Transferring Knowledge with Source Selection to Learn IR Functions on Unlabeled Collections

Parantapa Goswami Massih-Reza Amini Éric Gaussier
Université Joseph Fourier
Laboratoire d'Informatique de Grenoble
BP 53 - F-38041 Grenoble, France
firstname.lastname@imag.fr

ABSTRACT

We investigate the problem of learning an IR function on a collection without relevance judgements (called target collection) by transferring knowledge from a *selected* source collection with relevance judgements. To do so, we first construct, for each query in the target collection, relative relevance judgment pairs using information from the source collection closest to the query (selection and transfer steps), and then learn an IR function from the obtained pairs in the target collection (self-learning step). For the transfer step, the relevance information in the source collection is summarized as a grid that provides, for each term frequency and document frequency values of a word in a document, an empirical estimate of the relevance of the document. The self-learning step iteratively assigns pairwise preferences to documents in the target collection using the scores of the former learned function. We show the effectiveness of our approach through a series of extensive experiments on CLEF-3 and several collections from TREC used either as target or source datasets. Our experiments show the importance of selecting the source collection prior to transfer information to the target collection, and demonstrate that the proposed approach yields results consistently and significantly above state-of-the-art IR functions.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Search process*; I.2 [**Artificial Intelligence**]: Natural Language Processing - *text analysis* Learning - *Parameter learning*; H.1 [**Models and Principles**]: Miscellaneous

Keywords

Domain adaptation, knowledge transfer, source selection, learning to rank, transductive learning

CIKM '13, October 27 - November 01 2013, San Francisco, CA, USA
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.

1. INTRODUCTION

Ranking is a key component of many applications in information retrieval, like *ad hoc* retrieval, routing or collaborative filtering. Recently, learning to rank has emerged as a new research topic in ranking, and different approaches proposed under this framework aim to automatically find a combination of many diverse features using a training set. In this case, training data consist of a set of queries, a set of retrieved documents associated to each of the queries, as well as relevance judgements for all query-document pairs. Providing such relevance judgements is generally limited as it requires to assign relevance labels by passing over the lists of retrieved documents associated to training queries which is a time consuming and costly process. To alleviate this problem, researchers have looked at ways to transfer knowledge from one source dataset to a target collection (a field often referred to as cross-domain adaptation) on which to learn an IR ranking function, either in the case where few labeled information [5] or no labeled information is available on the target collection [7].

Our work fits within this latter line of research and aims at learning an IR function on a collection with no relevance judgements. The studies presented in [3] have shown that it was indeed possible to learn a good ranking IR function in this scenario. However, they have also shown that the function learned on the target domain heavily depends on the source domain used, and we focus here on a transfer method that allows one to easily select the appropriate source collection from which to transfer relevance information. The problem we face can be casted as a transfer learning problem where relevance information from a source collection is propagated to a target collection. Firstly, for each query in the target collection, the source grid to which it is the closest is first selected. Then the propagation is done by first constructing a grid associating normalized Document Frequency (DF) and Term Frequency (TF) values to a relevance score in the source collection, and then learning a transductive ranking function on the target collection.

The next section describes the problem of knowledge transfer from a source domain to a target domain for learning to rank. In Section 2 we present the grid information which forms the basis of the knowledge transfer we consider here, as well as the source selection procedure and the self-learning based algorithm. We validate our approach using CLEF-3, and various TREC collections and we show that our approach consistently and significantly improves over state-of-the-art IR functions (Section 3) Finally, we summarize the main findings of this study in Section 4.

2. TRANSFER LEARNING FOR RANKING

In the following sections, we first present the framework and then our transfer learning approach for ranking.

