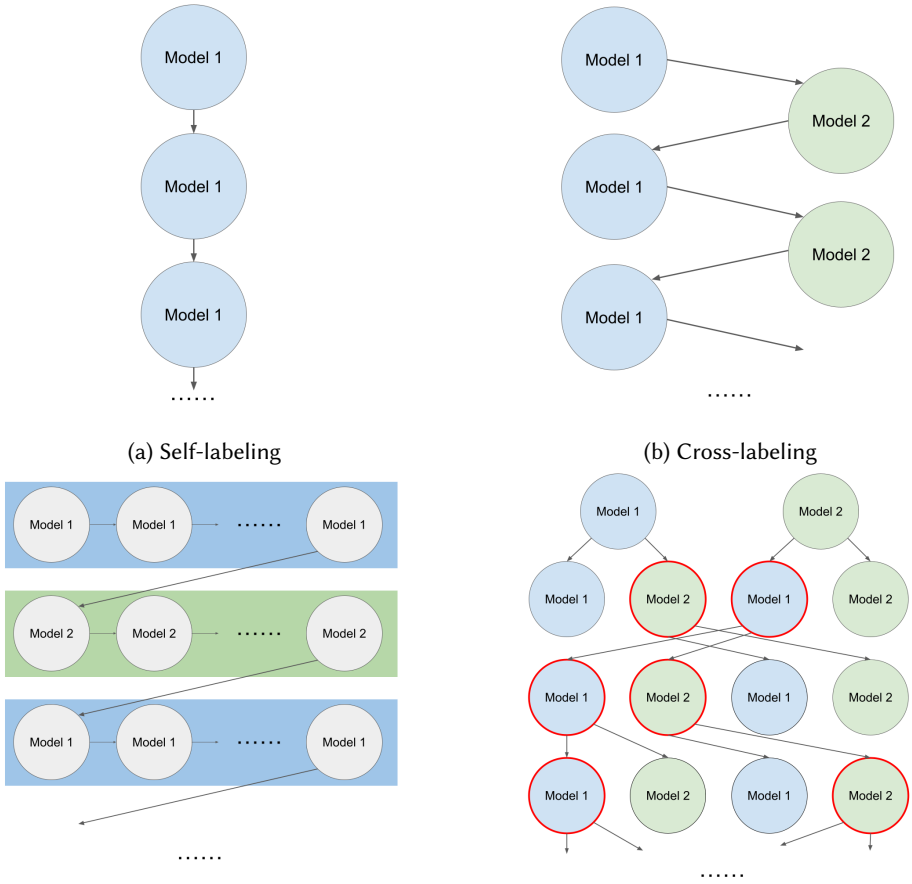


Fig. 1. Different GWS implementation for single-model and multi-model setting.

relevance score in D_{t-1}. In summary, we have the following update in the self-labeling phase:

$$T_{M_{\theta_t}} = \{(q_i, d_{ij}, M_{\theta_t}(q_i, d_{ij}))\} \text{ where } 1 \leq i \leq |Q|, 1 \leq j \leq k$$

Again, we focus on the re-ranking problem in this work and only update scores for the same pairs as the previous data.

Instead of re-labeling by only one model (i.e., self-labeling), we can use multiple weakly supervised models for re-labeling. The intuition is to increase the information diversity for the model in GWS. Because the training process runs on the same dataset by the same neural architecture, the over-fitting problem may deteriorate in iterative training. Thus, multi-model approaches aim to include different neural architectures in GWS to avoid this problem. In the following, we will introduce different implementations to let models be optimized and exchange their information.

4.2.2 Cross-labeling. As an alternative approach to self-labeling, Algorithm 2 aims to train m ranking models at each iteration of GWS training and exchange the generated weak signals. In our

Algorithm 1 Generalized Weak Supervision via Self-Labeling

Input (a) a set of queries Q; (b) a document collection C; (c) an unsupervised retrieval model $\widehat{M}$; (d) a loss function $\mathcal{L}$
Output a ranking model M_θ.
$M' \leftarrow \widehat{M}$
repeat
  Initialize θ.
  $T \leftarrow \emptyset$
  for $q \in Q$ **do**
    $T \leftarrow T \cup M'(q, C)$
  end for
  $\theta \leftarrow \arg\min_\theta \mathcal{L}(M_\theta, T)$
  $M' \leftarrow M_\theta$
until convergence
return M_θ

experiments, we show that even the simplest case where $m = 2$ improves self-labeling. That being said, Algorithm 2 can be used for any $m > 1$ models.

Without loss of generality, consider we have two ranking models parameterized by $\theta_t^{(1)}$ and $\theta_t^{(2)}$ at iteration t. In the re-labeling process of the t^{th} iteration, the two models generate two sets of weak supervision data, as follows:

$$T_{M_{\theta_t^{(1)}}} = \{(q_i, d_{ij}, M_{\theta_t^{(1)}}(q_i, d_{ij}))\} \text{ where } 1 \le i \le |Q|, 1 \le j \le k.$$

$$T_{M_{\theta_t^{(2)}}} = \{(q_i, d_{ij}, M_{\theta_t^{(2)}}(q_i, d_{ij}))\} \text{ where } 1 \le i \le |Q|, 1 \le j \le k.$$

During training, each model is trained on the data produced by the other model. Therefore, we have:

$$\theta_{t+1}^{(1)} = \arg\min_{\theta^{(1)}} \mathcal{L}(M_{\theta^{(1)}}, T_{M_{\theta_t^{(2)}}})$$

$$\theta_{t+1}^{(2)} = \arg\min_{\theta^{(2)}} \mathcal{L}(M_{\theta^{(2)}}, T_{M_{\theta_t^{(1)}}})$$

Through the operations, two models can exchange information during learning and avoid overfitting caused by self-labeling. In the case of two models, we easily set the supervision source as the other. For multiple models, we can choose a random one for each model and build one-to-one matching before training.

4.2.3 Joint Cross- and Self-labeling (JCS). Algorithm 3 combines self-labeling and cross-labeling settings. This approach still exchanges the generated weak signals among ranking models. However, it runs a self-labeling process for each ranking model before exchanging labels. Different from cross-labeling, Algorithm 3 aims to exchange the label from each model after convergence through self-labeling. As in the setting of cross-labeling, in the following, we only consider the simplest case where $m = 2$,

Without loss of generality, consider we have two ranking models parameterized by $\theta_t^{(1)}$ and $\theta_t^{(2)}$ at iteration t. In the re-labeling process at the t^{th} iteration, two models generate two sets of weak supervision data, as follows:

$$T_{M_{\theta_t^{(1)}}} = \{(q_i, d_{ij}, M_{\theta_t^{(1)}}(q_i, d_{ij}))\} \text{ where } 1 \le i \le |Q|, 1 \le j \le k.$$

