



UvA-DARE (Digital Academic Repository)

Learning to Rank from Samples of Variable Quality

Dehghani, M.; Kamps, J.

Publication date

2018

Document Version

Author accepted manuscript

Published in

The First International Workshop on Learning from Limited or Noisy Data for Information Retrieval

[Link to publication](#)

Citation for published version (APA):

Dehghani, M., & Kamps, J. (2018). Learning to Rank from Samples of Variable Quality. In *The First International Workshop on Learning from Limited or Noisy Data for Information Retrieval: July 12th, 2018, Ann Arbor, Michigan, USA, co-located with SIGIR 2018. Accepted papers* arXiv.org. <https://arxiv.org/abs/1806.08694>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Learning to Rank from Samples of Variable Quality

Mostafa Dehghani
University of Amsterdam
dehghani@uva.nl

Jaap Kamps
University of Amsterdam
kamps@uva.nl

ABSTRACT

Training deep neural networks requires many training samples, but in practice training labels are expensive to obtain and may be of varying quality, as some may be from trusted expert labelers while others might be from heuristics or other sources of weak supervision such as crowd-sourcing. This creates a fundamental quality-versus-quantity trade-off in the learning process. Do we learn from the small amount of high-quality data or the potentially large amount of weakly-labeled data? We argue that if the learner could somehow know and take the label-quality into account when learning the data representation, we could get the best of both worlds. To this end, we introduce “fidelity-weighted learning” (FWL) [9], a semi-supervised student-teacher approach for training deep neural networks using weakly-labeled data. FWL modulates the parameter updates to a *student* network (trained on the task we care about) on a per-sample basis according to the posterior confidence of its label-quality estimated by a *teacher* (who has access to the high-quality labels). Both student and teacher are learned from the data. We evaluate FWL on document ranking where we outperform state-of-the-art alternative semi-supervised methods.

1 INTRODUCTION

The success of deep neural networks to date depends strongly on the availability of labeled data which is costly and not always easy to obtain. Usually it is much easier to obtain small quantities of high-quality labeled data and large quantities of unlabeled data. The problem of how to best integrate these two different sources of information during training is an active pursuit in the field of semi-supervised learning [2]. However, for a large class of tasks it is also easy to define one or more so-called “weak annotators”, additional (albeit noisy) sources of *weak supervision* based on heuristics or “weaker”, biased classifiers trained on e.g. non-expert crowd-sourced data or data from different domains that are related. While easy and cheap to generate, it is not immediately clear if and how these additional weakly-labeled data can be used to train a stronger classifier for the task we care about. More generally, in almost all practical applications machine learning systems have to deal with data samples of variable quality. For example, in a large dataset of images only a small fraction of samples may be labeled by experts and the rest may be crowd-sourced using e.g. Amazon Mechanical Turk.

Assuming we can obtain a large set of weakly-labeled data in addition to a much smaller training set of “strong” labels, the simplest

approach is to expand the training set by including the weakly-supervised samples (all samples are equal). Alternatively, one may pretrain on the weak data and then fine-tune on observations from the true function or distribution (which we call strong data). Indeed, it has recently been shown that a small amount of expert-labeled data can be augmented in such a way by a large set of raw data, with labels coming from a heuristic function, to train a more accurate neural ranking model [6–8]. The downside is that such approaches are oblivious to the amount or source of noise in the labels. Simply speaking, they do not consider the cause of noise in the labels and only focus on the effect.

In this paper, we argue that treating weakly-labeled samples uniformly (i.e. each weak sample contributes equally to the final classifier) ignores potentially valuable information of the label quality. Instead, we propose Fidelity-Weighted Learning (FWL), a Bayesian semi-supervised approach that leverages a small amount of data with true labels to generate a larger training set with *confidence-weighted weakly-labeled samples*, which can then be used to modulate the fine-tuning process based on the fidelity (or quality) of each weak sample. By directly modeling the inaccuracies introduced by the weak annotator in this way, we can control the extent to which we make use of this additional source of weak supervision: more for confidently-labeled weak samples close to the true observed data, and less for uncertain samples further away from the observed data.