2.1 Transferring relevancy from a collection

In a typical transfer scenario for IR, relying on the same source collection for all the target queries prevents one from taking into account the fact that queries are usually different from each other and that different collections generally display different query types. We consider here a set of source collections $\{\mathcal{C}^{s_1}, \cdots, \mathcal{C}^{s_L}\}$ composed of a set of documents $\mathcal{D}^{s_i}$, a set of queries Q^{s_i} and relevance judgments for each query in Q^{s_i}, $1 \leq i \leq L$. These relevance judgements are assumed to be binary, which is the most common situation. We also consider a target collection $\mathcal{C}^t$, composed of a set of documents $\mathcal{D}^t$ and a set of m queries $Q^t = \{q_1^t, q_2^t, \ldots, q_m^t\}$ without any relevance judgments. Our goal here is to transfer relevance information to $\mathcal{C}^t$ from all the interesting sources in $\{\mathcal{C}^{s_1}, \cdots, \mathcal{C}^{s_L}\}$ and then learn a ranking function on $\mathcal{C}^t$. To do so, we propose (a) to induce relative relevance judgements between documents in $\mathcal{C}^t$ by selecting for each target query the closest source collection (in a sense defined below) and transferring relevance information from it, and (b) to learn a ranking function in $\mathcal{C}^t$ from the relative relevance judgements obtained previously. As in traditional learning to rank approaches for IR, each query-document pair in the target collection $(q^t, d) \in Q^t \times \mathcal{D}^t$ is represented as a K-dimensional feature vector $\mathbf{f}$. The vector attributes, considered in this work are standard features used in document retrieval as well as three state-of-the-art IR scoring functions. These features are presented in more details in section 3. For each query q^t in the target collection, the transfer of relevance information from the source collection to the target one results a set of relative judgement pairs for documents in $\mathcal{D}^t$, of the form $d \succ_{q^t} d'$, where $\succ_{q^t}$ denotes a preference relationship and means *more relevant to query* q^t *than.* From these sets, one can then construct a ranking function $h : \mathbb{R}^K \rightarrow \mathbb{R}$ that assigns a score to documents in $\mathcal{D}^t$ for each query $q^t \in Q^t$. Similarly to previous learning to rank studies, we focus here on linear ranking functions:

$$h_{\mathbf{w}}(\mathbf{f}(q^t, d)) = \langle \mathbf{w}, \mathbf{f}(q^t, d) \rangle$$

where $\langle ., . \rangle$ stands for an inner product and the weight vector $\mathbf{w}$ represents the model parameters. Table 1 gives the notations used in the paper.

Notation	Description
$N(\mathcal{C})$	# of documents in collection $\mathcal{C}$
$N_R^q(\mathcal{C})$	# of relevant documents in $\mathcal{C}$ for query q
$N_w(\mathcal{C})$	# of documents in collection $\mathcal{C}$ containing w
$z_w^{\mathcal{C}}$	Inverse document frequency of w in $\mathcal{C}$
$x_w^{\mathcal{C}}, x_w^d, x_w^q$	# of occurrences of term w in respectively a collection $\mathcal{C}$, a document d and a query q
$l_d, l_q, l_{\mathcal{C}}$	Length of doc., query, coll. in # of terms
$l_{avg}(\mathcal{C})$	Average length of documents in collection $\mathcal{C}$
$\mathcal{R}(q)$	Set of docs relevant to query q

Table 1: Notations

A word w can be characterized by two quantities which constitute the basis of all IR scoring functions: its normalized document frequency, $DF(w) = \frac{N_w(\mathcal{C})}{N(\mathcal{C})}$, and its normalized number of occurrences in any document d of the collection considered, which is set, following [1] and [6], to: $TF(w, d) = x_w^d \log(1 + \frac{l_{avg}(\mathcal{C})}{l_d})$. For any query q and document d in which w occurs, the contribution of w to the relevance of d to q can be estimated through the proportion of relevant documents (to any query q^s) in a source collection $\mathcal{C}^s$ that have the same (DF, TF) values as $DF(w)$ and $TF(w, d)$. However, as typical IR collections only contain few queries, very few words will have exactly the same (DF, TF) values. One way to avoid this problem is to consider regions in the (DF, TF) space into which the different values are considered equivalent. This amounts to discretize the DF and TF values, e.g. by defining intervals on each value range. The probability that d is relevant to q knowing w and a source collection $\mathcal{C}^s$, $\hat{P}(d \in \mathcal{R}(q)|w; \mathcal{C}^s)$, can then be written as:

$$\frac{|\{d' \in \mathcal{D}^s, \exists(q' \in \mathcal{Q}^s, w' \in q'), d' \in \mathcal{R}(q') \wedge \text{eq}_{dis}((w', d'), (w, d))\}|}{|\{d' \in \mathcal{D}^s, \exists(q' \in \mathcal{Q}^s, w' \in q' \cap d'), \text{eq}_{dis}((w', d'), (w, d))\}|}$$

with $\text{eq}_{dis}((w', d'), (w, d))$ meaning $DF_{dis}(w') = DF_{dis}(w)$ and $TF_{dis}(w', d') = TF_{dis}(w, d)$. Where, $DF_{dis}(w)$ and $TF_{dis}(w, d)$ denote the discrete DF and TF values associated to $DF(w)$ and $TF(w, d)$ respectively.

Figure 1 displays the proportion of relevant documents in 88 different regions of the normalized (DF, TF) space for the `TREC-3`, `TREC-7` and `CLEF-3` collections (described in Section 3). Here, the TF dimension has been discretized into 11 values, and the DF one into 8 discrete values (the difference is due to the difference in stretch for the two scores). We call such curves *grids* in the following.

