# Supervised and Semi-supervised Machine Learning Ranking

Jean-Noël Vittaut and Patrick Gallinari

Laboratoire d'Informatique de Paris 6
104, avenue du Président-Kennedy, F-75016 Paris, France
{vittaut, gallinari}@poleia.lip6.fr

**Abstract.** We present a Semi-supervised Machine Learning based ranking model which can automatically learn its parameters using a training set of a few labeled and unlabeled examples composed of queries and relevance judgments on a subset of the document elements. Our model improves the performance of a baseline Information Retrieval system by optimizing a ranking loss criterion and combining scores computed from doxels and from their local structural context. We analyze the performance of our supervised and semi-supervised algorithms on CO-Focussed and CO-Thourough tasks using a baseline model which is an adaptation of Okapi to Structured Information Retrieval.

## 1 Introduction

Different studies and developments have been recently carried out on ranking algorithms in the machine learning community. In the field of textual documents, they have been successfully used to combine features or preferences relations in tasks such as meta search [2] [3] [8], passage classification, automatic summarization [1] and recently for the combination of different sources of evidence in Information Retrieval (IR) [5]. One of the challenges of this paradigm is to reduce the complexity of the algorithms which is in the general case quadratic in the number of samples. This is why most real data applications of ranking are based on two-classes problems. Nevertheless, some linear methods has been proposed [8] [1] and under some conditions, fast rates of convergence are achieved with this class of methods [4].

Ranking algorithms work by combining features which characterize the data elements to be ranked. In our case, these features will depend on the doxel itself and on its structural context. Ranking algorithms will learn to combine these different features in an optimal way according to a specific loss function using a set of examples. It is hoped that ranking algorithms may help to improve the performance of existing techniques.

The first ideas to combine labeled and unlabeled data come from the statistician community at the end of the 70's . Most of these methods use a mixture of gaussian model and try to estimate its parameters by maximizing the joint likelihood of labeled and unlabeled data [10]. There is in general a one to one relation between classes and mixture components [9]. As it is usually impossible to perform this estimation directly, so they use a class of iterative algorithms called EM for Expectation Maximization [6]. The semi-supervised paradigm has also been used in IR [13]. It has showed its efficiency on a class of tasks where there are only a few labeled samples available. This is the case in SIR and more generally in IR: there are very few labeled databases, and the labeling of relevant units for particular queries is time consuming. There is also tasks where only a few samples are labeled like in relevance-feedback.

The paper is organized as follows, in section 2 we present the ranking model, in section 3 we show how we adapted it to CO-Focussed and CO-Thorough tasks. We also show how we learn using both labeled and unlabeled examples. In section 4 we comment the results reached by our model and compare it to a baseline Okapi method adapted for Structured Information Retrieval (SIR).

## 2 Ranking model

We present in this section a general model of ranking which can be adapted to IR or SIR. The idea of the ranking algorithms proposed in the machine learning community is to learn a total order on a set $\mathcal{X}$, which allows to compare any element pair in this set. Given this total order, we are able to order any subset of $\mathcal{X}$ in a ranking list. For instance in IR, $\mathcal{X}$ can be the set of couples (document, query), and the total order is the natural order on the document scores.

As for any machine learning technique, one needs a training set of labeled examples in order to learn how to rank. This training set will consist in ordered pairs of examples. This will provide a partial order on the elements of $\mathcal{X}$. The ranking algorithm will use this information to learn a total order on the elements of $\mathcal{X}$ and after that will allow to rank new elements. For plain IR, the partial ordering may be provided by human assessments on different documents for a given query.

### 2.1 Notations

Let $\mathcal{X}$ be a set of elements with a partial order $\prec$ defined on it. This means that some of the element pairs in $\mathcal{X}$ may be compared according to the $\prec$ relation. For Structured Information retrieval $\mathcal{X}$ will be the set of couples (doxel, query) for all doxels and queries in the document collection. This set is partially ordered according to the existing relevance judgments for each query.