We propose a setting consisting of two main modules. One is called the student and is in charge of learning a suitable data representation and performing the main prediction task, the other is the teacher which modulates the learning process by modeling the inaccuracies in the labels. We explain our approach in much more detail in Section 2, but at a high level it works as follows (see Figure 1): We pretrain the student network on weak data to learn an initial task-dependent data representation which we pass to the teacher along with the strong data. The teacher then learns to predict the strong data, but crucially, *based on the student’s learned representation*. This then allows the teacher to generate new labeled data from unlabeled data, and in the process correct the student’s mistakes, leading to a better data representation and better final predictor.

2 FIDELITY-WEIGHTED LEARNING (FWL)

In this section, we describe our proposed FWL approach for semi-supervised learning when we have access to weak supervision (e.g. heuristics or weak annotators). We assume we are given a large set of unlabeled data samples, a heuristic labeling function called the *weak annotator*, and a small set of high-quality samples labeled by experts, called the *strong dataset*, consisting of tuples of training samples x_i and their true labels y_i , i.e. $\mathcal{D}_s = \{(x_i, y_i)\}$. We consider the latter to be observations from the true target function that we are trying to learn. We use the weak annotator to generate labels for the unlabeled samples. Generated labels are noisy due to the limited accuracy of the weak annotator. This gives us the *weak dataset* consisting of tuples of training samples x_i and their weak labels \tilde{y}_i ,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](http://permissions.acm.org).

LND4IR '18, July 12, 2018, Ann Arbor, Michigan, USA

© 2018 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

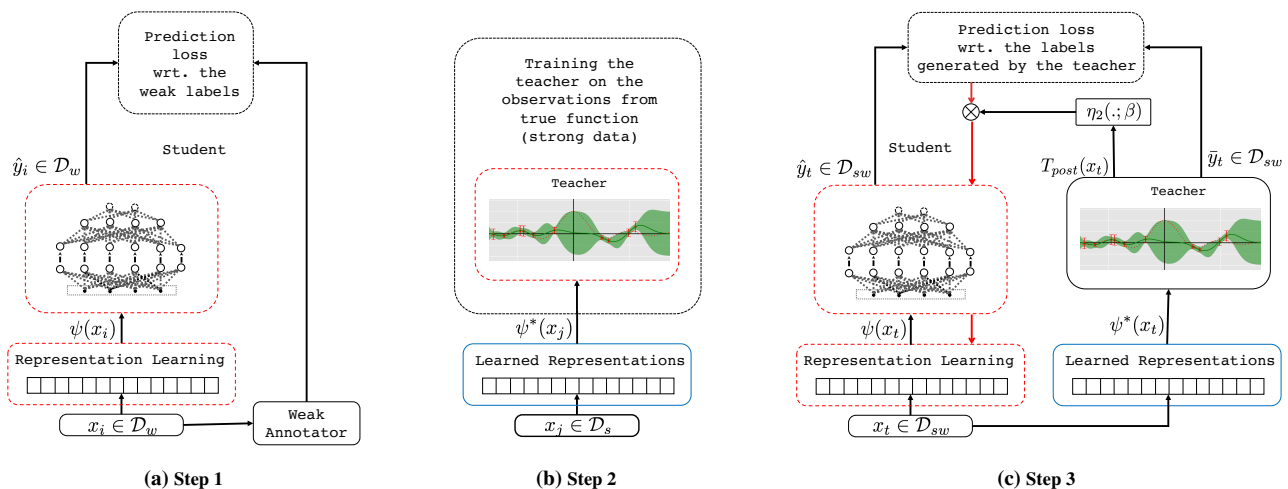


Figure 1: Illustration of Fidelity-Weighted Learning: Step 1: Pre-train student on weak data, Step 2: Fit teacher to observations from the true function, and Step 3: Fine-tune student on labels generated by teacher, taking the confidence into account. Red dotted borders and blue solid borders depict components with trainable and non-trainable parameters, respectively.

Algorithm 1 Fidelity-Weighted Learning.