From the above estimate, one can compute a global score, $RSV_{\mathcal{C}^s}(q_t, d)$[1], for a document d in the retrieved set of q^t on the basis of the (log) probability that d is relevant to q^t, that is: $RSV_{\mathcal{C}^s}(q_t, d) = \log P(d \in \mathcal{R}(q^t)|q^t)$. Using Bayes formula, assuming query terms are independent and also supposing that in the absence of any information on the query, all the documents in the collection have the same probability of being relevant, one has:

$$\begin{aligned} P(d \in \mathcal{R}(q^t)|q^t) \quad =_r & \prod_{w \in q^t \cap d} P(d \in \mathcal{R}(q^t)|w)^{x_w^{q^t}} \times \\ & \prod_{w \in q^t \setminus d} P(d \in \mathcal{R}(q^t)|w)^{x_w^{q^t}} \end{aligned}$$

where $=_r$ denotes an equality in rank. The quantity $P(d \in \mathcal{R}(q^t)|w)$, for $w \in q^t \cap d$, can be estimated by $\hat{P}(d \in \mathcal{R}(q^t)|w; \mathcal{C}^s)$. For $w \in q^t \setminus d$, we have: $P(d \in \mathcal{R}(q^t)|w) = P(d \in \mathcal{R}(q^t))$. As relevance judgements are not available in the target collection, one does not have a direct access to this quantity, but it can be estimated through the average of the proportion of relevant documents for queries in the source collection: $\frac{1}{|Q^s|} \sum_{q \in Q^s} \frac{N_R^q(\mathcal{C}^s)}{N(\mathcal{C}^s)}$. We thus finally obtain:

$$\begin{aligned} RSV_{\mathcal{C}^s}(q^t, d) \quad = & \sum_{w \in q^t \cap d} x_w^{q^t} \log(\hat{P}(d \in \mathcal{R}(q^t)|w; \mathcal{C}^s)) \quad (1) \\ & + \sum_{w \in q^t \setminus d} x_w^{q^t} \log\left(\frac{1}{|Q^s|} \sum_{q \in Q^s} \frac{N_R^q(\mathcal{C}^s)}{N(\mathcal{C}^s)}\right) \end{aligned}$$

It is important to note here that, for a target query q^t, $RSV_{\mathcal{C}^s}(q^t, d)$ is based on the information brought by source queries *similar* to q^t. Indeed, the first term in $RSV_{\mathcal{C}^s}(q^t, d)$ makes use of those query-document pairs in $\mathcal{C}^s$ that contain words with similar (DF, TF) values as the ones in (d, q^t).

[1] When the context is clear and in order to simplify the presentation we dropped the subscript q_t of d_{q_t}.

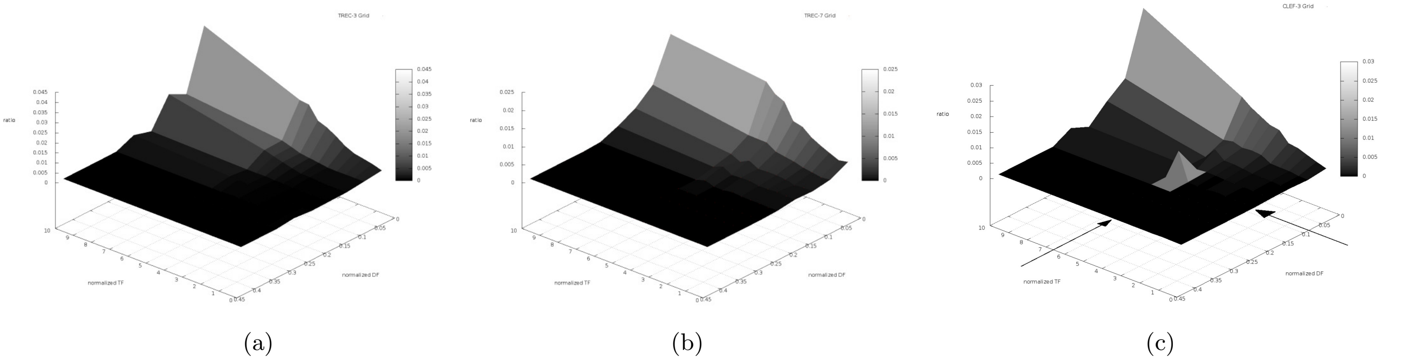

Figure 1: Proportion of relevant documents in each region of the normalized (DF, TF) space, for (a) the `TREC-3` collection, (b) the `TREC-7` collection, and (c) the `CLEF-3` collection. A gray scale is used to differentiate the proportions. The peak in the `CLEF-3` grid is however rendered whiter for readability reasons.

