Exploiting Hyperlinks to Learn a Retrieval Model

David Grangier **Samy Bengio**
IDIAP Research Institute, Martigny, Switzerland,
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{grangier,bengio}@idiap.ch

Abstract

Information Retrieval (IR) aims at solving a ranking problem: given a query q and a corpus C, the documents of C should be ranked such that the documents relevant to q appear above the others. This task is generally performed by ranking the documents $d \in C$ according to their similarity with respect to q, $sim(q, d)$. The identification of an effective function $a, b \to sim(a, b)$ could be performed using a large set of queries with their corresponding relevance assessments. However, such data are especially expensive to label, thus, as an alternative, we propose to rely on hyperlink data which convey analogous semantic relationships. We then empirically show that a measure sim inferred from hyperlinked documents can actually outperform the state-of-the-art *Okapi* approach, when applied over a non-hyperlinked retrieval corpus.

1 Introduction

Information Retrieval (IR) consists in finding documents that are relevant to a given query in a large corpus (e.g. more than $100,000$ documents). This task is generally formulated as a ranking problem: given a query q and a set of documents D, an IR system should output a document ranking in which the relevant documents appear above non-relevant ones. In order to achieve such a goal, a common approach consists in ranking the documents $d \in D$ according to their similarity $sim(q, d)$ with respect to the query q [1]. Hence, the identification of a reliable measure of the semantic similarity between text items is of crucial importance in IR. In fact, such a measure sim should ideally compare sequences of terms, referred to as documents and queries in this case, such that

$$\forall q, \forall d^+ \in R(q), \forall d^- \notin R(q), sim(q, d^+) - sim(q, d^-) > 0, \tag{1}$$

$R(q)$ being the set of documents which are relevant to q. This property actually ensures that relevant documents are ranked above non-relevant ones for any query.

The selection of an appropriate measure of similarity could hence be performed through the optimization of a criterion related to (1) over some training data [2, 5]. However, such a process would require a large set of labeled queries for training (i.e. queries with the corresponding relevance set) which are expensive to obtain [1]. As an alternative, we propose to identify an effective measure from already available hyperlinked data D_{train} that can then be applied over any IR corpus D_{test}, with or without hyperlinks.

This approach relies on hyperlinks for training, since such data contain information about the semantic proximity of documents which are close to the document/query relationships provided by relevance assessments. In fact, it has been observed [4] that, in most cases, a document d is semantically closer to a document l^+, hyperlinked with d, than to a document l^-, not hyperlinked with d:

$$\forall d \in D_{train}, \forall l^+ \in L(d), \forall l^- \notin L(d), sim(d, l^+) - sim(d, l^-) > 0, \qquad (2)$$

where $L(d)$ refers to the set of documents linked with d (i.e. the documents referring to d and the documents referred to by d). This kind of relationship is hence analogous to relevance assessments which state that a query q is semantically closer to a document d^+, relevant to q, than to a document d^-, not relevant to q (1).

Our task is hence to identify a measure sim that would ideally verify (2). For that purpose, we introduce a parameterized similarity measure sim_θ and a cost C which penalizes the parameters θ for which a large number of constraints (2) are not verified. The parameter θ^* that minimizes C is then selected through stochastic gradient descent (see Section 2). The function sim_{θ^*} inferred with this approach has then been compared with the state-of-the-art *Okapi* matching [6] over a benchmark IR corpus (TREC-9 queries over the TDT-2 documents). The performance of our approach is shown to outperform *Okapi* with respect to various IR measures (Precision at top 10, P10, Average Precision, AvgP, and Break-Even Point, BEP), the relative improvement being greater than 10% for all measures (see Section 4).

The remainder of this paper is organized as follows, Section 2 describes the proposed model, *LinkLearn*, Section 3 compares this model with alternative approaches, Section 4 presents IR experiments to assess the effectiveness of our approach, finally, Section 5 draws some conclusions.

2 The LinkLearn Model

In this Section, we describe the *LinkLearn* model: first, the parameterization is introduced and then, the training procedure is described.

Model Parameterization
LinkLearn relies on a parametric function sim_θ to compute the similarity between text items. To introduce such a function, we first present how query/document similarity is computed in ad-hoc retrieval systems and we then define a parameterized measure inspired from these approaches.

