A Boosting Algorithm for Learning Bipartite Ranking Functions with Partially Labeled Data

Massih-Reza Amini, Tuong-Vinh Truong
Université Pierre et Marie Curie
Laboratoire d'Informatique de Paris 6
104, avenue du Président Kennedy
75016 Paris, France
{amini,truong}@poleia.lip6.fr

Cyril Goutte
National Research Council Canada
Institute for Information Technology
283, boulevard Alexandre-Taché
Gatineau, QC J8X 3X7, Canada
Cyril.Goutte@nrc-cnrc.gc.ca

ABSTRACT

This paper presents a boosting based algorithm for learning a bipartite ranking function (BRF) with partially labeled data. Until now different attempts had been made to build a BRF in a *transductive* setting, in which the test points are given to the methods in advance as unlabeled data. The proposed approach is a semi-supervised *inductive* ranking algorithm which, as opposed to transductive algorithms, is able to infer an ordering on new examples that were not used for its training. We evaluate our approach using the TREC-9 `Ohsumed` and the `Reuters`-21578 data collections, comparing against two semi-supervised classification algorithms for ROCArea (`AUC`), uninterpolated average precision (`AUP`), mean precision@50 (`mT9P`) and Precision-Recall (`PR`) curves. In the most interesting cases where there are an unbalanced number of irrelevant examples over relevant ones, we show our method to produce statistically significant improvements with respect to these ranking measures.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - Information Routing; H.1 [**Models and Principles**]: Miscellaneous

General Terms

Algorithms, Experimentation, Theory

Keywords

Learning to Rank with partially labeled data, Boosting, Information Routing

1. INTRODUCTION

Learning with partially labeled data or semi-supervised learning has been widely studied under the classification framework [5, 13, 16, 20, 3, 11, 8]. However, there are many applications where the goal is to learn a ranking function and for which training sets are hard to obtain mainly because the labeling of examples is time consuming, and sometimes even unrealistic. This is for example the case for domain specific search engines or routing systems [18] where, for a stable user's information need there is a stream of incoming documents which have to be dynamically retrieved. The constitution of a labeled training set is, in this case, a difficult task. But, if such a training set is available the goal of learning would be to find a bipartite ranking function (BRF) which assigns a higher score to relevant examples than to irrelevant ones[1].

Recently some studies have addressed the use of unlabeled data to learn a BRF in a transductive setting [1, 22, 24, 23]. In this case, one is given sample points from a labeled training set and an unlabeled test set, and the goal is to build a prediction function which orders *only* unlabeled examples from the test set. This restriction makes the design of a ranking function that assigns scores to new examples inherently inconvenient. Neither can existing work on leveraging unlabeled data for classification of pairwise examples be applied directly to learn a BRF. As some pairs would have the same instances in common, this would violate the independence assumption on which classifier learning is based.

In this paper we present a new inductive ranking algorithm which builds a prediction function on the basis of two labeled and unlabeled training sets: one labeled and one unlabeled. In a first stage, the algorithm loops over the labeled set and assigns, for each labeled training example, the same relevance judgment to the most similar examples from the unlabeled training set. An extended version of the RankBoost algorithm [10] is then developed to produce a scoring function that minimizes the average number of incorrectly ordered pairs of (relevant, irrelevant) examples, over the labeled training set and the tentatively labeled part of the unlabeled data. The novelty of the approach is that the algorithm optimizes an exponential upper bound of a learning criterion which combines the misordering loss for both parts of the training set.

Experiments on the TREC-9 `Ohsumed` and `Reuters`-21578 datasets, show that the proposed approach is effective on `AUC`, `AUP`, `mT9P` and `PR` ranking measures especially when there are much less relevant examples than irrelevant ones. In addition, comparisons involving two semi-supervised clas-

SIGIR'08, July 20–24, 2008, Singapore.

[1]These forms of ranking correspond to the bipartite feedback case studied in [10] and we hence refer to them as the *bipartite* ranking problems.

sification algorithms indicate that, as in the supervised case [7, 9], the error rate of a semi-supervised classification function is not necessarily a good indicator of the accuracy of the ranking function derived from it.

In the remainder of the paper, we discuss, in section 2, the problem of semi-supervised learning for bipartite ranking. In sections 2.1 and 2.2, we present a boosting based algorithm to find such ranking functions. In section 4, we present experimental results obtained with our approach on the TREC-9 `Ohsumed` and the `Reuters`-21578 datasets. Finally, in section 5 we discuss the outcomes of this study and give some pointers to further research.