2.2 Selecting the source collection

As one can note from Figure 1, the grids obtained from the three collections differ on several regions, as for example the peak exhibited on the `CLEF-3` grid around the (DF, TF) values of 0.1 and 6.5 (pointed to by arrows in Figure 1). Relevance preferences extracted from the grid from `TREC-3` will miss the behavior of `CLEF-3` query-document pairs in this particular region. This is related to the fact, mentioned above, that different collections tend to use different types of queries, which motivates our will to select, for each target query, a source collection that is appropriate for transfer.

Let q^t be a target query, w a word in q^t and $TF_1, \cdots, TF_n$ the different discrete TF values. As before, $TF_{dis}(w, d)$ denotes the discrete TF value associated to $TF(w, d)$ and $DF_{dis}(w)$ the discrete DF value associated to $DF(w)$. For any collection $\mathcal{C} \in \mathcal{C}^t \cup \{\mathcal{C}^{s_1}, \cdots, \mathcal{C}^{s_L}\}$, let us consider the n-dimensional vector $\mathbf{x}(\mathbf{w}, \mathcal{C})$, the coordinates of which correspond to the normalized number of documents in $\mathcal{C}$ that contain w with a specific term frequency:

$$x(w, \mathcal{C})_i = \frac{|\{d \in \mathcal{C},\ s.t.\ TF_{dis}(w, d) = TF_i\}|}{N_w(\mathcal{C})},\ 1 \leq i \leq n$$

$\mathbf{x}(\mathbf{w}, \mathcal{C})$ thus indicates how w is distributed in the documents of $\mathcal{C}$, an information that can be summarized through the skewness value of $\mathbf{w}(\mathcal{C})$. Indeed, the skewness measures the asymmetry of an empirical distribution and aims at assessing whether the mass of a distribution is concentrated on the right or the left tail; the more similar two distributions are, the closer their skewness values will be. For each word w in q^t, one can thus see which source collection is closest by comparing the skewness values of $\mathbf{w}(\mathcal{C}^{\mathbf{t}})$ and $\mathbf{w}(\mathcal{C}^{\mathbf{s_i}})$ ($1 \leq i \leq L$) (i.e. $sk(\mathbf{x}(\mathbf{w}, \mathcal{C}^{\mathbf{t}}))$ and $sk(\mathbf{x}(\mathbf{w}, \mathcal{C}))$). The score for the query q^t is then simply defined as the average score (here equivalent to the sum) over the query words. The source collection used to build pairwise relevance judgements for q^t is thus defined as:

$$\mathcal{C}^s_{q^t} = \underset{\mathcal{C} \in \{\mathcal{C}^{s_1}, \cdots, \mathcal{C}^{s_L}\}}{\text{argmax}} \sum_{w \in q^t} |sk(\mathbf{x}(\mathbf{w}, \mathcal{C}^{\mathbf{t}})) - sk(\mathbf{x}(\mathbf{w}, \mathcal{C}))| \quad (2)$$

$\mathcal{C}^s_{q^t}$ corresponds to the source collection that displays, in average, the closest (DF, TF) distribution to the words contained in q^t.

2.3 Learning the ranking function

We now introduce the iterative approach we have followed to learn the ranking function h. The relevance information transferred from the selected source collection to the target one takes the form of pairwise relative judgements on which one can learn a pairwise classifier. To do so, for any query q^t in Q^t and any instance pair $(d, d') \in \mathcal{D}^t$, one first constructs a pseudo-label $\tilde{z}(d, d')$ as:

$$\tilde{z}_{d,d'} = \begin{cases} +1 \text{ if } RSV_{\mathcal{C}^s_{q^t}}(q^t, d) \geq RSV_{\mathcal{C}^s_{q^t}}(q^t, d') + \delta \\ -1 \text{ if } RSV_{\mathcal{C}^s_{q^t}}(q^t, d) \leq RSV_{\mathcal{C}^s_{q^t}}(q^t, d') - \delta \end{cases} \quad (3)$$

where $\mathcal{C}^s_q$ is the source collection selected for q^t as described above, and δ represents a margin over the grid values that allows to avoid possible noise in transfer. However, as illustrated in [4], queries with more retrieved documents will be associated with more examples and will impact in an undesired way the function learned.