### 2.2 Ranking

Let $f$ be a function from $\mathcal{X}$ to the set of real numbers. We can associate a total order $\prec_T$ to $f$ such that:

$$x \prec_T x' \Leftrightarrow f(x) < f(x') \ . \qquad (1)$$

Clearly, learning the $f$ function is the same as learning the total order. In the following, we will extend the partial order $\prec$ to a total order $\prec_T$, so we will use the same notation for both relations.

An element of $\mathcal{X}$ will be represented by a real vector of features:

$$x = (x_1, x_2, ..., x_d).$$

In our case, the features will be local scores computed on different contextual elements of a doxel. In the following, $f$ will be a linear combination of $x$'s features:

$$f_\omega(x) = \sum_{j=1}^{d} \omega_j x_j \tag{2}$$

where $\omega = (\omega_1, \omega_2, ..., \omega_d)$ are the parameters of the combination to be learned.

**Ranking loss.** $f_\omega$ is said to respect $x \prec x'$ if $f_\omega(x) < f_\omega(x')$. In this case, couple $(x, x')$ is said to be well ordered by $f_\omega$. The ranking loss [8] measures how much $f_\omega$ respects $\prec$.

By definition, the ranking loss measures the number of mis-ordered couples in $\mathcal{X}^2$:

$$R(\mathcal{X}, \omega) = \sum_{\substack{(x,x') \in \mathcal{X}^2 \\ x \prec x'}} \chi(x, x') \tag{3}$$

where $\chi(x, x') = 1$ if $f_\omega(x) > f_\omega(x')$ and 0 otherwise.

Ranking aims at learning $\omega$ for minimizing (3).

**Exponential loss.** In practice, this expression is not very useful since $\chi$ is not differentiable, ranking algorithms use to optimize another loss criterion called the exponential loss:

$$R_e(\mathcal{X}, \omega) = \sum_{\substack{(x,x') \in \mathcal{X}^2 \\ x \prec x'}} e^{f_\omega(x) - f_\omega(x')}. \tag{4}$$

If is straightforward that $R(\mathcal{X}, \omega) \leq R_e(\mathcal{X}, \omega)$. (4) is differentiable and convex, and then can be minimized using standard optimization techniques. Minimizing (4) will allow to minimize $R(\mathcal{X}, \omega)$.

We can compute a gradient descent. The components of the gradient of $R_e$ are:

$$\frac{\partial R_e}{\partial \omega_k}(\mathcal{X}, \omega) = \sum_{\substack{(x,x') \in \mathcal{X}^2 \\ x \prec x'}} (x_k - x'_k) e^{f_\omega(x) - f_\omega(x')}. \tag{5}$$

With no more hypothesis, the computation of (5) is in $O(|\mathcal{X}|^2)$.

## 3 Application to CO tasks

### 3.1 Definitions

Let denote $\mathcal{D}$ is the set of doxels for all the documents in the collection and $\mathcal{Q}$ the set of CO queries. $\mathcal{X} = \mathcal{Q} \times \mathcal{D}$ is the set of elements we want to order.

We suppose that there exists a partial order $\prec$ on $\mathcal{X} = \mathcal{Q} \times \mathcal{D}$, this partial order will reflect for some queries, the evidence we have about preferences between doxels provided via manual assessments. Note that these relevance assessments are only needed for a few queries and doxels in the collection. We consider here the task which consists in producing a ranked list of doxels which answer the query $q \in \mathcal{Q}$. For that, we will train the ranking model to learn a total strict order on $\mathcal{X}$.

### 3.2 Combination of preference relations

We suppose each element $x \in \mathcal{X}$ can be ordered according to several preference relations $pref_i$. Let denote $\mathcal{T}$ the set of doxel types, which are defined according to the DTD of the document collection: article, abstract, sections, paragraphs, lists...

We used the following combination:

$$f_\omega(x) = \sum_{t \in \mathcal{T}} 1_{[node\_type(x)=t]} \cdot \omega_t^{\emptyset} \cdot \left(1 + \sum_i \omega_t^i \cdot pref_i(x)\right)$$

where $t$ is the node type of $x$ and $pref_i$ is a preference relation among doxels based on textual content. In our experiments, we used a SIR adapted Okapi model [11] described in [12]. This adaptation consists in using doxels rather than documents for computing the term frequencies, and using as normalization factor for each doxel, the mean size of the doxels with the same node type. For each doxel, this Okapi model was computed on its textual content, on its parent textual content and also on the whole document containing it.