Algorithm 2 Generalized Weak Supervision via Cross Labeling

Input (a) a set of queries Q; (b) a document collection C; (c) an unsupervised retrieval model $\widehat{M}$; (d) a loss function $\mathcal{L}$.
Output m ranking models $M_{\theta^{(1)}}, M_{\theta^{(2)}}, \cdots, M_{\theta^{(m)}}$.
$M^{(1)}, M^{(2)}, \cdots, M^{(m)} \leftarrow \widehat{M}$
repeat
  $T^{(1)}, T^{(2)}, \cdots, T^{(m)} \leftarrow \emptyset$
  Initialize $\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(m)}$.
  for $i \in [1, 2 \cdots, m]$ **do**
    for $q \in Q$ **do**
      $T^{(i)} \leftarrow T^{(i)} \cup M^{(i)}(q, C)$
    end for
  end for
  for $i \in [1, 2 \cdots, m]$ **do**
    $\theta^{(i)} \leftarrow \arg\min_{\theta^{(i)}} \mathcal{L}(M_{\theta^{(i)}}, T^{(i-1)})$
    $M^{(i)} \leftarrow M_{\theta^{(i)}}$
  end for
until convergence
return $M_{\theta^{(1)}}, M_{\theta^{(2)}}, \cdots, M_{\theta^{(m)}}$

$$T_{M_{\theta_t^{(2)}}} = \{(q_i, d_{ij}, M_{\theta_t^{(2)}}(q_i, d_{ij}))\} \text{ where } 1 \le i \le |Q|, 1 \le j \le k.$$

Following Algorithm 1, each model is trained on the data produced by itself. Therefore, we have:

$$\theta_{t+1}^{(1)} = \arg\min_{\theta^{(1)}} \mathcal{L}(M_{\theta^{(1)}}, T_{M_{\theta_t^{(1)}}})$$

$$\theta_{t+1}^{(2)} = \arg\min_{\theta^{(2)}} \mathcal{L}(M_{\theta^{(2)}}, T_{M_{\theta_t^{(2)}}})$$

Assume two models converge in the $L_1{}^{\text{th}}$ and $L_2{}^{\text{th}}$ iteration through the self-labeling process, we do an additional update to exchange the labels as the following:

$$\theta_{L_1+1}^{(1)} = \arg\min_{\theta^{(1)}} \mathcal{L}(M_{\theta^{(1)}}, T_{M_{\theta_{L_2}^{(2)}}},)$$

$$\theta_{L_2+1}^{(2)} = \arg\min_{\theta^{(2)}} \mathcal{L}(M_{\theta^{(2)}}, T_{M_{\theta_{L_1}^{(1)}}},)$$

After the exchange, we start a re-labeling process again as before.

4.2.4 Greedy Multi-Labeling. Greedy multi-labeling is a generalized version of cross-labeling. Different from choosing one fixed weak signal provider for each model as in Algorithms 2 and 3, we consider all possible m models to build m weak signal sets for one model structure, train m checkpoints and pick the best one as the signal provider for the next iteration. In other words, at each iteration, we use all m weak labelers as teachers, train all m students, and then select the best student models.

Consider we have two ranking models parameterized by $\theta_t^{(1)}$ and $\theta_t^{(2)}$ at iteration t. In the re-labeling of the t^{th} iteration, two models generate their own weak supervision data:

$$T_{M_{\theta_t^{(1)}}} = \{(q_i, d_{ij}, M_{\theta_t^{(1)}}(q_i, d_{ij}))\} \text{ where } 1 \le i \le |Q|, 1 \le j \le k.$$

$$T_{M_{\theta_t^{(2)}}} = \{(q_i, d_{ij}, M_{\theta_t^{(2)}}(q_i, d_{ij}))\} \text{ where } 1 \le i \le |Q|, 1 \le j \le k.$$

Algorithm 3 Generalized Weak Supervision via Joint Cross- and Self-Labeling

Input (a) a set of queries Q; (b) a document collection C; (c) an unsupervised retrieval model $\widehat{M}$; (d) a loss function $\mathcal{L}$.
Output m ranking models $M_{\theta^{(1)}}, M_{\theta^{(2)}}, \cdots, M_{\theta^{(m)}}$.
$M^{(1)}, M^{(2)}, \cdots, M^{(m)} \leftarrow \widehat{M}$
repeat
  $T^{(1)}, T^{(2)}, \cdots, T^{(m)} \leftarrow \emptyset$
  for $i \in [1, 2 \cdots, m]$ **do**
    for $q \in Q$ **do**
      $T^{(i)} \leftarrow T^{(i)} \cup M^{(i)}(q, C)$
    end for
  end for
  for $i \in [1, 2 \cdots, m]$ **do**
    $\theta^{(i)} \leftarrow \arg\min_{\theta} \mathcal{L}(M_{\theta}, T^{(i-1)})$
    $M^{(i)} \leftarrow M_{\theta^{(i)}}$
  end for
  for $i \in [1, 2 \cdots, m]$ **do**
    $M'_{\theta^{(i)}} \leftarrow$ Algorithm 1$(Q, C, M'_{(i)}, L)$
    $M'_{(i)} \leftarrow M_{\theta^{(i)}}$
  end for
until convergence
return $M_{\theta^{(1)}}, M_{\theta^{(2)}}, \cdots, M_{\theta^{(m)}}$

In the next iteration, each model needs to be trained on all weak supervision data. Therefore, we have m^2 candidate models θ' as follows:

$$\theta^{(1,1)} = \arg\min_{\theta^{(1)}} \mathcal{L}(M_{\theta^{(1)}}, T_{M_{\theta_t^{(1)}}})$$
$$\theta^{(1,2)} = \arg\min_{\theta^{(1)}} \mathcal{L}(M_{\theta^{(1)}}, T_{M_{\theta_t^{(2)}}})$$
$$\theta^{(2,1)} = \arg\min_{\theta^{(2)}} \mathcal{L}(M_{\theta^{(2)}}, T_{M_{\theta_t^{(1)}}})$$
$$\theta^{(2,2)} = \arg\min_{\theta^{(2)}} \mathcal{L}(M_{\theta^{(2)}}, T_{M_{\theta_t^{(2)}}})$$

For each model structure, we choose the best one as the weak signal provider in the next iteration as the following:

$$\theta_{t+1}^{(1)} = \arg\min_{\theta \in \{\theta^{(1,1)}, \theta^{(1,2)}\}} V(\theta)$$
$$\theta_{t+1}^{(2)} = \arg\min_{\theta \in \{\theta^{(2,1)}, \theta^{(2,2)}\}} V(\theta)$$

Where V is the validation error function for the candidate models. Again, we still need a small validation set to judge which checkpoints are the best ones for our tasks.