To avoid that, we randomly sample the set of examples associated to each query so as to have exactly N_p examples for each query. The final training set, $\mathcal{R}$, for learning the pairwise classifier is then collected from all instance pairs in $\mathcal{D}^t$ and their associated pseudo-labels:

$$\mathcal{R} = \{(\mathbf{f}(q^t, d) - \mathbf{f}(q^t, d'), \tilde{z}_{d,d'}); q^t \in Q^t, (d, d') \in \mathcal{D}^t\}$$

The function h is then learned from this set, which contains $N_p \times |Q^t|$ pseudo-labeled vectors, with a standard ranking SVM algorithm [4, 8, 10]. It has to be noted however that, because of the fixed number of examples per query imposed here, the query bias of ranking SVM highlighted in [4] is not present. The procedure we use for transferring relevance information from a source collection $\mathcal{C}^s$ to a target query q^t can then be summarized as follows :

1. Construct the grid for $\mathcal{C}^s$ that provides the proportion of relevant documents in regions (as defined above) of the (DF, TF) space;

2. For all documents in the target collection, compute $RSV_{\mathcal{C}^S}(q^t, d)$ according to Eq. 1;

3. If two documents d and d' in the retrieved set of q^t are such that $RSV_{\mathcal{C}^s}(q^t, d)$ is sufficiently larger than $RSV_{\mathcal{C}^s}(q^t, d')$, then assume that $d \succ_{q^t} d'$.

4. For each query q^t, in the target domain, its most similar source is first selected (Eq. 2), then pseudo-relevance judgments are assigned to N_p random pairs chosen from the retrieved set of q^t (Eq. 3). The pairwise learning/pseudo-labeling steps are then iterated by alternately training a new pairwise classifier on the training set built from all the sets of document pairs and their associated pseudo-labels and assigning pseudo labels to randomly chosen document pairs for queries in the target collection, Q^t. These pseudo-label assignments also follow from equation 3, but using this time the pairwise preferences given by the current ranking model rather than the grid. This algorithm is an instance of the discriminant CEM algorithm [11, p. 39] and it is easy to show that it converges to a discriminant version of the log-classification-likelihood over document pairs. Other variants of the discriminant CEM algorithm have been used in a more traditional semi-supervised learning setting, and applied to various IR problems [2].

3. EXPERIMENTS

We conducted a number of experiments aimed at evaluating to which extend the knowledge transfer presented above can help to learn an efficient ranking function on the target domain. We used nine standard IR collections from `TREC` and `CLEF`[2] evaluation campaigns. Simple statistics of these collections are shown in table 2. Among these data sets, `TREC-6`, `TREC-7` and `TREC-8` use the same document sets (`TREC` disks 4 and 5) but different query sets, whereas rest use unique document sets and unique query sets. We appended `TREC-9` and `TREC-10` Web tracks to experiment with `WT10G`, and `TREC-2004` and `TREC-2005` Terabyte tracks for experimenting with `GOV2`. Our preprocessing steps in creating an index uses Porter stemmer and stop-words removal using the stopword list provided by Terrier[3] [13].

Collection, $\mathcal{C}$	$N(\mathcal{C})$	$l_{avg}(\mathcal{C})$	Index size	$\|Q\|$
`GOV2`	25,177,217	646	19.6 GB	100
`WT10G`	1,692,096	398	1.3 GB	100
`TREC-3`	741,856	261	427.7 MB	50
`TREC-4`	567,529	323	379.0 MB	50
`TREC-5`	524,929	339	378.0 MB	50
`TREC-6`	528,155	296	373.0 MB	50
`TREC-7`				50
`TREC-8`				50
`CLEF-3`	169,477	301	126.2 MB	60

Table 2: Statistics of test collections.

In our *transductive* transfer learning setting, we use all the *unlabeled* set of queries and their associated retrieved document lists in the target collection for training. For evaluation, we use the true relevance judgements provided with the collections. In order to compare the performance of the algorithms we computed the mean average precision (`MAP`) and the average of precision at 10 documents (`P@10`) across queries. Finally, from now on, we designate the transfer from a source collection $\mathcal{C}^s$ to a target collection $\mathcal{C}^t$ using the grid built over the source collection, $\mathcal{G}_{\mathcal{C}^s}$, by: $\mathcal{C}^s \underset{\mathcal{G}_{\mathcal{C}^s}}{\curvearrowright} \mathcal{C}^t$.

To constitute the grid $\mathcal{G}_{\mathcal{C}^s}$, we ranked DF and TF values occurring in a source collection $\mathcal{C}^s$, in an increasing order and considered a step of 0.05 in the DF dimension, and of 0.5 in the TF dimension (the scale difference is due to the fact that DF scores are lower than the TF scores). Furthermore, as in all the test collections we considered, very few terms have DF values above 0.35 and TF values above 5, all the data points above these two values are grouped in the same interval. We finally obtained an 11×8 grid for any source collection, based on 11 discrete values for TF and 8 discrete values for DF. To validate the transfer learning to rank approach with source selection described in the previous section, we considered the following models.