This combination take into account the information provided by the context of the doxel and the structural information given by the node type of the doxel.

### 3.3 Reduction of complexity

In this section, we use some properties of SIR in order to decrease the complexity of the computation of (4) and (5).

**Queries.** Comparing elements from different queries has no sense. We can define a partition $\mathcal{X} = \bigcup_{q \in \mathcal{Q}} \mathcal{X}_q$, where

$$\mathcal{X}_q = \{x = (d, q') \in \mathcal{X} / q' = q\}$$

and we can rewrite (4):

$$R_e(\mathcal{X},\omega) = \sum_{q\in\mathcal{Q}} \left\{ \sum_{\substack{(x,x')\in\mathcal{X}_q\times\mathcal{X}_q \\ x\prec x'}} e^{f_\omega(x)} e^{-f_\omega(x')} \right\}. \tag{6}$$

**Assessments.** For each subset $\mathcal{X}_q$, we suppose we have information indicating preferences among doxels:

- an information of exhaustivity, which measures how much a doxel answers the totality of an information need
- an information of specificity, which measures how much a doxel answers only the information need

For doxels which are equally exhaustive or specific, there is no preference.

An assessment is a couple (exhaustivity, specificity). Let denote $\mathcal{A}$ the set of assessments and $A(x)$ the assessment of element $x$. We can define a partition $\mathcal{X}_q = \bigcup_{a\in\mathcal{A}} \mathcal{X}_q^a$, where

$$\mathcal{X}_q^a = \{x \in \mathcal{X}_q / A(x) = a\}.$$

We can rewrite (6):

$$R_e(\mathcal{X},\omega) = \sum_{q\in\mathcal{Q}} \sum_{a\in\mathcal{A}} \left\{ \left( \sum_{x\in\mathcal{X}_q^a} e^{f_\omega(x)} \right) \left( \sum_{\substack{b\in\mathcal{A} \\ \mathcal{X}_q^b \prec \mathcal{X}_q^a}} \sum_{x\in\mathcal{X}_q^b} e^{-f_\omega(x)} \right) \right\}. \tag{7}$$

where $\mathcal{X}_q^b \prec \mathcal{X}_q^a$ means that the assessments of the elements of $\mathcal{X}_q^a$ are better than those of $\mathcal{X}_q^b$.

The complexity for computing this expression is $O(K \cdot |\mathcal{Q}| \cdot |\mathcal{X}|)$ whereas it is $O(|\mathcal{X}|^2)$ for (4) where $K$ is the number of sets in the partition of $\mathcal{X}$. The worst case occurs when $K = |\mathcal{X}|$.

### 3.4 Gradient descent

Since (7) is convex, we can use a gradient descent technique to minimize it. The components of the gradient has the following form:

$$\begin{aligned} \frac{\partial R_e}{\partial \omega_k}(\mathcal{X},\omega) = \sum_{q\in\mathcal{Q}} \sum_{a\in\mathcal{A}} \Bigg\{ & \left( \sum_{x\in\mathcal{X}_q^a} x_k e^{f_\omega(x)} \right) \left( \sum_{\substack{b\in\mathcal{A} \\ \mathcal{X}_q^b \prec \mathcal{X}_q^a}} \sum_{x\in\mathcal{X}_q^b} e^{-f_\omega(x)} \right) \\ & + \left( \sum_{x\in\mathcal{X}_q^a} e^{f_\omega(x)} \right) \left( \sum_{\substack{b\in\mathcal{A} \\ \mathcal{X}_q^b \prec \mathcal{X}_q^a}} \sum_{x\in\mathcal{X}_q^b} -x_k e^{-f_\omega(x)} \right) \Bigg\} \end{aligned}. \tag{8}$$

The complexity for computing the gradient is the same ($O(K \cdot |\mathcal{Q}| \cdot |\mathcal{X}|)$) as that of (7).