4.3 Relationship of GWS and Expectation-Maximization

To better understand the theoretical foundation of GWS, we draw a connection between GWS and Expectation-Maximization (EM), which has been widely explored in various machine learning tasks. To this aim, we need to revisit GWS from the probabilistic view. For simplicity, this section focuses on the self-labeling approach (Algorithm 1). Let $R \in \{0, 1\}$ be a binary random variable that represents whether a document is relevant to a query or not. Thus, self-labeling is equivalent to

Algorithm 4 Generalized Weak Supervision via Greedy Multi-Labeling

1: **Input** (a) a set of queries Q; (b) a document collection C; (c) an unsupervised retrieval model $\widehat{M}$; (d) a loss function $\mathcal{L}$; (e) a validation error function V.
2: **Output** m ranking models $M_{\theta^{(1)}}, M_{\theta^{(2)}}, \cdots, M_{\theta^{(m)}}$.
3: $M^{(1)}, M^{(2)}, \cdots, M^{(m)} \leftarrow \widehat{M}$
4: **repeat**
5: $\quad T^{(1)}, T^{(2)}, \cdots, T^{(m)} \leftarrow \emptyset$
6: $\quad$ **for** $i \in [1, 2 \cdots, m]$ **do**
7: $\quad\quad$ **Initialize** $\theta^{(1,i)}, \theta^{(2,i)}, \cdots, \theta^{(m,i)}$.
8: $\quad$ **end for**
9: $\quad$ **for** $i \in [1, 2 \cdots, m]$ **do**
10: $\quad\quad$ **for** $q \in Q$ **do**
11: $\quad\quad\quad T^{(i)} \leftarrow T^{(i)} \cup M^{(i)}(q, C)$
12: $\quad\quad$ **end for**
13: $\quad$ **end for**
14: $\quad$ **for** $i \in [1, 2 \cdots, m]$ **do**
15: $\quad\quad$ **for** $j \in [1, 2 \cdots, m]$ **do**
16: $\quad\quad\quad \theta^{(i,j)} \leftarrow \arg\min_{\theta^{(i,j)}} \mathcal{L}(M_{\theta^{(i,j)}}, T^{(j)})$
17: $\quad\quad$ **end for**
18: $\quad\quad G \leftarrow \arg\min_j V(\theta^{(i,j)})$
19: $\quad\quad M^{(i)} \leftarrow M_{\theta^{(i,G)}}$
20: $\quad$ **end for**
21: **until** convergence
22: **return** $M_{\theta^{(1)}}, M_{\theta^{(2)}}, \cdots, M_{\theta^{(m)}}$

inferring labels based on $P(R = 1|q, d; \theta)$ and $P(R = 0|q, d; \theta)$. For the iterative training, minimizing a loss function $\mathcal{L}(M_\theta, T)$ can be considered as the negative log-likelihood for the current relevance judgment in T.

Now let us focus on the EM algorithm, a general learning framework for unsupervised learning problems. Given a joint distribution $P(X, Z|\theta)$, where X is the observed data, Z is the hidden or missing variable, and θ is a set of model parameters, the EM algorithm aims to maximize $P(X|\theta)$ by the following steps:

(1) Initialize θ_0
(2) E-step: Estimate Z by $P(Z|X, \theta_{t-1})$
(3) M-step: $\theta_t = \arg\min_\theta - P(Z|X, \theta_{t-1}) \log P(X, Z|\theta)$
(4) Repeat step 2 and 3 until it converges.

Comparing the E-step and M-step of the EM algorithm with the probabilistic view of self-labeling and iterative training, it is clear that the process of GWS could be connected to EM if we regard R as the hidden variable Z; and Q and D as the observed data X. In other words, in Algorithm 1, lines 6-9 can be connected to the E-Step, and Lines 10-11 can be connected to the M-Step of the EM algorithm. However, the GWS framework behaves differently for initialization, which significantly affects the performance of the model.

The result of EM algorithm is always affected by the initialization of parameters. For retrieval tasks, random initialization on hidden variables (as often done in the EM algorithms), which are relevance judgments, is not applicable because relevance judgment is always complex and

imbalanced in the collection. On the other hand, training a ranking model on randomly generated ranked lists may not converge to an effective parameter setting for ranking tasks.

We regard our process as an EM process with a weak supervision initialization. Through weak supervision signals, we get non-random initialization and have noisy but useful results for the first expectation step. Therefore, the following EM process has an excellent base to generate a feasible model for ranking tasks.

For the other three labelings with multi-model settings, they could not be directly linked to EM process, but we consider them as a more generic process than EM. Our experiments also show that they perform better than self-labeling which is equivalent to EM.

4.4 Loss Function in GWS

Following the empirical results presented by Dehghani et al. [7] and the theoretical results presented by Zamani and Croft [51], we use a pairwise loss function for optimizing GWS models. However, as is also shown in the experiments, we observed that existing loss functions are not sufficiently effective for GWS optimization. Because the initial weak labeler is imperfect, the poor performance of the weak labeler on some queries is inevitable. If we assume all queries have the same importance through our training process, the poorly performing queries are expected to have a negative impact on the final performance. To keep up the quality of initial weak supervision data, we assign a weight to each query based on its estimated ranking performance and integrate it into our optimization.

Assume for each query q, the corresponding importance is w_q. We can use in-batch re-weighting to normalize the importance of each training instance. For each training batch $B = \{(q_1, d_{q_1,1}, d_{q_1,2}), (q_2, d_{q_2,1}, d_{q_2,2}), \cdots, q_{|B|}, d_{q_{|B|},1}, d_{q_{|B|},2})\}$, our loss function is defined as follows:

$$
\begin{aligned}
l(B) &= \sum_{i=1}^{|B|} l(q_i, d_{q_i,1}, d_{q_i,2}; M_\theta, M') \\
&= \sum_{i=1}^{|B|} \frac{w_{q_i}}{\sum_{j=1}^{|B|} w_{q_j}} l_{\text{hinge}}(q_i, d_{q_i,1}, d_{q_i,2}; M_\theta, M') \\
&= \sum_{i=1}^{|B|} \frac{w_{q_i}}{\sum_{j=1}^{|B|} w_{q_j}} \max\left(0, \epsilon - \text{sign}(M'(q_i, d_{q_i,1}) - M'(q_i, d_{q_i,2}))(M_\theta(q_i, d_{q_i,1}) - M_\theta(q_i, d_{q_i,2}))\right)
\end{aligned}
$$

where $l_{\text{hinge}}(q_i, d_{q_i,1}, d_{q_i,2}; M_\theta, M')$ is the hinge loss for the pairwise training instance $(q_i, d_{q_i,1}, d_{q_i,2})$ and the ranking model M_θ. The labels come from the weak labeler M'. In hinge loss, ϵ is a margin. We set $\epsilon = 1$ in our experiment.

For estimating query weights, we rely on query performance prediction (QPP). The goal of QPP is to predict a retrieval model's quality for a given query when neither explicit nor implicit relevance information is available [6]. Thus, we can leverage unsupervised QPP models for estimating the quality of a ranked list produced by the weak labeler during training and filter out noisy data in the weak supervision signal.

Among all the available QPP methods, we choose Normalized Query Commitment (NQC) [39] as our QPP estimator because of its robust performance and simplicity. That being said, the choice of QPP method is orthogonal to the GWS optimization process and it can be replaced by any other QPP method. NQC estimates the retrieval performance by computing the normalized standard deviation of the retrieval scores assigned to the top retrieved documents. The formula is as follows:

$$
NQC(q; C, M') = \frac{\sqrt{\frac{1}{n} \sum_{d \in \pi^k_{M'}(q;C)} (\text{score}(q, d) - \mu)^2}}{\text{score}(q, C)},
$$

Table 2. Statistic of WikiPassageQA and ANTIQUE datasets.