- The proposed transfer learning to rank approach with source selection (denoted as $\texttt{TLR}_{ss}$). For pseudo-label assignments (Eq. 3), we fixed the initial margin $\delta^{(0)}$ as 10% of the diameter of the scoring intervals returned by the aggregation function $RSV_{\mathcal{C}^s_{q^t}}$ (Eq. 1). The increasing step μ is also fixed to 10% of the scoring intervals returned either by the aggregation or the ranking function learned at each step. The precision ϵ is set to 10^{-3}. For learning the ranking function, we employed SVM on the pairwise representation of documents using the *SVMLight* [9] implementation. We fixed the hyperparameter C of the SVM to 10^{-4} and also the number of document pairs for each query to be added in the training set is set to 150.

- The learning to rank approach using training data from the related domain proposed in [7], denoted as $\texttt{LRT}_d$.

- We also considered as baseline models the three standard IR models: language model with Dirichlet smoothing [16] denoted as `LM`, `BM25` [15], and the log-logistic model of the information-based family [6], denoted as `LGD`. Furthermore, because relevance information is not available in the target collection, we fixed the hyperparameters of the these models to their default values provided within the Terrier IR platform, that is: for `BM25`, $b = k_1 = 0.75$ and $k_3 = 7$; for `LM`, the smoothing parameter μ is set to 2500; for `LGD`, the parameter c is fixed to 1.

- A Ranking function learned exclusively on the source collection (denoted as $\texttt{rSVM}_s$) and using the same features than those employed for $\texttt{TLR}_{ss}$. The hyperparameter C is found by cross-validation.

Furthermore, in order to define the attributes, we used the three IR models (`BM25`, `LM`, `LGD`) in the feature vector $\mathbf{f}$ created for each document-query pair. The inherent idea behind this choice is to evaluate in which cases the combination of *standard* scoring functions would be beneficial on a new target collection using the grid. The other vector attributes we considered are standard features used in document retrieval [12]. Features are depicted in table 3.

Finally, experiments are performed on Terrier IR platform v3.5 as all standard modules are integrated. We implemented our models inside this framework and used other necessary standard modules by Terrier, mainly the indexing and the evaluation components.

We start our evaluation by comparing the transfer knowledge based ranking algorithms $\texttt{TLR}_{ss}$ (section 2.3) and $\texttt{LRT}_d$

[2] `http://www.clef-campaign.org`

[3] `http://www.terrier.org`

Features			
1.	$\sum_{w\in q\cap d} \log(1+x_w^d)$	2.	$\sum_{w\in q\cap d} \log(1+\frac{l_{\mathcal{C}}}{x_w^{\mathcal{C}}})$
3.	$\sum_{w\in q\cap d} \log(z_w^{\mathcal{C}})$	4.	$\sum_{w\in q\cap d} \log(1+\frac{x_w^d}{l_d})$
5.	$\sum_{w\in q\cap d} \log(1+\frac{x_w^d}{l_d}.z_w^{\mathcal{C}})$	6.	$\sum_{w\in q\cap d} \log(1+\frac{x_w^d}{l_d}.\frac{l_{\mathcal{C}}}{x_w^{\mathcal{C}}})$
7.	BM25(q,d)	8.	LM(q,d)
9.	LGD(q,d)		

Table 3: Features in the vector representation of (q,d), see table 1 for notations.

[7] using different but fixed source collections to get a first insight into the effectiveness of cross-domain knowledge transfer for learning to rank, and moreover the necessity of source selection. Finally, we analyze the effects of the number of sources on the performance of the learned function.

Table 4, shows MAP results on TREC-3 and TREC-4 taken as target collections using a fixed source collection for knowledge transfer. For LRT$_d$, we took named page finding (NP) adaptation to topic distillation (TD) in the same years (2003 or 2004) and from NP 2003 to TD 2004. For TLR$_{ss}$ we performed knowledge transfer from CLEF-3, TREC-3, TREC-6 and TREC-7 collections to respectively TREC-3 and TREC-4.