- 1: Train the student on samples from the weakly-annotated data D_w .
- 2: Freeze the representation-learning component $\psi(\cdot)$ of the student and train teacher on the strong data $D_s = (\psi(x_j), y_j)$. Apply teacher to unlabeled samples x_t to obtain soft dataset $D_{sw} = \{(x_t, \tilde{y}_t)\}$ where $\tilde{y}_t = T(x_t)$ is the soft label and for each instance x_t , the uncertainty of its label, $\Sigma(x_t)$, is provided by the teacher.
- 3: Train the student on samples from D_{sw} with SGD and modulate the step-size η_t according to the per-sample quality estimated using the teacher (Equation 1).

i.e. $\mathcal{D}_w = \{(x_i, \tilde{y}_i)\}$. Note that we can generate a large amount of weak training data \mathcal{D}_w at almost no cost using the weak annotator. In contrast, we have only a limited amount of observations from the true function, i.e. $|\mathcal{D}_s| \ll |\mathcal{D}_w|$.

Our proposed setup comprises a neural network called the **student** and a Bayesian function approximator called the **teacher**. The training process consists of three phases which we summarize in Algorithm 1 and Figure 1.

Step 1 Pre-train the student on \mathcal{D}_w using weak labels generated by the weak annotator.

The main goal of this step is to learn a *task dependent* representation of the data as well as pretraining the student. The student function is a neural network consisting of two parts. The first part $\psi(\cdot)$ learns the data representation and the second part $\phi(\cdot)$ performs the prediction task (e.g. classification). Therefore the overall function is $\hat{y} = \phi(\psi(x_i))$. The student is trained on all samples of the weak dataset $\mathcal{D}_w = \{(x_i, \tilde{y}_i)\}$. For brevity, in the following, we will refer to both data sample x_i and its representation $\psi(x_i)$ by x_i when it is obvious from the context. From the self-supervised feature learning point of view, we can say that representation learning in this step is solving a surrogate task of approximating the expert knowledge, for which a noisy supervision signal is provided by the weak annotator.

Step 2 Train the teacher on the strong data $(\psi(x_j), y_j) \in \mathcal{D}_s$ represented in terms of the student representation $\psi(\cdot)$ and then use the teacher to generate a soft dataset \mathcal{D}_{sw} consisting of $\langle \text{sample}, \text{predicted label}, \text{confidence} \rangle$ for all data samples.

We use a Gaussian process as the teacher to capture the label uncertainty in terms of the student representation, estimated w.r.t the strong data. A prior mean and co-variance function is chosen for \mathcal{GP} . The learned embedding function $\psi(\cdot)$ in Step 1 is then used to map the data samples to dense vectors as input to the \mathcal{GP} . We use the learned representation by the student in the previous step to compensate lack of data in \mathcal{D}_s and the teacher can enjoy the learned knowledge from the large quantity of the weakly annotated data. This way, we also let the teacher see the data through the lens of the student.

The \mathcal{GP} is trained on the samples from \mathcal{D}_s to learn the posterior mean m_{post} (used to generate soft labels) and posterior co-variance $K_{\text{post}}(\cdot, \cdot)$ (which represents label uncertainty). We then create the soft dataset $\mathcal{D}_{sw} = \{(x_t, \tilde{y}_t)\}$ using the posterior \mathcal{GP} , input samples x_t from $\mathcal{D}_w \cup \mathcal{D}_s$, and predicted labels \tilde{y}_t with their associated uncertainties as computed by $T(x_t)$ and $\Sigma(x_t)$:

$$\begin{aligned} T(x_t) &= g(m_{\text{post}}(x_t)) \\ \Sigma(x_t) &= h(K_{\text{post}}(x_t, x_t)) \end{aligned}$$

The generated labels are called *soft labels*. Therefore, we refer to \mathcal{D}_{sw} as a soft dataset. $g(\cdot)$ transforms the output of \mathcal{GP} to the suitable output space. For example in classification tasks, $g(\cdot)$ would be the softmax function to produce probabilities that sum up to one. For multidimensional-output tasks where a vector of variances is provided by the \mathcal{GP} , the vector $K_{\text{post}}(x_t, x_t)$ is passed through an aggregating function $h(\cdot)$ to generate a scalar value for the uncertainty of each sample. Note that we train \mathcal{GP} only on the strong dataset \mathcal{D}_s but then use it to generate soft labels $\tilde{y}_t = T(x_t)$ and uncertainty $\Sigma(x_t)$ for samples belonging to $\mathcal{D}_{sw} = \mathcal{D}_w \cup \mathcal{D}_s$.

In practice, we furthermore divide the space of data into several regions and assign each region a separate \mathcal{GP} trained on samples from that region. This leads to a better exploration of the data space

and makes use of the inherent structure of data. The algorithm called clustered \mathcal{GP} gave better results compared to a single GP.