	ANTIQUE	WikiPassageQA	NQ	MSMARCO
# training queries	2,466	3332	132803	808731
# validation queries	-	417	-	-
# testing queries	200	416	3452	6980
# training docs	27,422	194314	18060996	8841823
# validation docs	2,466	25841	-	-
# testing docs	6589	23981	2681468	8841823
# label 3	13067	-	-	-
# label 2	9276	-	-	-
# label 1	8754	6260	132803	532761
# label 0	2914	-	-	-

where $\pi^k_{M'}(q;C)$ is the top k documents retrieved by the retrieval model M' (which is the weak labeler in our case) in response to query q. μ is the average of the scores in $\pi^k_{M'}(q;C)$. score(q,C) concatenates all documents in the collection and computes the relevance score. In this work, we directly adopt $NQC(q;C,M')$ to estimate w_q. For the ranking models based on pre-train language models, we cannot compute score(q,C), so we ignore this normalization term for them in the experiment, and it does not affect our computation for $l(B)$.

For optimization, we adopt the batch stochastic gradient descent algorithm. For each batch, we compute the average loss over all document pairs in the batch and update the parameters.

5 EXPERIMENT

In this section, we introduce the experiments and discuss the results. We describe the four datasets we used, explain the evaluation metrics, show the details of our experimental setup, and discuss the results and additional analysis.

5.1 Data

In our experiments, we use four datasets for evaluation. The first one is **ANTIQUE**, which is a dataset for non-factoid questions, created by Hashemi et al. [12] based on Yahoo! Webscope L6. Relevance annotations are collected through crowdsourcing based on the standard pooling technique. Relevance labels are between 0 and 3. The second dataset is **WikiPassageQA** [4], which is a passage retrieval dataset from Wikipedia articles for questions generated through crowdsourcing. WikiPassageQA provides binary relevance labels. The third dataset is **NQ**, which is an open domain question answering dataset. The question set is from Google Search engine, and annotators find corresponding answers on Wikipedia pages. We use the version from BEIR[1] [41] to run our experiment. The last dataset is **MSMARCO**, which is a passage retrieval dataset commonly used for research works of neural retrieval. The statistics of all datasets are reported in Table 2. The scale of NQ and MSMARCO is much larger than WikiPassageQA and ANTIQUE, so we will conduct retrieval tasks for all four datasets to show that GWS can deal with small and large-scale settings, and do additional analysis on WikiPassageQA and ANTIQUE to better demonstrate GWS.

Note that, given the focus of this paper on weak supervision, none of the relevance judgments are used for training.

[1]https://public.ukp.informatik.tu-darmstadt.de/thakur/BEIR/datasets/

5.2 Evaluation Metrics

To evaluate retrieval effectiveness, we report four standard evaluation metrics: (1, 2) normalized discounted cumulative gain (NDCG) at two ranking cut-offs 1 and 10. NDCG is a standard metric that considers graded relevance labels. (3) Mean reciprocal rank (MRR) that measures the reciprocal rank of the first relevant retrieved document, and (4) mean average precision (MAP) that is a standard recall-oriented metric introduced by TREC. We only consider the documents in the re-ranking scope for measuring MAP; We do not include Precision@k and Recall@k because we focused on reranking tasks on top-k (top 20 in most cases) results from the weak labeler in this paper. In the reranking setting, the model does not consider documents not in top-k results and only enhances the quality of top results, so Precision@k and Recall@k are not sensitive and fail to show the improvement the model actually makes. As mentioned earlier, ANTIQUE provides four-level graded relevance annotation, while the last two metrics (MRR and MAP) only take binary labels. To convert graded relevance labels to binary labels, we followed the instructions provided by the ANTIQUE dataset: labels 0 and 1 are non-relevant, and labels 2 and 3 are relevant.

Statistically significant differences in metric values are determined using the two-tailed paired t-test with Bonferroni correction and 95% confidence interval ($p\_value < 0.05$).

5.3 Experimental Setup

GWS is a framework which is compatible with any ranking architectures and initial weak labelers. In this part, we describe the actual experimental setup of GWS. Following Dehghani et al. [7], we choose BM25 as the initial weak labeler, which has shown robust and strong performance across collections. In our experiments, we use Anserini's implementation of BM25 [49]. For the ranking architecture, we choose two pre-trained language models, BERT [8] and RoBERTa [22]. Recently, fine-tuning BERT for ranking tasks has received notable attention [11, 28]. Compared to a neural ranking model trained from scratch, BERT and other language models improve the ranking performance significantly. Besides, fine-tuning a pre-trained language model also decreases the required volume of weak supervision data.

For the input of BERT, we concatenate a query and a document with a `[SEP]` token to compute their relevance. For the text-matching task, the pooled output of BERT (the encoding of [CLS] token) would be fed into a feed-forward network to compute a matching score. For the score, we can compute the loss function for ranking and fine-tune the parameters according to the loss. Fine-tuning BERT and optimizing the final feed-forward network with the ranking loss function is a general method to apply BERT for learning to rank. RoBERTa has the same usage as BERT.

All ranking models are implemented by PyTorch [29] and the HuggingFace Transformer library [45]. For the pre-trained language models used in our experiments, we used the checkpoints for BERT-base [8] and RoBERTa-base [22] implementations of HuggingFace. For optimization, we adopt the AdamW optimizer [23] with the initial learning rate of 5×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and weight decay of 0.01. we set the batch size as 16, and the total training steps as 10000.

For teacher/student model selection, we rely on the performance of a held-out validation set. For WikiPassageQA, we use the original development set for validation. However, other datasets do not have an explicit validation set. Thus, we randomly select 300 training queries as the validation set. We check the performance on the validation set every 1000 steps and use the best one as the final model. The validation sets are used for all our models and all the baseline methods.

For re-ranking tasks, we need to decide the number of documents to be considered for re-ranking. WikiPassageQA provides an explicit set of documents to be re-ranked for each query. For the other three datasets, we re-rank the top 20 documents retrieved by BM25. The same setting is used for all methods, including baselines. To create a weakly supervised dataset for training, we created 20

Table 3. The retrieval performance obtained by GWS and the baselines on WikiPassageQA and ANTIQUE datasets. The superscripts * respectively denote that the improvements over the weakly supervised models are statistically significant. The highest value in each column of the table is marked in bold.