The Vector Space Model (VSM) is the most common framework to compare text items in IR systems. In this context, each document d is first *indexed* with a vocabulary-sized vector,

$$d = (d_1, \ldots, d_V),$$

where d_i is the weight of term i in document d and V is the vocabulary size. Then, the dot product between such vectors is then used to assess the document similarity. This VSM approach is also often referred to as the *bag-of-words* model, as term ordering is not taken into account. The weights d_i are generally computed as an a-priori defined function of some features of i and d, such as $tf_{i,d}$ the number of occurrences of i in d, df_i the number of documents of D_{train} containing term i, l_d the length of document d (i.e. the total number of term occurrences in d). For instance, the most common weighting function, *Okapi BM25* [6], computes such a weight as

$$d_i = \frac{(K+1) \cdot tf_{i,d} \cdot idf_i}{K \cdot ((1-B) + B \cdot (l_d/L)) + tf_{i,d}},$$

where idf_i is defined as $\log(N/df_i)$, N is the total number of documents in D_{train}, L is the mean of l_d over D_{train}, and K, B are hyperparameters to select.

We hence adopt a similar approach to parameterize our model. In our case, the weight of a term in a document is computed as,

$$d_i^\theta = f_\theta(tf_{i,b}, idf_i, l_b),$$

where f_θ is the product of the outputs of three single-output Multi-Layer Perceptrons (MLP),

$$f_\theta : x, y, z \rightarrow MLP_{\theta_1}(x) \cdot MLP_{\theta_2}(y) \cdot MLP_{\theta_3}(z), \tag{3}$$

and $\theta = [\theta_1, \theta_2, \theta_3]$ corresponds to the MLP parameters. This hence leads to the following parameterized measure of similarity,

$$sim_\theta : a, b \rightarrow \sum_{i=1}^{V} f_\theta(tf_{i,a}, idf_i, l_a) \cdot f_\theta(tf_{i,b}, idf_i, l_b).$$

This measure therefore only relies on simple features of term occurences which makes it *vocabulary-independent*, i.e. the learned parameters are not linked to a specific term set and the function sim inferred from one corpus can therefore be applied to another corpus, possibly indexed with a different vocabulary (e.g. in Section 4, for TREC experiments, training and testing are performed using vocabulary extracted from different corpora).

The proposed parameterization (3) involves 3 different MLPs, each one having a real valued input, which is a limitation with respect to a model where function f would be a unique MLP with a 3-dimensional input. Such a simplification is however necessary in order to apply the model over large corpora since it significantly reduces the required computational cost for both training and testing: instead of evaluating an MLP function for all triplets $\forall d, i, (tf_{d,i}, idf_i, l_d)$, it should only be evaluated for each possible value of $tf_{d,i}$, idf_i and l_d. In Section 4, the number of MLP evaluations would for instance have been $\sim 1,000$ times greater with a single MLP. Moreover, the experimental results show that this simplified parameterization does not prevent our model from reaching good performance.

Model Criterion and Training

This Section describes how the parameter vector θ of the function sim_θ is selected such that most constraints of (2) are respected. For that purpose, we introduce a cost C related to (2) that can be minimized through stochastic gradient descent.

A simple cost to minimize in this context could be the number of constraints which are not verified,

$$C^{0/1} = \sum_{d \in D_{train}} C_d^{0/1}, \tag{4}$$

$$\text{where} \qquad C_d^{0/1} = \sum_{l^+, l^- \in L(d) \times \overline{L(d)}} I\{sim_\theta(d, l^+) - sim_\theta(d, l^-) < 0\} \tag{5}$$

and $I\{\cdot\}$ is the indicator function ($I\{c\} = 1$ if c is true and 0 otherwise).

However, similarly to the $0/1$ loss in the case of classification problems, this cost is obviously not suitable for gradient descent (i.e. its gradient is null everywhere). We hence propose to minimize an upper bound of this quantity:

$$C = \sum_{d \in D_{train}} C_d, \tag{6}$$

$$\text{where} \qquad C_d = \sum_{l^+, l^- \in L(d) \times \overline{L(d)}} |1 - sim_\theta(d, l^+) + sim_\theta(d, l^-)|_+ \tag{7}$$

and $x \rightarrow |x|_+$ is x if $x > 0$, 0 otherwise. This cost is actually an upper bound of $C^{0/1}$ since $\forall x, |1 - x|_+ \geq I\{x < 0\}$. C is then minimized through stochastic gradient descent, i.e. we iteratively pick documents in D_{train} and update θ according to $\partial C_d / \partial \theta$. The hyperparameters of the model (i.e. the number of hidden units in the MLPs, the number of training iterations and the learning rate) are selected through cross-validation (see Section 4).