Step 3 Fine-tune the weights of the student network on the soft dataset, while modulating the magnitude of each parameter update by the corresponding teacher-confidence in its label.

The student network of Step 1 is fine-tuned using samples from the soft dataset $\mathcal{D}_{s,w} = \{(x_t, \bar{y}_t)\}$ where $\bar{y}_t = T(x_t)$. The corresponding uncertainty $\Sigma(x_t)$ of each sample is mapped to a confidence value according to Equation 1 below, and this is then used to determine the step size for each iteration of the stochastic gradient descent (SGD). So, intuitively, for data points where we have true labels, the uncertainty of the teacher is almost zero, which means we have high confidence and a large step-size for updating the parameters. However, for data points where the teacher is not confident, we down-weight the training steps of the student. This means that at these points, we keep the student function as it was trained on the weak data in Step 1.

More specifically, we update the parameters of the student by training on $\mathcal{D}_{s,w}$ using SGD:

$$\begin{aligned} \mathbf{w}^* &= \arg \min_{\mathbf{w} \in \mathcal{W}} \frac{1}{N} \sum_{(x_t, \bar{y}_t) \in \mathcal{D}_{s,w}} l(\mathbf{w}, x_t, \bar{y}_t) + \mathcal{R}(\mathbf{w}), \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \eta_t (\nabla l(\mathbf{w}, x_t, \bar{y}_t) + \nabla \mathcal{R}(\mathbf{w})) \end{aligned}$$

where $l(\cdot)$ is the per-example loss, η_t is the total learning rate, N is the size of the soft dataset $\mathcal{D}_{s,w}$, \mathbf{w} is the parameters of the student network, and $\mathcal{R}(\cdot)$ is the regularization term.

We define the total learning rate as $\eta_t = \eta_1(t)\eta_2(x_t)$, where $\eta_1(t)$ is the usual learning rate of our chosen optimization algorithm that anneals over training iterations, and $\eta_2(x_t)$ is a function of the label uncertainty $\Sigma(x_t)$ that is computed by the teacher for each data point. Multiplying these two terms gives us the total learning rate. In other words, η_2 represents the *fidelity* (quality) of the current sample, and is used to multiplicatively modulate η_1 . Note that the first term does not necessarily depend on each data point, whereas the second term does. We propose

$$\eta_2(x_t) = \exp[-\beta \Sigma(x_t)], \quad (1)$$

to exponentially decrease the learning rate for data point x_t if its corresponding soft label \bar{y}_t is unreliable (far from a true sample). In Equation 1, β is a positive scalar hyper-parameter. Intuitively, small β results in a student which listens more carefully to the teacher and copies its knowledge, while a large β makes the student pay less attention to the teacher, staying with its initial weak knowledge. More concretely speaking, as $\beta \rightarrow 0$ student places more trust in the labels \bar{y}_t estimated by the teacher and the student copies the knowledge of the teacher. On the other hand, as $\beta \rightarrow \infty$, student puts less weight on the extrapolation ability of \mathcal{GP} and the parameters of the student are not affected by the correcting information from the teacher.

3 EXPERIMENTS

In this section, we apply FWL to *document ranking*. We evaluate the performance of our method compared to the following baselines:

- (1) **WA**. The weak annotator, i.e. the unsupervised method used for annotating the unlabeled data.
- (2) **NN_W**. The student trained only on weak data.
- (3) **NN_S**. The student trained only on strong data.

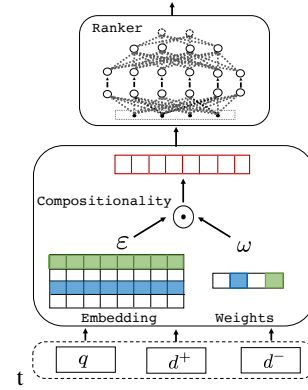


Figure 2: The student for the document ranking task.

- (4) **NN_{S+W}**. The student trained on samples that are alternately drawn from \mathcal{D}_w without replacement, and \mathcal{D}_s with replacement. Since $|\mathcal{D}_s| \ll |\mathcal{D}_w|$, it oversamples the strong data.
- (5) **NN_{W→S}**. The student trained on weak dataset \mathcal{D}_w and fine-tuned on strong dataset \mathcal{D}_s .
- (6) **FWL**. Our FWL model, i.e. the student trained on the weakly labeled data and fine-tuned on examples labeled by the teacher using the confidence scores.