Model	NDCG@1	NDCG@10	MAP	MRR
WikiPassageQA				
Baselines				
BM25 (initial weak labeler)	0.4087	0.5374	0.4685	0.5479
BERT - Zero Shot	0.0337	0.1102	0.1115	0.1146
RoBERTa - Zero Shot	0.0601	0.1485	0.1373	0.1534
BERT - WS	0.4928	0.6345	0.5574	0.6379
RoBERTa - WS	0.5000	0.6316	0.5879	0.6692
GWS with Self-Labeling				
BERT - Self	0.5553*	0.6895*	0.6116*	0.6938*
RoBERTa - Self	0.6058*	0.7310*	0.6588*	0.7413*
GWS with Cross-Labeling				
BERT - Cross	0.5745*	0.7052*	0.6307*	0.7097*
RoBERTa - Cross	0.5673*	0.7007*	0.6248*	0.7042*
GWS with JCS				
BERT - JCS	**0.6611***	0.7669*	**0.6995***	**0.7774***
RoBERTa - JCS	0.6394*	0.7519*	0.6836*	0.7653*
GWS with Greedy Multi-Labeling				
BERT - Multi	0.6490*	0.7492*	0.6805*	0.7683*
RoBERTa - Multi	0.6394*	**0.7685***	0.6787*	0.7630*
ANTIQUE				
Baselines				
BM25 (initial weak labeler)	0.4417	0.3675	0.1540	0.5277
BERT - Zero Shot	0.3867	0.3591	0.1494	0.4818
RoBERTa - Zero Shot	0.2783	0.2727	0.1123	0.3797
BERT - WS	0.4967	0.3981	0.1753	0.5794
RoBERTa - WS	0.4617	0.3776	0.1652	0.5706
GWS with Self-Labeling				
BERT - Self	0.5383*	0.4202*	0.1863*	0.6300*
RoBERTa - Self	0.5917*	0.4270*	0.1923*	0.6648*
GWS with Cross-Labeling				
BERT - Cross	0.5717*	0.4285*	0.1930*	0.6446*
RoBERTa - Cross	0.5833*	0.4246*	0.1941*	0.6645*
GWS with JCS				
BERT - JCS	0.5833*	0.4303*	0.1887*	0.6488*
RoBERTa - JCS	0.6067*	0.4270*	0.1936*	0.6745*
GWS with Greedy Multi-Labeling				
BERT - Multi	0.5867*	**0.4337***	0.1942*	0.6509*
RoBERTa - Multi	**0.6250***	0.4327*	**0.1957***	**0.6851***

pairs of documents per query. Unlike random sampling on arbitrary pairs, we adopt a policy that regards only the top half passages in the list as positive and the other half as negative samples. We randomly pick one passage from both sets to build a training pair.

Table 4. The retrieval performance obtained by GWS and the baselines on Natural Questions (NQ) and MSMARCO datasets. The superscripts * respectively denote that the improvements over the weakly supervised models are statistically significant. The highest value in each column of the table is marked in bold.

Model	NDCG@1	NDCG@10	MAP	MRR
NQ				
Baselines				
BM25 (initial weak labeler)	0.1648	0.3055	0.2586	0.2748
BERT - Zero Shot	0.0159	0.0813	0.0749	0.0792
RoBERTa - Zero Shot	0.0313	0.1364	0.1064	0.1141
BERT - WS	0.2094	0.3615	0.3043	0.3244
RoBERTa - WS	0.2381	0.3782	0.3239	0.3463
GWS with Self-Labeling				
BERT - Self	0.2210*	0.3712*	0.3134*	0.3343*
RoBERTa - Self	0.2668*	0.3988*	0.3471*	0.3710*
GWS with Cross-Labeling				
BERT - Cross	0.2683*	0.4048*	0.3527*	0.3748*
RoBERTa - Cross	0.2735*	**0.4110***	**0.3587***	**0.3819***
GWS with JCS				
BERT - JCS	0.2590*	0.3925*	0.3411*	0.3635
RoBERTa - JCS	0.2636*	0.4010*	0.3472*	0.3708*
GWS with Greedy Multi-Labeling				
BERT - Multi	0.2642*	0.3967*	0.3453*	0.3671*
RoBERTa - Multi	**0.2801***	0.4083*	0.3569*	0.3806*
MSMARCO				
Baselines				
BM25 (initial weak labeler)	0.1043	0.2341	0.1902	0.1940
BERT - Zero Shot	0.0334	0.1516	0.1094	0.1117
RoBERTa - Zero Shot	0.0271	0.1132	0.0894	0.0911
BERT - WS	0.1080	0.2455	0.1980	0.2018
RoBERTa - WS	0.1110	0.2492	0.2009	0.2047
GWS with Self-Labeling				
BERT - Self	0.1017	0.2402	0.1919	0.1953
RoBERTa - Self	0.1093	0.2497	0.2004	0.2044
GWS with Cross-Labeling				
BERT - Cross	0.1097	0.2410	0.1961	0.1994
RoBERTa - Cross	**0.1119**	0.2512	0.2021	0.2059
GWS with JCS				
BERT - JCS	0.1090	0.2473	0.1993	0.2027
RoBERTa - JCS	0.1039	0.2460	0.1971	0.2002
GWS with Greedy Multi-Labeling				
BERT - Multi	0.1148	0.2458	0.2010	0.2043
RoBERTa - Multi	0.1116	**0.2519**	**0.2029**	**0.2062**

5.4 Results and Discussion

In this section, we report and discuss the results obtained from GWS models and the baselines on four datasets.

Baseline Results. We compare GWS with three sets of baselines: (1) the initial weak labeler (i.e., Anserini's BM25), (2) the BERT and RoBERTa ranking models under the zero-shot setting, and (3) the BERT and RoBERTa models fine-tuned using the original weak supervision approach of Dehghani et al. [7]. Table 3 and Table 4 present the results for the baselines and the models trained via GWS.

As was also discovered by other researchers [31], large language models, such as BERT and RoBERTa, have a poor zero-shot retrieval performance; thus, fine-tuning them with a retrieval objective is necessary. Zero-shot retrieval has meaningful results on ANTIQUE because we focus on only the top-20 re-ranking from BM25 results, and most of them have relevance scores from 1-3, contributing to evaluation metrics.

That being said, once these models are fine-tuned using weakly supervised data (i.e., BERT - WS and RoBERTa - WS), they substantially outperform their weak labeler (i.e., BM25) without any manually labeled data. For example, BERT - WS outperforms BM25 by 18%, by 8% and by 18% in terms of NDCG@10 on WikiPassageQA, ANTIQUE and NQ datasets, respectively. This once again confirms the power of weak supervision training for neural ranking models that was originally discovered by Dehghani et al. [7]. In the weak supervision setting, there is no clear winner between BERT and RoBERTa models; RoBERTa performs better on WikiPassageQA, especially in terms of MAP and MRR, while BERT outperforms RoBERTa on ANTIQUE with respect to all metrics. However, seeing the results of MSMARCO, the improvement between BM25 and WS is small and not significant in all metrics. Considering the performance of BM25 on MSMARCO is relatively low, we think the low quality of weak supervision signals limits the effectiveness of WS. We will show that it also restricts GWS in the following discussion.