2. SEMI-SUPERVISED BIPARTITE RANKING

In bipartite ranking problems such as information routing [17], examples from an instance space $\mathcal{X} \in \mathbb{R}^d$ are assumed to be relevant or irrelevant to a given search profile or topic. The goal of learning can then be defined as the search of a scoring function $H : \mathcal{X} \to \mathbb{R}$ which assigns higher scores to relevant instances than to irrelevant ones [10].

In this setting, the system is usually given a sample of n labeled instances $\mathcal{Z}_\ell = \{x_i, y_i\}_{i\in\{1,\dots,n\}}$ where each example $x \in \mathcal{Z}_\ell$ is associated with a relevance judgment $y_i \in \{-1, 1\}$, representing the fact that x_i is relevant ($y_i = +1$) or not ($y_i = -1$) to that topic. The learning task reduces to minimization of the average number of irrelevant instances in $\mathcal{Z}_\ell$ scored better than relevant ones by H [10]. In the semi-supervised setting, we further assume that together with labeled examples in $\mathcal{Z}_\ell$, we also have a set of m unlabeled examples $X_\mathcal{U} = \{x'_i\}_{i\in\{n+1,\dots,n+m\}}$ and, we propose to learn a ranking function on the basis of these two training sets.

In order to exploit information from $X_\mathcal{U}$, we assume that an unlabeled instance from $X_\mathcal{U}$ that is similar to a labeled instance from $\mathcal{Z}_\ell$ should have similar label. We begin by selecting examples in $X_\mathcal{U}$ that are the most similar to a labeled example $x \in \mathcal{Z}_\ell$ and assign them the corresponding relevance judgment y. At this stage, a simple approach would consist in adding these new examples from $X_\mathcal{U}$ to the labeled training set and then learn a ranking function as in the usual supervised case. This training scheme suffers from a real drawback in that as unlabeled examples are given error-prone labels, the resulting ranking would be highly dependent on how robust the training scheme is to noisy labels. Therefore, instead of mixing $\mathcal{Z}_\ell$ with selected examples in $X_\mathcal{U}$, we suggest minimizing the ranking errors on each training sets separately.

Formally, let $\mathcal{A}_\mathcal{C}$ be an unsupervised algorithm which specifies, for each labeled data, the unlabeled instances that are the most similar to it. And, let $\mathcal{Z}_{\mathcal{U}'}$ be the set of unlabeled examples obtained from $\mathcal{A}_\mathcal{C}$ that have been assigned labels according to similar labeled data. Our goal is thus to find a function H, which minimizes the average numbers of irrelevant examples scored better than relevant ones in $\mathcal{Z}_\ell$ and $\mathcal{Z}_{\mathcal{U}'}$ separately. We call this quantity the *average ranking loss*, $\hat{R}_{n+m}$, defined as:

$$\hat{R}_{n+m}(H, \mathcal{Z}_\ell \cup \mathcal{Z}_{\mathcal{U}'}) = \frac{1}{P_{\mathcal{Z}_\ell}} \sum_{x_+,x_- \in \mathcal{X}_+\times\mathcal{X}_-} [\![H(x_-) \geq H(x_+)]\!] + \frac{\lambda}{P'_{\mathcal{Z}_{\mathcal{U}'}}} \sum_{x'_+,x'_- \in \mathcal{X}'_+\times\mathcal{X}'_-} [\![H(x'_-) \geq H(x'_+)]\!]$$

Where, $\mathcal{X}_+$ and $\mathcal{X}_-$ (resp.) represent the sets of relevant and irrelevant (resp.) instances in $\mathcal{Z}_\ell$, while $\mathcal{X}'_+ = \mathcal{A}_\mathcal{C}(\mathcal{X}_+)$ and $\mathcal{X}'_- = \mathcal{A}_\mathcal{C}(\mathcal{X}_-)$ (resp.) are the sets of unlabeled data which are similar to relevant and irrelevant (resp.) instances in $\mathcal{Z}_\ell$. $P_{\mathcal{Z}_\ell} = |\mathcal{X}_-||\mathcal{X}_+|$ and $P'_{\mathcal{Z}_{\mathcal{U}'}} = |\mathcal{X}'_-||\mathcal{X}'_+|$ (resp.) are the number of relevant and irrelevant pairs in $\mathcal{Z}_\ell$ and $\mathcal{Z}_{\mathcal{U}'}$ (resp.). (x_+, x_-) and (x'_+, x'_-) (resp.) denote pairs of (relevant, irrelevant) examples in $\mathcal{X}_+ \times \mathcal{X}_-$ and $\mathcal{X}'_+ \times \mathcal{X}'_-$ (resp.). Finally, λ is a discount factor[2] and $[\![\pi]\!]$ is equal to 1 if the predicate π holds and 0 otherwise.