The use of C has two main advantages: from a theoretical perspective, the minimization of C can be interpreted as margin maximization [3]. Moreover, from a practical point of view, the gradient $\partial C_d/\partial\theta$ is especially inexpensive to compute since

$$1 - sim_\theta(d, l^+) + sim_\theta(d, l^-) < 0 \Rightarrow \frac{\partial}{\partial\theta}|1 - sim_\theta(d, l^+) + sim_\theta(d, l^-)|_+ = 0.$$

This effectiveness aspect is crucial for training over large datasets, giving to *LinkLearn* a scalability advantage over alternative approaches, as explained in the following.

3 Related Works

The inference of document similarity measures (or equivalently document distance metrics) from a set of proximity constraints P_{train} of type

"document a is closer to document b than it is to document c,"

is a recent research topic in Machine Learning. In the following, two alternative models are described: *Ranking SVM*, a Support Vector Machine approach, and *RankNet*, a model based on MLP and gradient descent optimization.

Ranking SVM [7] is a distance learning model: it aims at identifying d_w,

$$d_w : x, y \rightarrow \sqrt{\sum_{i=1}^{V} w_i (x_i - y_i)^2},$$

where $\forall i, w_i > 0$, from the constraint set P_{train}:

$$\forall (a, b, c) \in P_{train}, d_w(a, b) < d_w(a, c).$$

As a distance is always positive, the constraints can be reformulated as,

$$\forall (a, b, c) \in P_{train}, d_w(a, c)^2 - d_w(a, b)^2 > 0.$$

To ensure good generalization performance, a margin maximization approach is then adopted, leading to the following problem,

$$\begin{array}{ll} \min\limits_{w,\xi} & \|w\|_2 \quad + \quad C \sum\limits_{(a,b,c)\in P_{train}} \xi_{a,b,c} \\ s.t. & \left\{ \begin{array}{l} \forall (a, b, c) \in P_{train}, d_w(a, c)^2 - d_w(a, b)^2 \geq 1 - \xi_{a,b,c} \\ \forall (a, b, c) \in P_{train}, \xi_{a,b,c} \geq 0 \\ \forall i = 1 \ldots V, w_i \geq 0. \end{array} \right. \end{array} \tag{8}$$

where C is an hyperparameter that control the trade-off between the margin size and the number of non-verified constraints. Such a model has shown to be effective empirically: e.g. it has notably been used to combine different search engine outputs [5]. However, the resolution of (8) through quadratic optimization becomes computationally costly as the training set size $|P_{train}|$ grows, i.e. $\sim O(|P_{train}|^p), 2 < p \leq 3$, making gradient descent approaches like *LinkLearn* or *RankNet* a suitable alternative for large constraint sets.

RankNet [2] is a gradient based approach to similarity measure learning. Like *ranking SVM* and *LinkLearn*, this model is also trained from a set proximity constraints P_{train},

$$\forall (a, b, c) \in P_{train}, sim(a, b) > sim(a, c).$$

In this case, each $(a, b, c) \in P_{train}$ is additionally labeled with $p_{a,b,c}$, the probability that constraint (a, b, c) is actually true. This allows for including some confidence information about the training constraints while not preventing to use a set P_{train} without probability (i.e. in this case, it can be assumed that $\forall (a, b, c) \in P_{train}, p_{a,b,c} = 1$).

RankNet relies on some feature vector[1] $\phi(a, b)$ to compute the similarity between text items a and b,

$$sim_\theta(a, b) = MLP_\theta(\phi(a, b))$$

The parameter vector θ is then identified from P_{train} through the minimization of the cross-entropy (CE) criterion:

$$C^{(CE)} = \sum_{(a,b,c)\in P_{train}} C^{(CE)}_{a,b,c}, \tag{9}$$

$$\text{where} \quad C^{(CE)}_{a,b,c} = -p_{a,b,c} \log o_{a,b,c} - (1 - p_{a,b,c}) \log(1 - o_{a,b,c}) \tag{10}$$

$$\text{and} \quad o_{a,b,c} = \frac{\exp(sim_\theta(a, b) - sim_\theta(a, c))}{1 + \exp(sim_\theta(a, b) - sim_\theta(a, c))}. \tag{11}$$

Like for *LinkLearn*, this cost can then be minimized through gradient descent optimization. *RankNet* and *LinkLearn* approaches are hence close: the use of gradient descent allows for their application over large training sets. Moreover, they could be applied with any differentiable function sim_θ which enables to easily include some a-priori knowledge about document similarity measures.