### 3.5 Incorporation of unlabeled samples

The natural way to incorporate an unlabeled sample $y$ would be to compute all probabilities $P(y \prec x)$ for $x$ in $\mathcal{X}$. But this method would be computationnaly costly because we could not use property 7 to reduce the complexity since $y$ is not in a particular $\mathcal{X}_q^a$.

A better method is to affect $y$ to only one $\mathcal{X}_q^a$, using a certain probability of belonging. We say that a sample belongs to the group with which it has the maximum probability of indifference:

$$P(y \in \mathcal{X}_q^a) = P(\{y\} \perp \mathcal{X}_q^a) = \prod_{x \in \mathcal{X}_q^a} P(y \perp x) = \prod_{x \in \mathcal{X}_q^a} P(y \prec x) P(x \prec y)$$

where $P(y \perp x)$ is the probability that there is no preference between $x$ and $y$.

If we use the exponential ranking loss, we will choose the group $\mathcal{X}_q^a$ which minimize the ranking loss:

$$\exp(f_\omega(y)) \sum_{x \in \mathcal{X}_q^a} \exp(-f_\omega(x)) + \exp(f_\omega(-y)) \sum_{x \in \mathcal{X}_q^a} \exp(f_\omega(x)).$$

Note this is not ordinal regression because there is some groups ($\mathcal{X}_q$'s) which are each other indifferent and if we could know a priori if an unlabeled sample belongs to one of this group. Even if our model produce a total order on $\mathcal{X}$, there is some comparisons without sense (for instance, we see in section 3.3 that comparing search results from different queries is none sense).

We sum up the semi-supervised model in the following algorithm:

**Algorithm 1**

```
1. Minimize the ranking loss on labeled samples.
2. Repeat until convergence:
    3. Affect each unlabeled sample to a group $\mathcal{X}_q^a$
       according to the minimum ranking loss:
       $\exp(f_\omega(y)) \sum_{x \in \mathcal{X}_q^a} \exp(-f_\omega(x)) + \exp(f_\omega(-y)) \sum_{x \in \mathcal{X}_q^a} \exp(f_\omega(x))$.
    4. Minimize the ranking loss on labeled and unlabeled samples.
```

## 4 Experiments

### 4.1 Learning base

The Wikipedia collection [7] has been used with a small set of three queries. Then we made 300 assessments for these three queries to be used as a learning base. We collected these assessments using the following method :

```
1. the system is initialized with equal preferences
   between node types and using only one
   preference relation on doxels' content
2. Repeat several times:
   3. compute the results
   4. re-order 25 not assessed results
   5. learn with the new partial order provided by user assessments
```

### 4.2 Filtering

In CO-Focussed task, overlapping doxels were not allowed. In order to suppress all overlapping elements from the lists computed by the ranking algorithm, we used a strategy which consists in removing all elements which are overlapping with an element ranked higher in the list.

As for Okapi model, we used the same strategy exept that biggest doxels like articles or bdy's were not allowed in the final ranking list to reach better performance.

### 4.3 Results

We comment here the results obtained with the nxCG official metric with generalized quantization which is more related to the ranking loss criterion and the different levels of assessment we have used in our model.

**CO-Focussed.** We have plotted in figures 1 and 2 the evaluation of the lists produced by the ranking algorithms and by the modified Okapi for CO-Focussed task. We evaluate the model taking overlap into account (overlap on) or not (overlap off). We can see that the ranking algorithms perform better than Okapi, but semi-supervised model has not been able to increase the performance of the ranking algorithm.

Table 1 and 2 show that the ranking models are always better than its baseline Okapi model, and that is quite good to retrieve the most informative doxels in the begining of the list comparing to other INEX participan approaches.