		Targets ($\mathcal{C}^t$)	
		TREC-3	TREC-4
LRT$_d$ [7]	$NP \underset{\mathcal{G}_{\mathcal{C}^s}}{\curvearrowright} TD$	0.222	0.179
	$NP03 \underset{\mathcal{G}_{\mathcal{C}^s}}{\curvearrowright} TD04$	-	0.166
TLR$_{ss}$	CLEF-3 $\underset{\mathcal{G}_{\mathcal{C}^s}}{\curvearrowright} \mathcal{C}^t$	**0.257**	0.178
	TREC-3 $\underset{\mathcal{G}_{\mathcal{C}^s}}{\curvearrowright} \mathcal{C}^t$	-	**0.187**
	TREC-6 $\underset{\mathcal{G}_{\mathcal{C}^s}}{\curvearrowright} \mathcal{C}^t$	0.253	0.166
	TREC-7 $\underset{\mathcal{G}_{\mathcal{C}^s}}{\curvearrowright} \mathcal{C}^t$	0.242	0.167

Table 4: MAP measures on TREC-3 and TREC-4 taken as target collections and when only one fixed source is used. Best results are shown in bold.

From these results, it can be seen that the performance of both transfer learning algorithms, on a given target collection, may vary significantly depending on the source collection in use. For example, on the same target TREC-4, the MAP performance of TLR$_{ss}$ varies about 2% depending whether TREC-6 or TREC-3 is used as source collection, while the MAP performance of LRT$_d$ differs about 1% on the same target collection. These results also suggest that the efficiency of transfer learning may highly depend on the adequacy of the source collection with respect to the target one. We now study the behaviour of TLR$_{ss}$ when there are different sources available for knowledge transfer.

Using TREC-3, TREC-4, TREC-5, TREC-6 and CLEF-3 as source collections, we measured the MAP and P@10 of all the models (except LRT$_d$ for which the source selection step is not trivial to carry out) on the remaining data sets. Table 5 summarizes these results. From these results it becomes clear that the transfer ranking algorithm TLR$_{ss}$ consistently and significantly improves over other IR models on MAP and P@10 in most cases. Further, on GOV2, the improvements on the MAP and P@10 are significantly better at a p-value threshold of 0.01 with respect to all non-transfer based IR models. This tendency seems to be less true on smaller target collections like TREC-7 and TREC-8. And finally, the pairwise SVM ranking function (rSVM$_s$) gives also the lowest results, suggesting that although learning to rank functions may be effective on the test collections from the same domains than training sets they are learned, however, they cannot be efficaciously applied to datasets from other domains.

	Targets			
Sources	TREC-7	TREC-8	WT10G	GOV2
TREC-3	24%	22%	14%	16%
TREC-4	18%	20%	11%	9%
TREC-5	14%	14%	14%	16%
TREC-6	24%	24%	43%	26%
CLEF-3	20%	20%	18%	33%

Table 6: Percentage of sources selected for queries of each target collection using the proposed grid-based source selection strategy (Eq. 2).

Another interesting observation is that the information provided by sources becomes less effective as a set of sources which are the most similar to queries in the target collection is identified. This observation is confirmed with results on figure 2 which show that for increasing number of sources added sequentially with respect to their sizes (see table 2), the MAP measures on TREC-7 and WT10G target collections improve slowly after the second smallest source collection (TREC-6) is taken in the set of sources used in our experiments. Table 6 provides the percentage of sources selected for queries of each target collection using the strategy presented in Section 2.2. As indicated in table 6, these two smallest source collections are the most similar to merely half of the queries in TREC-7 and WT10G.

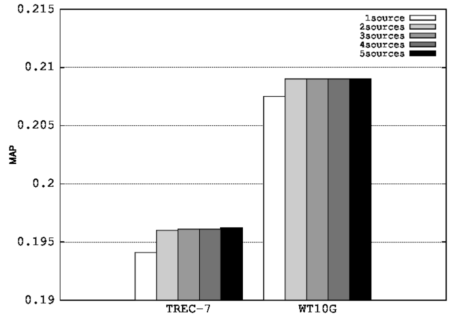

Figure 2: The evolution of MAP on TREC-7 and WT10G target collections with respect to the increasing number of sources from 1 to 5 added in increasing order of their index size (see table 2).

4. CONCLUSION

We have addressed in this study the problem of learning a ranking function from collections without relevance information. To do so, we have used a transfer learning approach that consists in selecting the most similar source collection to each query in the target dataset and to derive, from absolute relevance judgements available in the selected source