These two models are however not identical. On one hand, *RankNet* allows for the assignment of different confidence levels for the proximity constraints (through $p_{a,b,c}$), which can be advantageous in the case where the constraints come from several annotators that may disagree. On the other hand, *LinkLearn* cost allows for efficient gradient computation (see Section 2), which makes it suitable for large training set (e.g. in next Section, *LinkLearn* has been trained over $\sim 10^{11}$ constraints).

4 Experiments and Results

In this Section, we assess the performance of *LinkLearn* according to the following experimental setup: the model is first trained over the *Wikipedia* hyperlinked corpus and the inferred measure sim_{θ^*} is then used to rank the documents of *TDT-2* corpus with respect to *TREC-9* ad-hoc queries. The IR performance over this test set is then compared with respect to the state-of-the-art *Okapi* approach.

Training over Wikipedia Corpus
The Wikipedia corpus[2] consists of encyclopedia articles, each article referring to other related articles using hyperlinks. To train *LinkLearn*, two subsets D_{train} and D_{valid} of $\sim 150,000$ documents have been randomly extracted from the whole dataset ($\sim 450,000$ documents) such that no document belongs to both sets. The hyperlinks which does not start and end in the same subset have been removed, resulting in an average of 13.4 and 12.5 links per documents for D_{train} and D_{valid}. The D_{train} set is used for gradient descent (i.e. C is minimized over this set) and D_{valid} is used to select the model hyperparameters. In order to have an estimate of the IR performance on D_{valid}, the following artificial retrieval task is introduced: each document $d \in D_{valid}$ is considered to be a query whose relevant documents are the documents linked with d and average precision is measured for this task (Figure 1 reports this measurement during training).

Evaluation with TREC-9 queries
In this Section, *LinkLearn* and *Okapi* are compared on TREC-9 queries for the TDT-2 corpus[3]. The TDT-2 corpus contains $24,823$ documents and there are 50 TREC-9 queries, each query having, on average, 13.2 relevant documents. For *LinkLearn*, no re-training

[1] We do not describe ϕ since it has only been briefly presented in the original description of *RankNet* [2].
[2] Wikipedia corpus and documentation are available at download.wikimedia.org.
[3] TREC data and documentation are available at trec.nist.gov.

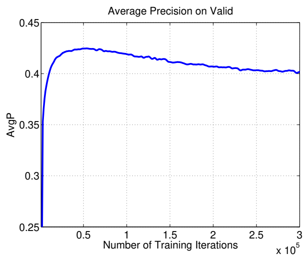

Figure 1: Validation Performance during Training
This plot depicts validation performance up to $300,000$ iterations but early stopping criterion has actually stopped training before over-fitting on the $AvgP$ curve (i.e. after $54,000$ iterations).

Table 1: Retrieval Results over TDT-2/TREC-9 data

	Okapi	*LinkLearn*
P10	38.8%	43.2% (+11%)
BEP	30.3%	35.2% (+16%)
AvgP	29.3%	34.5% (+18%)

or adaptation have been performed. The *LinkLearn* measure inferred from Wikipedia has directly been applied as a query/document matching measure to TDT-2/TREC-9. For *Okapi*, the hyperparameters K, B have been selected through cross-validation over TREC-8 queries. To assess the IR performances of both methods, Precision at top 10, $P10$, Average Precision, $AvgP$, and Break-Even Point, BEP results are reported in Table 1. According to all measures, *LinkLearn* performs better than *Okapi* and the relative improvement is more than 10% in all cases.

5 Conclusions

In this paper, we introduced *LinkLearn*, a gradient descent approach to derive a document similarity measure from a hyperlinked training corpus: the measure is selected such that, in most cases, a document is considered more similar to the documents with which it is linked than to the other documents. The inferred measure can then be applied to compare any text items with or without hyperlinks. In particular, a measure learned with *LinkLearn* over an encyclopedia corpus (Wikipedia) has shown to outperform state-of-the-art *Okapi* matching measure when used to compare documents and queries in an IR ranking problem (TDT-2/TREC-9).

Acknowledgments:
This work has been performed with the support of the Swiss NSF through the NCCR–IM2 project. It was also supported by the PASCAL European Network of Excellence, funded by the Swiss OFES.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[2] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.

[3] R. Collobert and S. Bengio. Links between perceptrons, MLPs and SVMs. In *ICML*, 2004.

[4] B. D. Davison. Topical locality in the web. In *SIGIR*, 2000.

[5] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[6] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *TREC*, 1994.

[7] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2003.