Several successful Machine Learning algorithms, including various versions of boosting and support vector machines are based on replacing the loss function by a convex function [4]. This approach has important computational advantages, as the minimization of the empirical convex functional is feasible by, for example, gradient descent algorithms. In the following we extend the majoration of the supervised ranking loss proposed by [10] to the semi-supervised bipartite ranking case. Using the upper bound $[\![x \geq 0]\!] \leq e^x$, we get the following exponential loss:

$$\mathcal{E}_{n+m}(H) = \frac{1}{P_{\mathcal{Z}_\ell}} \sum_{x_+,x_-} e^{H(x_-)-H(x_+)} + \frac{\lambda}{P'_{\mathcal{Z}_{\mathcal{U}'}}} \sum_{x'_+,x'_-} e^{H(x'_-)-H(x'_+)}$$

As for the meta-search task [21] and following [10], we define the final ranking function, $H = \sum_t \alpha_t f_t$, as a weighted sum of the ranking features f_t. Each ranking feature f_t is uniquely defined by an input feature $j_t \in \{1, ..., d\}$ and a threshold θ_t:

$$f_t(x) = \begin{cases} 1, & \text{if } \varphi_{j_t}(x) > \theta_t \\ 0, & \text{else} \end{cases} \quad (1)$$

where, $\varphi_j(x)$ is the j^{th} feature characteristic of x.

The learning task is then defined as the search of the combination weights α_t and the ranking features f_t for which the minimum of the exponential loss $\mathcal{E}_{n+m}(H)$ is reached.

2.1 A Boosting based Algorithm Derivation

The ranking problem presented in the previous section can be implemented efficiently by extending the RankBoost algorithm proposed in [10]. This extension iteratively maintains two distributions D_t and $\tilde{D}_t$ over pairs of (relevant, irrelevant) examples in $\mathcal{Z}_\ell$ and $\mathcal{Z}_{\mathcal{U}'}$.

At the beginning, all pairs are supposed to be uniformly distributed, that is, $\forall(x_-, x_+) \in X_- \times X_+, D_1(x_-, x_+) = \frac{1}{P_{\mathcal{Z}_\ell}}$ and $\forall(x'_-, x'_+) \in X'_- \times X'_+, \tilde{D}_1(x'_-, x'_+) = \frac{1}{P'_{\mathcal{Z}_{\mathcal{U}'}}}$. At each round, D_t and $\tilde{D}_t$ are then gradually updated in order to give increasing weights to pairs that are difficult to rank correctly.

The weight for each pair is therefore increased or decreased depending on whether f_t orders that pair incorrectly, leading to the update rules:

$$D_{t+1}(x_-, x_+) = \frac{D_t(x_-, x_+) \exp(\alpha_t(f_t(x_-) - f_t(x_+)))}{Z_t} \quad (2)$$

and

$$\tilde{D}_{t+1}(x'_-, x'_+) = \frac{\tilde{D}_t(x'_-, x'_+) \exp(\alpha_t(f_t(x'_-) - f_t(x'_+)))}{\tilde{Z}_t} \quad (3)$$

[2]For $\lambda = 0$, we fall back to the situation of standard supervised learning.

where $Z_t = \sum_{x_-,x_+} D_t(x_-,x_+)e^{\alpha_t(f_t(x_-)-f_t(x_+))}$ and $\tilde{Z}_t = \sum_{x'_-,x'_+} D_t(x'_-,x'_+)e^{\alpha_t(f_t(x'_-)-f_t(x'_+))}$ are normalization factors such that D_{t+1} and $\tilde{D}_{t+1}$ remain probability distributions.