**Table 1.** Rank of Okapi and ranking models among all participant submissions using nxCG metric with generalised quantization and overlap on for CO-Focussed task

| | @5 | @10 | @25 | @50 |
|---|---|---|---|---|
| Ranking | 2 | 1 | 2 | 3 |
| Ranking semi-supervised | 14 | 12 | 12 | 8 |
| Okapi | 50 | 39 | 37 | 32 |

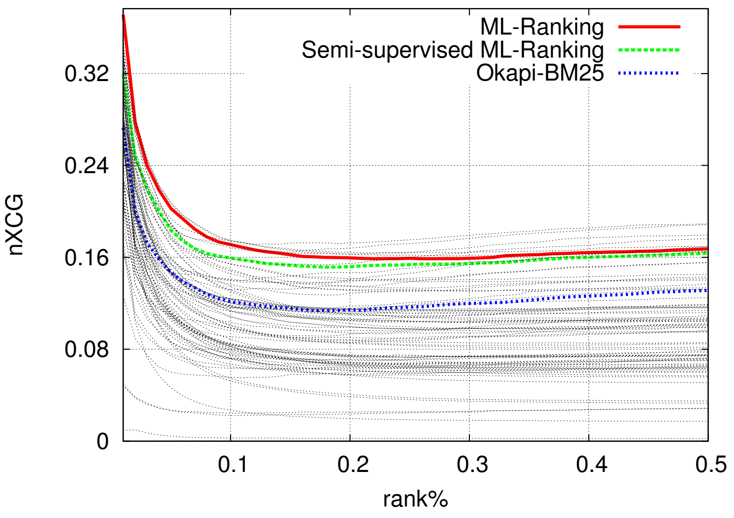

**Fig. 1.** Performance of ranking and Okapi models for CO-Focussed task evaluated with the cumulated gain based metric ncXG with overlap on.

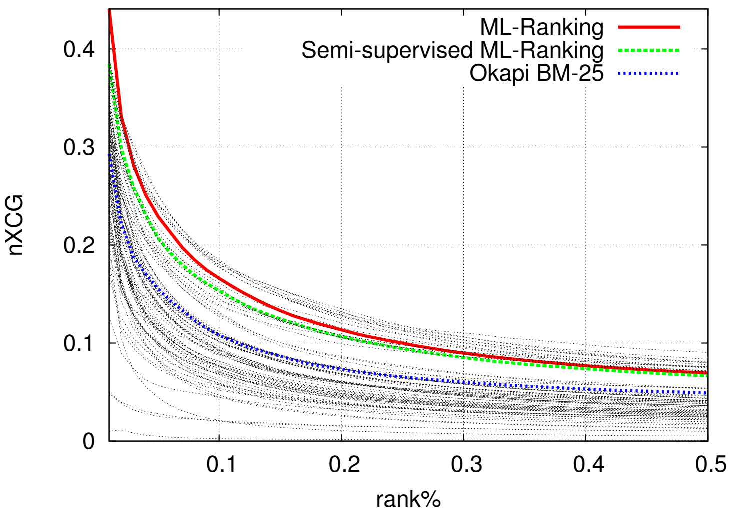

**Fig. 2.** Performance of ranking and Okapi models for CO-Focussed task evaluated with the cumulated gain based metric ncXG with overlap off.

**Table 2.** Rank of Okapi and ranking models among all participant submissions using nxCG metric with generalised quantization and overlap off for CO-Focussed task

| | @5 | @10 | @25 | @50 |
|---|---|---|---|---|
| Ranking | 1 | 1 | 4 | 8 |
| Ranking semi-supervised | 1 | 7 | 9 | 11 |
| Okapi | 49 | 39 | 40 | 33 |

**CO-Thorough.** Figure 3 show the evaluation of the lists produced by the ranking algorithm and modified Okapi where overlap was not removed. We can see that the ranking algorithms performs clearly better than Okapi but the semi-supervised model do not perform better than the semi-supervised model.

Table 3 shows that the ranking models are always better than their baseline Okapi model, and that is quite good to retrieve the most informative doxels in the begining of the list. This can be explained by the expression of the ranking loss which highly penalize an irrelevant doxel when it is located in the begining of the list.