GWS with Self-Labeling Results. Results on WikiPassageQA, ANTIQUE and NQ datasets confirm that GWS with Self-Lebeling outperforms all the baselines, including weakly supervised models. For example, a BERT model that is initially trained on the BM25's weak labels and then uses the proposed self-labeling and iterative training strategy achieves 8%, 5.5% and 2.6% higher NDCG@10 values than BERT - WS. These improvements, for both BERT and RoBERTa models, are statistically significant. Therefore, *we can conclude that GWS with Self-Labeling leads to retrieval performance improvements in all cases if weak supervision works on the weak labeler.* It is notable that the most impacted evaluation metrics by self-labeling are NDCG@1 and MRR. This suggests that self-labeling most impacts the model's behavior in identifying the first relevant document at top positions. These metrics are often important in non-factoid question answering tasks. Interestingly, RoBERTa benefits more from self-labeling; RoBERTa - Self outperforms RoBERTa - WS by 16% and 13% in terms of NDCG@10 on WikiPassageQA and ANTIQUE datasets, respectively. Obtaining such substantial improvements without using labeled training data is the first evidence of the potential impacts of GWS. On NQ dataset, the improvement is only 5.4%, which is relatively not effective. For MSMARCO, due to the ineffectiveness of WS, we cannot observe any improvement in self-labeling with GWS.

To better understand the behavior of GWS with self-labeling, we plot a curve of ranking performance at each re-labeling iteration for two datasets with significant improvement. The results are depicted in Figure 2. Note that the results for iteration 0 come from the initial weak labeler, BM25 in our experiment. The results for iteration 1 are equivalent to results obtained by the original weak supervision approach. In WikiPassageQA, we observe that the ranking performance generally

increases through the iterations. Although the curve sometimes drops, both models reach the best performance after iteration 6 on all metrics, which highlights the importance of iterative re-labeling. Overall, both the BERT and RoBERTa curves follow a similar trend on WikiPassageQA. Results in ANTIQUE are different. Both the BERT and RoBERTa reach their best performance in the early iterations. The performance curves for BERT remains stable in the following iterations, however, RoBERTa observes a substantial performance drop in late iterations. That being said, RoBERTa, at its best-performing iteration, outperforms BERT. Besides the importance of self-labeling, these plots suggest that the iterative optimization behavior in GWS is dataset-dependent, and sometimes an early stopping approach is needed. Therefore, a validation set for determining the best-performing iteration may play a vital role.

GWS with Cross-Labeling Results. From Table 3, we observe that GWS with cross-labeling significantly outperforms all the baselines. Compared to self-labeling, cross-labeling does not provide a consistent improvement. For example, BERT with cross-labeling outperforms BERT with self-labeling on WikiPassageQA, ANTIQUE and NQ datasets, however, this is not the case for RoBERTa. RoBERTa learns better from self-labeling for WikiPassageQA, and both self-labeling and cross-labeling strategies have a comparable impact on RoBERTa for the ANTIQUE datasets. One reason may be that RoBERTa plays a better role as a teacher model, thus whenever it's a teacher, either as a RoBERTa - Self or BERT - Cross, it leads to superior performance. Also, because both BERT and RoBERTa with weak supervision and self-labeling bring limited impact for MSMARCO, the multi-model setting also does not effectively improve ranking performance on MSMARCO. We consider that the quality of weak labeler is a key factor in deciding the effectiveness of GWS based on MSMARCO's results. Because MSMARCO does not show significant changes in the following two settings either, we will discuss only the other three datasets later.

GWS with Joint Cross- and Self-Labeling Results. Joint Cross- and Self-Lebeling (JCS) demonstrates a successful performance compared to the previous implementations of GWS on WikiPassageQA and ANTIQUE.The improvements brought by JCS are higher in WikiPassageQA. There is no clear winner between BERT - JCS and RoBERTa - JCS; BERT - JCS performs better on WikiPassageQA, while RoBERTa - JCS does well on the ANTIQUE dataset. However, cross-labeling still performs better on NQ, which means it is hard to select the best implementation for GWS.

GWS with Greedy Multi-Labeling Results. The results obtained by the Greedy Multi-Labeling approach are consistent with JCS. This approach performs better than each of the self-labeling and cross-labeling approaches on ANTIQUE and WikiPassageQA, but not better on NQ.

In fact, there is no one absolutely best-performing GWS approach. On WikiPassageQA, our best-performing model outperforms the initial weak labeler (BM25) by 56% and 43% in terms of NDCG@1 and NDCG@10, respectively. On ANTIQUE, the improvements are slightly smaller; our best-performing model respectively outperforms the initial weak labeler by 41% and 18% in terms of NDCG@1 and NDCG@10. On NQ, our best-performing model respectively outperforms the initial weak labeler by 70% and 34%, which are also very significant numbers.

The Impact of Query Importance Weighting on GWS.. In Table 5, we report the results with and without query importance weighting for WikiPassageQA and ANTIQUE. We only focus on self-labeling approach, however, our observations generalize to other GWS re-labeling approaches too. According to the table, query importance weighting using NQC always leads to statistically significant improvements. It helps GWS to focus on more effective examples through weak supervision and query importance weighting is a crucial part of GWS optimization. Future work can explore the impact of various QPP approaches on GWS performance.

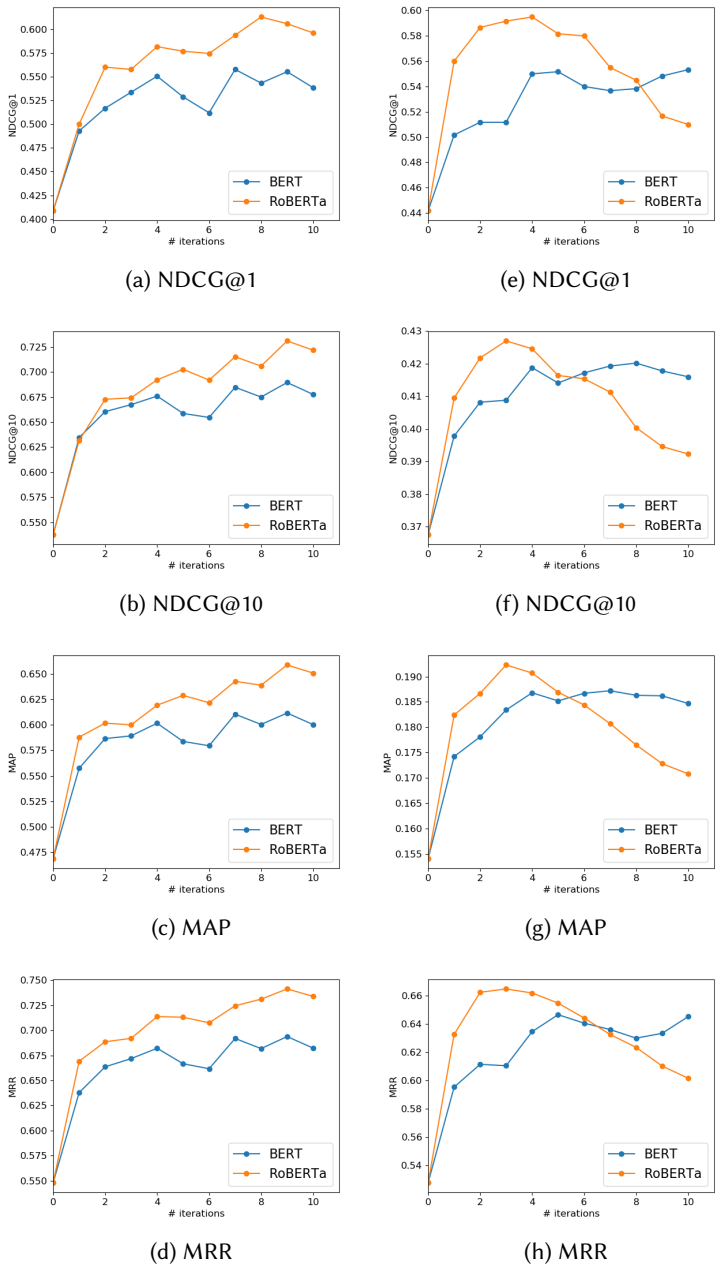
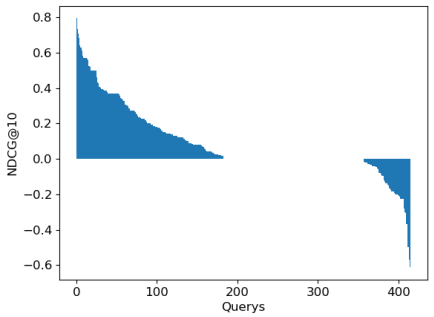