	$\mathcal{C}^t \equiv$ TREC-7		$\mathcal{C}^t \equiv$ TREC-8		$\mathcal{C}^t \equiv$ WT10G		$\mathcal{C}^t \equiv$ GOV2	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
rSVM$_s$	$0.181^{\downarrow}$	$0.408^{\downarrow\downarrow}$	$0.241^{\downarrow\downarrow}$	$0.460^{\downarrow}$	$0.182^{\downarrow\downarrow}$	$0.257^{\downarrow\downarrow}$	$0.244^{\downarrow\downarrow}$	$0.477^{\downarrow\downarrow}$
BM25	$0.182^{\downarrow}$	$0.418^{\downarrow\downarrow}$	$0.241^{\downarrow\downarrow}$	0.472	$0.184^{\downarrow\downarrow}$	$0.291^{\downarrow}$	$0.271^{\downarrow\downarrow}$	$0.533^{\downarrow\downarrow}$
LM	0.186	$0.392^{\downarrow\downarrow}$	$0.240^{\downarrow}$	$0.432^{\downarrow\downarrow}$	0.204	$0.293^{\downarrow}$	$0.277^{\downarrow\downarrow}$	$0.549^{\downarrow\downarrow}$
LGD	$0.188^{\downarrow}$	0.428	$0.254^{\downarrow\downarrow}$	**47.40**	$0.194^{\downarrow}$	$0.287^{\downarrow}$	$0.284^{\downarrow\downarrow}$	$0.536^{\downarrow\downarrow}$
TLR$_{ss}$	**0.196**	**0.446**	**0.262**	0.468	**0.209**	**0.310**	**0.309**	**0.582**

Table 5: MAP and P@10 measures on different target collections where query wise source data sets are selected from CLEF-3, TREC-3,4,5,6. The best results are in bold, and a $^{\downarrow}$ (respectively $^{\downarrow\downarrow}$) indicates a result that is statistically significantly worse than TLR$_{ss}$, according to a Wilcoxon rank sum test [14] used at a p-value threshold of 0.05 (respectively 0.01).

collection, relative relevance judgements in a target collection. This derivation relies on a grid that associates to each (DF, TF) value of a term in a query-document pair a relevance score, which is then combined over all query terms. A ranking SVM system is then deployed on the obtained relative relevance judgements, and further improved through a self-learning mechanism. The experiments we have conducted show that the ranking function obtained in this way consistently and significantly outperfoms state-of-the-art IR ranking functions in the majority of cases on 4 large collections from TREC with respect to the MAP the P@10 measures.

Our approach directly learns a ranking function on the target collection (as opposed to previous approaches developed in the same setting and which learned the ranking function on re-weighted version of the source collection), which allows one to develop a simple source selection procedure. This is, to our knowledge, the first time that a source selection procedure for transfer learning for IR is proposed and validated on a whole range of IR collections, including large scale collections as WT10G and GOV2.

The consistent improvements obtained with TLR$_{ss}$ show that it is possible to learn an efficient combination of state-of-the-art IR scoring functions from the relevance judgements provided by a *selected* source collection. These findings go one step further than results presented table 4, indicating that source selection using the grid information (section 2.2) may be effective for learning a domain adaptive ranking function.As depicted in table 6, it can be seen that sources are not uniformly selected for each target collection suggesting which of these source datasets are more related to each of the latter according to our selecting scheme. Finally, the information provided by the grid and the state-of-the-art IR scoring functions allowing to learn this function (Eq. 3) is also complimentary.

5. REFERENCES

[1] G. Amati and C. J. V. Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

[2] M. R. Amini and P. Gallinari. The use of unlabeled data to improve supervised learning for text summarization. In *Proceedings of ACM SIGIR*, pages 105–112, 2002.

[3] P. Cai, W. Gao, A. Zhou, and K.-F. Wong. Query weighting for ranking model adaptation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, ACL'11, pages 112–122, 2011.

[4] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking SVM to document retrieval. In *Proceedings of ACM SIGIR*, pages 186–193, 2006.

[5] D. Chen, Y. Xiong, J. Yan, G.-R. Xue, G. Wang, and Z. Chen. Knowledge transfer for cross domain learning to rank. *Information Retrieval*, 13(3):236–253, June 2010.

[6] S. Clinchant and E. Gaussier. Information-based models for ad hoc IR. In *Proceedings of ACM SIGIR*, pages 234–241, 2010.

[7] W. Gao, P. Cai, K.-F. Wong, and A. Zhou. Learning to rank only using training data from related domain. In *Proceedings of ACM SIGIR*, SIGIR '10, pages 162–169, New York, NY, USA, 2010. ACM.

[8] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *In International Conference on Artificial Neural Networks*, pages 97–102, 1999.

[9] T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods*. MIT Press, 1999.

[10] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of ACM SIGKDD*, pages 133–142, 2002.

[11] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, Inc. New York, 1992.

[12] R. Nallapati. Discriminative models for information retrieval. In *Proceedings of ACM SIGIR*, pages 64–71, 2004.

[13] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

[14] J. A. Rice. *Mathematical Statistics and Data Analysis.* Duxbury Advanced, 2006.

[15] S. E. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 2009.

[16] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of ACM SIGIR*, 2001.