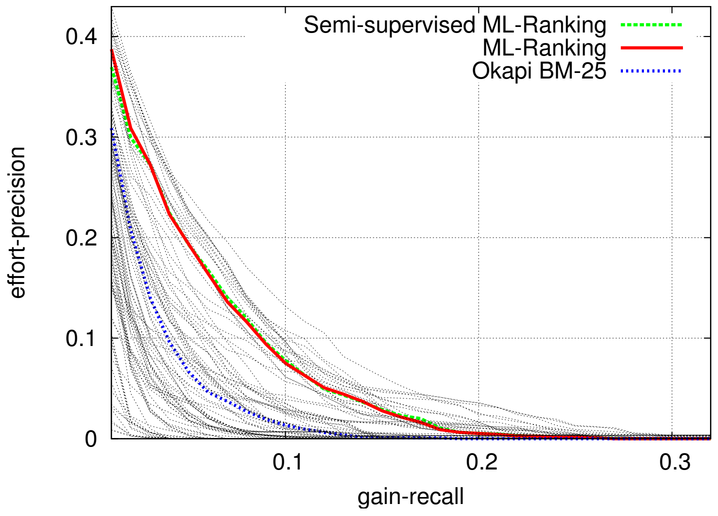

**Fig. 3.** Performance of ranking and Okapi models for CO-Thorough task evaluated with the cumulated gain based metric ncXG.

For all experiments, the ranking algorithm has been able to increase the performance of the baseline Okapi. Ranking methods thus appear as a promising direction for improving SIR search engine performance. It remains to perform tests with additional features (for example the scores of additional IR systems).

**Table 3.** Rank of Okapi and ranking models among all participant submissions using nxCG metric for CO-Thorough task

| | rank | score |
|---|---|---|
| Ranking | 18 | 0.0281 |
| Ranking semi-supervised | 18 | 0.0281 |
| Okapi | 27 | 0.0186 |

## 5 Conclusion

We have described our ranking model for CO tasks which relies on a combination of scores from the Okapi model and takes into account the document structure. This score combination is learned from a training set by a ranking algorithm.

For both tasks, the ranking algorithm has been able to increase by a large amount the performance of the baseline Okapi with a very small set of labeled examples. Ranking methods thus appear as a promising direction for improving SIR search engine performance. It remains to perform tests with additional features (for example the scores of additional IR systems).

Eventually, there is also work in progress to make unlabeled data useful for Ranking algorithms.

## References

1. Amini, M.R., Usunier, N., Gallinari, P.: Automatic text summarization based on word-clusters and ranking algorithms. In: ECIR. (2005) 142–156
2. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. In Jordan, M.I., Kearns, M.J., Solla, S.A., eds.: Advances in Neural Information Processing Systems. Volume 10., The MIT Press (1998)
3. Bartell, B.T., Cottrell, G.W., Belew, R.K.: Automatic combination of multiple ranked retrieval systems. In: Research and Development in Information Retrieval. (1994) 173–181
4. Clémençon, S., Lugosi, G., Vayatis, N.: Ranking and scoring using empirical risk minimization. In Auer, P., Meir, R., eds.: COLT. Volume 3559 of Lecture Notes in Computer Science., Springer (2005) 1–15
5. Craswell, N., Robertson, S., Zaragoza, H., Taylor, M.: Relevance weighting for query independent evidence. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference. (2005)
6. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via EM algorithm. Journal of the Royal Statistical Society, Vol. B, n°39 (1977) 1-38
7. Denoyer L., Gallinari P.: The Wikipedia XML Corpus SIGIR Forum (2006)
8. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. In: Proceedings of ICML-98, 15th International Conference on Machine Learning. (1998)
9. Miller, D., Uyar, H.: A Mixture of Experts classifier with learning based on both labeled and unlabeled data. Advances in Neural Information Processing Systems 9 (1996) 571–577

10. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM. In Proceedings of National Conference on Artificial Intel-ligence (1998)
11. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC. In: Text REtrieval Conference. (1992) 21–30
12. Vittaut, J.N., Piwowarski, B., Gallinari, P.: An algebra for structured queries in bayesian networks. In: Advances in XML Information Retrieval. Third Workshop of the INitiative for the Evaluation of XML Retrieval. (2004)
13. Vittaut, J.N., Amini M.R., Gallinari, P.: Learning Classification with Both Labeled and Unlabeled Data. In: ECML '02: Proceedings of the 13th European Conference on Machine Learning. (2002)