The search of the ranking feature f_t and its associated weight α_t are carried out by directly minimizing the exponential loss, $\mathcal{E}_{n+m}$. This optimization is performed first by noticing that the exponential loss $\mathcal{E}_{n+m}$ writes:

$$\mathcal{E}_{n+m}(H) = \prod_{t=1}^{T} Z_t + \lambda \prod_{t=1}^{T} \tilde{Z}_t, \quad (4)$$

where, T is the maximum number of rounds. This rewriting results from the update rules (2-3), the use of a linear ranker $H = \sum_{t=1}^{T} \alpha_t f_t$, the exponential homomorphism property $e^{x+y} = e^x e^y$ and the fact that D_t and $\tilde{D}_t$ sum to one for every t.

The selection of ranking features is presented in section 2.2. The choice of the weight combinations, α_t, results from the minimization of (4). At each iteration, this minimization is performed by rewriting the exponential loss as

$$\mathcal{E}_{n+m}(H) = A_{t-1}Z_t + \lambda B_{t-1}\tilde{Z}_t, \quad (5)$$

where, $A_{t-1} = \prod_{k=1}^{t-1} Z_k$ and $B_{t-1} = \prod_{k=1}^{t-1} \tilde{Z}_k$. Eq. 5 can then be upper-bounded by the following expression:

$$\begin{aligned}\mathcal{E}_{n+m}(H) \quad \leq \quad & A_{t-1}\left[\left(\frac{1-r_t}{2}\right)e^{\alpha_t} + \left(\frac{1+r_t}{2}\right)e^{-\alpha_t}\right] \\ & +\lambda B_{t-1}\left[\left(\frac{1-\tilde{r}_t}{2}\right)e^{\alpha_t} + \left(\frac{1+\tilde{r}_t}{2}\right)e^{-\alpha_t}\right],\end{aligned}$$

where, $r_t = \sum_{x_-,x_+} D_t(x_-,x_+)(f_t(x_+) - f_t(x_-))$ and $\tilde{r}_t = \sum_{x'_-,x'_+} \tilde{D}_t(x'_-,x'_+)(f_t(x'_+) - f_t(x'_-))$.

This follows from Jensen's inequality and the convexity of $e^{\alpha x}$, which yields $e^{\alpha x} \leq (\frac{1+x}{2})e^{\alpha} + (\frac{1-x}{2})e^{-\alpha}$.

The right-hand side of the above inequality is minimized when:

$$\alpha_t^* = \frac{1}{2}\ln\frac{A_{t-1}(1+r_t) + \lambda B_{t-1}(1+\tilde{r}_t)}{A_{t-1}(1-r_t) + \lambda B_{t-1}(1-\tilde{r}_t)} \quad (6)$$

Plugging this back into the inequality yields

$$\mathcal{E}_{n+m}(H) \leq \sqrt{(A_{t-1}+\lambda B_{t-1})^2 - (A_{t-1}r_t + \lambda B_{t-1}\tilde{r}_t)^2} \quad (7)$$

The complexity of the algorithm, if implemented with distributions D_t and $\tilde{D}_t$, is $O(|X_-||X_+| + |X'_-||X'_+|)$. As in the supervised case, it is possible to reduce this complexity to $O(n + |X'_-| + |X'_+|)$ by setting

$$D_t(x_-,x_+) = \nu_t(x_-)\nu_t(x_+) \quad (8)$$

$$\tilde{D}_t(x'_-,x'_+) = \tilde{\nu}_t(x'_-)\tilde{\nu}_t(x'_+) \quad (9)$$

Where ν_t and $\tilde{\nu}_t$ are two sets of weights over respectively $\mathcal{Z}_\ell$ and $\mathcal{Z}_{\mathcal{U}}$.

Thanks to the homomorphism property of the exponential, we have $Z_t = Z_t^- . Z_t^+$, with $Z_t^- = \sum_{x_- \in \mathcal{Z}_\ell} \nu_t(x_-)e^{\alpha_t f_t(x_-)}$ and $Z_t^+ = \sum_{x_+ \in \mathcal{Z}_\ell} \nu_t(x_+)e^{-\alpha_t f_t(x_+)}$, and similarly for $\tilde{Z}_t$. As a consequence, eqs. (8) and (9) are preserved by the update rules for D_t and $\tilde{D}_t$ (eqs. 2 and 3), and hold on round $t+1$.