Fig. 2. The retrieval performance obtained by GWS with self-labeling at different iterations. Results are presented on both WikiPassageQA ((a)-(d)) and ANTIQUE ((e)-(h)) datasets. Iteration 0 denotes the weak labeler's performance.

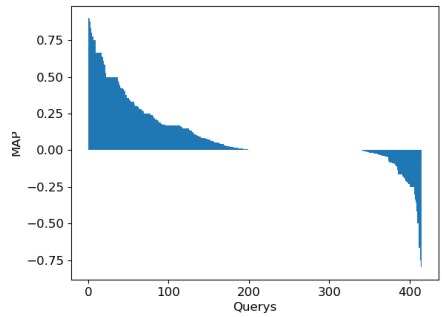
Query-level Performance Analysis. For a deeper understanding of GWS performance, in this experiment, we focus on query-level performance differences achieved by GWS. In more

Table 5. The impact of query importance weighting in GWS training on the retrieval performance. The superscript * denotes that the improvements obtained by query importance weighting are statistically significant.

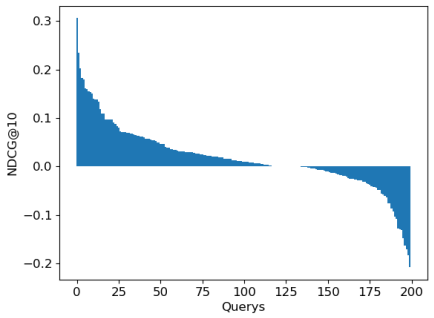
Model	NDCG@1	NDCG@10	MAP	MRR
WikiPassageQA				
BERT - Self w/o NQC	0.5337	0.6637	0.5910	0.6785
BERT - Self	0.5553	0.6895	0.6116	0.6938
RoBERTa - Self w/o NQC	0.5889	0.6889	0.6183	0.7099
RoBERTa - Self	0.6058	0.7310	0.6588	0.7413
ANTIQUE				
BERT - Self w/o NQC	0.5300	0.4066	0.1863	0.6188
BERT - Self	0.5383	0.4202	0.1863	0.6300
RoBERTa - Self w/o NQC	0.5433	0.4059	0.1867	0.6306
RoBERTa - Self	0.5917	0.4270	0.1923	0.6648



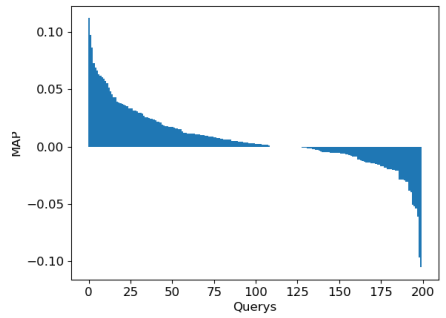
(a) NDCG@10 on WikiPassageQA



(b) MAP on WikiPassageQA



(c) NDCG@10 on ANTIQUE



(d) MAP on ANTIQUE

Fig. 3. The difference of ranking performance between RoBERTa - GWS (self) and RoBERTa - WS over all queries in terms of NDCG@10 and MAP on WikiPassageQA((a)-(b)) and ANTIQUE((c)-(d)) .

detail, we focus on the RoBERTa ranking model training using GWS with self-labeling and plot its

Table 6. Results fine-tuned on small manually labeled data. GWS is the best model selected from Table 3 and 'From scratch' refers to fine-tuning a pre-trained RoBERTa model.

WikiPassageQA				
Model	**NDCG@1**	**NDCG@10**	**MAP**	**MRR**
GWS	0.6394	0.7685	0.6787	0.7630
From scratch	0.5913	0.7097	0.6384	0.7203
GWS+fine-tuning	**0.6875**	**0.7812**	**0.7103**	**0.7897**
ANTIQUE				
GWS	**0.6250**	**0.4327**	**0.1957**	**0.6851**
From scratch	0.3517	0.3337	0.1458	0.4910
GWS+fine-tuning	0.4250	0.3595	0.1655	0.5548

performance difference with RoBERTa - WS in Figure 3. Due to the smoothness of metrics, we only plot NDCG@10 and MAP for a clear demonstration.

Regarding 0.01 as a bound for a notable amount of change, 46.1% and 43.9% of the queries are improved over WS in terms of NDCG@10 and MAP for WikiPassageQA, respectively. Considering the proportion of the degraded queries, 16.8% and 14.1%, the cases enhanced by GWS are more than the deteriorated cases. For ANTIQUE, 35% and 50.5% of the queries are respectively improved over WS in terms of NDCG@10 and MAP, with 19.5% and 25% for deteriorated queries. These plots show that the average improvements obtained by GWS are not dominated by drastic increases in a few queries.

Comparison to the model trained on small Data. We additionally compare the final model of GWS to the models trained on small data of human labels. We aim to understand if large weak signals could outperform small real data. Besides, we also check if GWS could help to build a better initial model if we aim to conduct fine-tuning on small real data.

We take 10% queries and their relevance judgment as the representative of small data from WikiPassageQA and ANTIQUE. In the analysis, we include three model: the best RoBERTa from GWS, RoBERTa trained on the small data, and the best GWS model fine-tuned on the small data. The result is shown in Table 6. We can observe that for both dataset, GWS outperforms the model trained on the real data if the data is not sufficient. Actually, the model trained on insufficient data has very weak performance on both datasets. Considering fine-tuning on WikiPassageQA, GWS with small real data can further improve the ranking performance. On ANTIQUE, fine-tuning on small data even decreases the ranking performance.

6 CONCLUSION AND FUTURE WORK

In this work, we proposed generalized weak supervision (GWS), a generic framework for training retrieval models without requiring any manually labeled training data. Based on weak supervision, which automatically produces training data using existing retrieval models, we generalized the definition of weak labeler to include the weakly supervised models themselves. We provided four implementations of the GWS framework: self-labeling, cross-labeling, joint cross- and self-labeling (JCS), and greedy multi-labeling. We also presented the theoretical relationship between GWS and the Expectation-Maximization algorithm. Besides, we provided a query importance weighting based on query performance prediction for effective training of GWS models.