3.1 Document Ranking

This task is the core information retrieval problem and is challenging as the ranking model needs to learn a representation for long documents and capture the notion of relevance between queries and documents. Furthermore, the size of publicly available datasets with query-document relevance judgments is unfortunately quite small (~ 250 queries). We employ a pairwise neural ranker architecture as the student [8]. In this model, ranking is cast as a regression task. Given each training sample x as a triple of query q , and two documents d^+ and d^- , the goal is to learn a function $\mathcal{F} : \{ \langle q, d^+, d^- \rangle \} \rightarrow \mathbb{R}$, which maps each data sample x to a scalar output value y indicating the probability of d^+ being ranked higher than d^- with respect to q .

The student follows the architecture proposed in [8]. The first layer of the network, i.e. representation learning layer $\psi : \{ \langle q, d^+, d^- \rangle \} \rightarrow \mathbb{R}^m$ maps each input sample to an m -dimensional real-valued vector. In general, besides learning embeddings for words, function ψ learns to compose word embedding based on their global importance in order to generate query/document embeddings. The representation layer is followed by a simple fully-connected feed-forward network with a sigmoidal output unit to predict the probability of ranking d^+ higher than d^- . The general schema of the student is illustrated in Figure 2.

The teacher is implemented by clustered \mathcal{GP} algorithm.

The weak annotator is BM25 [10], a well-known unsupervised method for scoring query-document pairs based on statistics of the matched terms.

Results and Discussions We conducted 3-fold cross validation on \mathcal{D}_s (the strong data) with 80/20 training/validation split, and report two standard evaluation metrics for ranking: mean average precision (MAP) of the top-ranked 1,000 documents and normalized discounted cumulative gain calculated for the top 20 retrieved documents (nDCG@20). In all of the experiments, the experimental

Table 1: Performance of FWL approach and baseline methods for ranking task. ^{▲1} indicates that the improvements with respect to the baseline i are statistically significant at the 0.05 level using the paired two-tailed t-test with Bonferroni correction.

Method	Robust04		ClueWeb09-CatB ¹	
	MAP	nDCG@20	MAP	nDCG@20
1 WA _{BM25}	0.2503 ^{▲3}	0.4102 ^{▲3}	0.1021 ^{▲3}	0.2070 ^{▲3}
2 NN _W [8]	0.2702 ^{▲13}	0.4290 ^{▲13}	0.1297 ^{▲13}	0.2201 ^{▲13}
3 NN _S	0.1790	0.3519	0.0782	0.1730
4 NN _{S+W}	0.2763 ^{▲123}	0.4330 ^{▲123}	0.1354 ^{▲123}	0.2319 ^{▲123}
5 NN _{W→S}	0.2810 ^{▲123}	0.4372 ^{▲123}	0.1346 ^{▲123}	0.2317 ^{▲123}
6 FWL	0.3124^{▲12345}	0.4607^{▲12345}	0.1472^{▲12345}	0.2453^{▲12345}

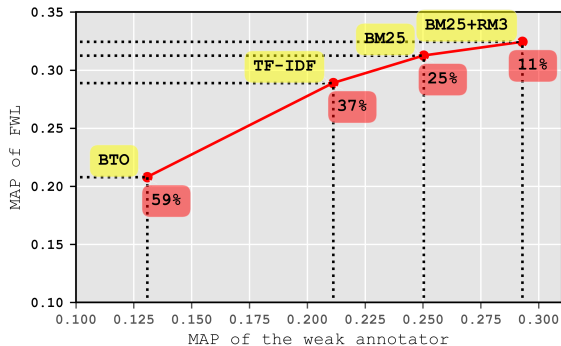


Figure 3: Performance of FWL versus the correspondence weak annotator in the document ranking task, on Robust04 dataset.

setup and preprocessing are similar to [5, 8]. Table 1 shows the performance on both datasets. As can be seen, FWL provides a significant boost on the performance over all datasets. In the ranking task, the student is designed in particular to be trained on weak annotations [8], hence training the network only on weak supervision, i.e. NN_W performs better than NN_S. This can be due to the fact that ranking is a complex task requiring many training samples, while relatively few data with true labels are available.