Algorithm 1: Learning BRF with partially labeled data

Given :
- $\mathcal{Z}_\ell = X_+ \cup X_-$, a labeled training set and $\mathcal{Z}_{\mathcal{U}} = X'_+ \cup X'_-$, a labeled subset of $X_{\mathcal{U}}$ obtained from $\mathcal{A}_{\mathcal{C}}$,
- Set $\nu_1(x) = \frac{1}{|X_+|}$ if $x \in X_+$ and $\nu_1(x) = \frac{1}{|X_-|}$ if $x \in X_-$,
- Set $\tilde{\nu}_1(x') = \frac{1}{|X'_+|}$ if $x' \in X'_+$ and $\tilde{\nu}_1(x') = \frac{1}{|X'_-|}$ if $x' \in X'_-$
- Set $A_0 = 1$ and $B_0 = 1$.

for $t := 1, ..., T$ **do**

- Train a ranking feature $f_t : \mathcal{X} \to \mathbb{R}$ using ν_t and $\tilde{\nu}_t$
- Choose the weight α_t, as defined by equation (6)
- Update

$$\nu_{t+1}(x) \quad = \quad \begin{cases} \frac{\nu_t(x)\exp(-\alpha_t f_t(x))}{Z_t^+} \text{ if } x \in X_+ \\ \frac{\nu_t(x)\exp(\alpha_t f_t(x))}{Z_t^-} \text{ if } x \in X_- \end{cases}$$

  where Z_t^+ and Z_t^- normalize ν_t over X_+ and X_-
- Update

$$\tilde{\nu}_{t+1}(x') \quad = \quad \begin{cases} \frac{\tilde{\nu}_t(x')\exp(-\alpha_t f_t(x'))}{\tilde{Z}_t^+} \text{ if } x' \in X'_+ \\ \frac{\tilde{\nu}_t(x')\exp(\alpha_t f_t(x'))}{\tilde{Z}_t^-} \text{ if } x' \in X'_- \end{cases}$$

  where $\tilde{Z}_t^+$ and $\tilde{Z}_t^-$ normalize $\tilde{\nu}_t$ over X'_+ and X'_-
- Update

$$A_t \leftarrow A_{t-1}Z_t^- Z_t^+$$
$$B_t \leftarrow B_{t-1}\tilde{Z}_t^- \tilde{Z}_t^+$$

end

Output : The final ranking function $H = \sum_{t=1}^{T} \alpha_t f_t$

The pseudocode for this implementation is given in algorithm 1, where we have stated the update rules in terms of ν and $\tilde{\nu}$ instead of D and $\tilde{D}$. At each iteration of the algorithm, first a ranking feature f_t and its associated weight α_t are chosen in order to minimize the empirical exponential loss $\mathcal{E}_{n+m}$. Then ν_t and $\tilde{\nu}_t$ are updated, and finally the coefficients A_t and B_t are estimated for the calculation of the loss (5) in the next round.

2.2 Learning ranking features

In this section, we work for a given t and therefore drop t from the notation.

As in the supervised case [10], ranking features f_t can be learned efficiently in a greedy manner. Indeed, since each ranking feature is $\{0,1\}$-valued (equation 1), learning reduces to the search in the discrete space of a feature characteristics j and thresholds θ of the minimum of the upper-bound (7). This is equivalent to maximizing $|Ar + \lambda B\tilde{r}|$.

Algorithm 2: Semi-supervised learning of ranking features

Given :
- Two sets of weights ν and $\tilde{\nu}$ over respectively $\mathcal{Z}_\ell$ and $\mathcal{Z}_\mathcal{U}$
- A set of features $\{\varphi_j\}_{j=1}^d$,
- For each φ_j a set of thresholds $\{\theta_k\}_{k=1}^K$ such that $\theta_1 \geq .... \geq \theta_K$,
- A, B and λ,
- Set $r^* = 0$.

for $j := 1, ..., d$ **do**
- $L \leftarrow 0$
- **for** $k := 1, ..., K$ **do**
  - $L \leftarrow L + A \sum_{x:\varphi_j(x)\in[\theta_{k-1},\theta_k[} y\nu(x) + \lambda B \sum_{x':\varphi_j(x')\in[\theta_{k-1},\theta_k[} \tilde{y}'\tilde{\nu}(x')$
  - **if** $|L| > |r^*|$ **then**
    - $r^* \leftarrow L$
    - $j^* \leftarrow j$
    - $\theta^* \leftarrow \theta_k$
  - **end**
- **end**

end

Output : (φ_j^*, θ^*)