In the experiment, we evaluated GWS on four datasets. Our experiments showed that GWS achieves substantial improvements compared to weak supervision if weak signals are sufficiently

reliable. We observed larger improvements when the power of multi-model setting was applied. Furthermore, we showed that query selection via an unsupervised query performance predictor can have a significant impact on GWS performance. Our analysis suggested that a large portion of test queries benefit from GWS training.

For future work, we aim to theoretically analyze how GWS affects the training of neural ranking models. Besides, we intend to extend the GWS framework by leveraging multiple weak labelers as well as multiple query performance predictors in order to minimize the noise introduced by the weak labels.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, and in part by an Alexa Prize grant from Amazon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Minmin Chen, Kilian Q. Weinberger, and John Blitzer. 2011. Co-Training for Domain Adaptation. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011*. 2456–2464.

[2] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

[3] Daniel Cohen, Bhaskar Mitra, Katja Hofmann, and W. Bruce Croft. 2018. Cross Domain Regularization for Neural Ranking Models using Adversarial Learning. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 1025–1028.

[4] Daniel Cohen, Liu Yang, and W. Bruce Croft. 2018. WikiPassageQA: A Benchmark Collection for Research on Non-factoid Answer Passage Retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 1165–1168.

[5] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting Query Performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02)*. Association for Computing Machinery, New York, NY, USA, 299–306. https://doi.org/10.1145/564376.564429

[6] Stephen Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*. ACM, 299–306.

[7] 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*. ACM, 65–74.

[8] 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*. Association for Computational Linguistics, 4171–4186.

[9] Dehong Gao, Linbo Jin, Ben Chen, Minghui Qiu, Peng Li, Yi Wei, Yi Hu, and Hao Wang. 2020. FashionBERT: Text and Image Matching with Adaptive Loss for Cross-modal Retrieval. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020*. ACM, 2251–2260.

[10] 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*. ACM, 55–64.

[11] Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2020. Learning-to-Rank with BERT in TF-Ranking. *CoRR* abs/2004.08476 (2020). https://arxiv.org/abs/2004.08476

[12] Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W. Bruce Croft. 2020. ANTIQUE: A Non-factoid Question Answering Benchmark. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020*. Springer, 166–173.

[13] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2020. Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Association for Computing Machinery, New York,

NY, USA, 1131–1140.

[14] Djoerd Hiemstra, Stephen Robertson, and Hugo Zaragoza. 2004. Parsimonious Language Models for Information Retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '04)*. Association for Computing Machinery, New York, NY, USA, 178–185. https://doi.org/10.1145/1008992.1009025

[15] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666* (2020).

[16] Peng Hu, Liangli Zhen, Dezhong Peng, and Pei Liu. 2019. Scalable Deep Multimodal Learning for Cross-Modal Retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019*. ACM, 635–644.

[17] 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*. ACM, 2333–2338.

[18] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 6769–6781.

[19] 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, Virtual Event, China, July 25-30, 2020*. ACM, 39–48.

[20] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics* (2019).

[21] Pengfei Li, Mark Sanderson, Mark J. Carman, and Falk Scholer. 2020. Self-labeling methods for unsupervised transfer ranking. *Inf. Sci.* 516 (2020), 293–315.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

[23] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[24] Yosi Mass and Haggai Roitman. 2020. Ad-hoc Document Retrieval using Weak-Supervision with BERT and GPT2. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4191–4197.

[25] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017*. ACM, 1291–1299.

[26] Yifan Nie, Alessandro Sordoni, and Jian-Yun Nie. 2018. Multi-level Abstraction Convolutional Model with Weak Supervision for Information Retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018*. ACM, 985–988.

[27] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Mach. Learn.* 39, 2/3 (2000), 103–134.

[28] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). http://arxiv.org/abs/1901.04085

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. 8024–8035.

[30] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*. ACM, 275–281.

[31] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. *CoRR* abs/1904.07531 (2019).

[32] Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with History Answer Embedding for Conversational Question Answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019*. ACM, 1133–1136.

[33] Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W. Bruce Croft, and Mohit Iyyer. 2019. Attentive History Selection for Conversational Question Answering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019*. ACM, 1391–1400.

[34] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 2825–2835.

[35] Stephen Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *J. Documentation* 60, 5 (2004), 503–520.

[36] Stephen E. Robertson and Steve Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval., 3-6 July 1994 (Special Issue of the SIGIR Forum)*. ACM/Springer, 232–241.

[37] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994 (NIST Special Publication)*, Vol. 500-225. NIST, 109–126.

[38] 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*. ACM, 101–110.

[39] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting Query Performance by Query-Drift Estimation. *ACM Trans. Inf. Syst.* 30, 2 (2012), 11:1–11:35.

[40] Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu, and Paul Bennett. 2021. Few-Shot Text Ranking with Meta Adapted Synthetic Weak Supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*. Association for Computational Linguistics, 5030–5043.

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

[42] Amir Vakili Tahami, Kamyar Ghajar, and Azadeh Shakery. 2020. Distilling knowledge for fast retrieval-based chat-bots. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2081–2084.

[43] Nikos Voskarides, Edgar Meij, Ridho Reinanda, Abhinav Khaitan, Miles Osborne, Giorgio Stefanoni, Prabhanjan Kambadur, and Maarten de Rijke. 2018. Weakly-supervised Contextualization of Knowledge Graph Facts. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018*. ACM, 765–774.

[44] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*. Association for Computational Linguistics, 2345–2360.

[45] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. [n. d.]. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*. Association for Computational Linguistics, 38–45.

[46] 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 *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[47] Peng Xu, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Passage Ranking with Weak Supervsion. *CoRR* abs/1905.05910 (2019). arXiv:1905.05910 http://arxiv.org/abs/1905.05910

[48] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018*. ACM, 245–254.

[49] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible Ranking Baselines Using Lucene. *ACM J. Data Inf. Qual.* 10, 4 (2018), 16:1–16:20.

[50] 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*. ACM, 505–514.

[51] 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*. ACM, 147–154.

[52] 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 and Development in Information Retrieval, SIGIR 2018*. ACM, 105–114.

[53] Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik G. Learned-Miller, and Jaap Kamps. 2018. From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*. ACM, 497–506.

[54] Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum Learning for Dense Retrieval Distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 1979–1983. https://doi.org/10.1145/3477495.3531791

[55] Chengxiang Zhai and John Lafferty. 2001. Model-Based Feedback in the Language Modeling Approach to Information Retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM '01)*. Association for Computing Machinery, New York, NY, USA, 403–410. https://doi.org/10.1145/502585.502654

[56] Jingtao Zhan, Qingyao Ai, Yiqun Liu, Jiaxin Mao, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Disentangled Modeling of Domain and Relevance for Adaptable Dense Retrieval. *CoRR* abs/2208.05753 (2022).

[57] Kaitao Zhang, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2020. Selective Weak Supervision for Neural Information Retrieval. In *WWW '20: The Web Conference 2020*. ACM / IW3C2, 474–485.