Alternating between strong and weak data during training, i.e. NN_{S+W} seems to bring little (but statistically significant) improvement. However, we can gain better results by the typical fine-tuning strategy, NN_{W→S}.

3.2 Sensitivity to Weak Annotation Quality

Our proposed setup in FWL requires defining a so-called “weak annotator” to provide a source of weak supervision for unlabelled data. Now, in this section, we study how the quality of the weak annotator may affect the performance of the FWL, for the task of document ranking.

To do so, besides BM25 [10], we use three other weak annotators: vector space model [11] with binary term occurrence (BTO) weighting schema and vector space model with TF-IDF weighting schema, which are both weaker than BM25, and BM25+RM3 [1] that uses pseudo-relevance feedback [4] on top of BM25, leading to better labels.

Figure 3 illustrates the performance of these four weak annotators in terms of their mean average precision (MAP) on the test data, versus the performance of FWL given the corresponding weak

¹Spam documents were filtered out using the Waterloo spam scorer (<http://plg.uwaterloo.ca/~gvcormac/clueweb09spam/>) [3] with the default threshold 70%.

annotator. As it is expected, the performance of FWL depends on the quality of the employed weak annotator. The percentage of improvement of FWL over its corresponding weak annotator on the test data is also presented in Figure 3. As can be seen, the better the performance of the weak annotator is, the less the improvement of the FWL would be.

4 CONCLUSION

Training neural networks using large amounts of weakly annotated data is an attractive approach in scenarios where an adequate amount of data with true labels is not available, a situation which often arises in practice. In this paper, we make use of fidelity-weighted learning (FWL), a new student-teacher framework for semi-supervised learning in the presence of weakly labeled data. We applied FWL to document ranking and empirically verified that FWL speeds up the training process and improves over state-of-the-art semi-supervised alternatives.

Our general conclusion is that explicitly modeling label quality is both possible and useful for learning task dependent data representations. The student-teacher configuration conceptually allows us to distinguish between the role of the student who learns the target representation, and the teacher who both learns to estimate the confidence in labels as well as adapts to the needs of the student by taking the student’s current state into account. One key observation is that the model with explicit feedback about the strong labels (i.e., pre-training and fine-tuning) is not as effective as the teacher model that implicitly gives feedback on the strong labels in terms of label confidence. While this can be explained in terms of avoiding overfitting or loss of generalization, there is also a conceptual explanation in terms of promoting the student to learn by not overruling or correcting, but by giving the student the right feedback to allow for a learning by discovery approach, retaining the full generative power of the student model. Arguably, this even resembles a kind of “Socratic dialog” between student and teacher.

Acknowledgments This research is funded in part by the Netherlands Organization for Scientific Research (NWO; ExPoSe project, NWO CI # 314.99.108).

REFERENCES

- [1] N. Abdul-jaleel, J. Allan, W. B. Croft, O. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. Umass at trec 2004: Novelty and hard. In *TREC-13*, 2004.
- [2] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2006.
- [3] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Inf. Retr.*, 14(5):441–465, 2011.
- [4] M. Dehghani, H. Azarbonyad, J. Kamps, D. Hiemstra, and M. Marx. Luhn revisited: Significant words language models. In *CIKM '16*, 2016.
- [5] M. Dehghani, S. Rothe, E. Alfonseca, and P. Fleury. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of The international Conference on Information and Knowledge Management (CIKM'17)*, 2017.
- [6] M. Dehghani, A. Severyn, S. Rothe, and J. Kamps. Learning to learn from weak supervision by full supervision. In *NIPS2017 workshop on Meta-Learning (MetaLearn 2017)*, 2017.
- [7] M. Dehghani, A. Severyn, S. Rothe, and J. Kamps. Avoiding your teacher’s mistakes: Training neural networks with controlled weak supervision. *arXiv preprint arXiv:1711.00313*, 2017.
- [8] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft. Neural ranking models with weak supervision. In *SIGIR '17*, 2017.
- [9] M. Dehghani, A. Mehrjou, S. Gouws, J. Kamps, and B. Schölkopf. Fidelity-weighted learning. In *International Conference on Learning Representations (ICLR)*, 2018.
- [10] S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.
- [11] G. Salton and C.-S. Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973.