Let us first rewrite $Ar + \lambda B\tilde{r}$ in terms of ν, $\tilde{\nu}$ and f:

$$\begin{aligned} Ar + \lambda B\tilde{r} &= A\sum_{x_+}\sum_{x_-}\nu(x_+)\nu(x_-)(f(x_+) - f(x_-)) \\ &+ \lambda B\sum_{x'_+}\sum_{x'_-}\tilde{\nu}(x'_+)\tilde{\nu}(x'_-)(f(x'_+) - f(x'_-)) \end{aligned}$$

As f is $\{0,1\}$-valued and ν (resp. $\tilde{\nu}$) sums to one on each subset X_-(resp. X'_-) and X_+ (resp. X'_+), this equation can be rewritten as

$$Ar + \lambda B\tilde{r} = A\sum_{x|\varphi_j(x)>\theta} y\nu(x) + \lambda B\sum_{x'|\varphi_j(x')>\theta} y'\tilde{\nu}(x')$$

The search algorithm for the candidate ranking feature f^* is described in algorithm 2. For each feature characteristic $j \in \{1, ..., d\}$, the algorithm incrementally evaluates $|Ar + \lambda B\tilde{r}|$ on a sorted list of candidate thresholds $\{\theta_k\}_{k=1}^K$ and stores the values j^* and θ^* for which $|Ar + \lambda B\tilde{r}|$ is maximal. Thus a straightforward implementation of this algorithm requires $O((n + |X'_-| + |X'_+|) \times K \times d)$ time to generate a ranking feature.

3. EXPERIMENT SETUP

We conducted a number of experiments aimed at evaluating how unlabeled data can help to learn an efficient bipartite ranking function. To this end, we ran two versions of the supervised RankBoost (RB) algorithm [10]. The first one uses the labeled training set only: this provides a baseline which we hope to outperform thanks to the unlabeled data. The second uses both the labeled training set as well as the unlabeled training set *with their true labels*: this provides an upper bound on the achievable performance, as the latter labels are not available in the semi-supervised setting. This comparison gives a first insight into the contribution of unlabeled data for learning a BRF.

We also compared our algorithm with two semi-supervised algorithms proposed in the classification framework. The first one is the so-called transductive SVM (TSVM) implemented in *SVMlight* [13]. The second is an EM-like algorithm which was successfully applied to extractive document summarization [2]. It operates by first training a logistic regression (LR) classifier on the labeled training set, then outputs of the classifier are used to estimate class labels for unlabeled data and a new classifier is learnt on the basis of both the labeled data and these newly labeled instances. These two steps (labeling and learning) are iterated until a local maxima of the complete data likelihood is reached. We refer to this second algorithm as semi-supervised LR, or ssLR.

The unsupervised algorithm $\mathcal{A}_\mathcal{C}$ used for the constitution of $\mathcal{Z}_\mathcal{U}$ was the nearest neighbors (`NN`) algorithm. For each example in the labeled training set $\mathcal{Z}_\ell$, we assigned the same label to k of its nearest neighbors in $X_\mathcal{U}$. The choice of the `NN` algorithm for $\mathcal{A}_\mathcal{C}$ here is essentially motivated by its computational efficiency. We refer to our proposed algorithm by ssRB which stands for semi-supervised RankBoost.

Finally, each experiment is performed over 10 random splits (labeled training/unlabeled training/test) sets of the initial collection.

3.1 Data Sets

We conducted our experiments on TREC-9 `Ohsumed` and `Reuters`-21758 datasets. Following TREC-9 filtering tasks, the selected topic categories in `Reuters`-21758 served as filtering topics. We shall now describe the corpora and methodology.

3.1.1 Ohsumed

The `Ohsumed` document collection [12] is a set of 348,566 articles from the on-line medical information database (MEDLINE) consisting of titles and abstracts from 270 journals over a period of 5 years (1987 to 1991). We carried out our experiments on 63 topics defined for the routing track of the TREC-9 Filtering tasks [17]. The number of relevant documents varies from 5 to 188 with an average of 59.8 relevant documents per topic. We indexed documents having an abstract (with an existing `.W` field - this represents 233,445 documents) and took terms appearing in the title, abstract as well as human assigned MeSH indexing terms. All words were converted to lowercase, digits were mapped to a single *digit* token and non alpha-numeric characters were suppressed. We also used a stop-list to remove very frequent words and also filtered terms occurring in less than 3 documents.

3.1.2 Reuters

The `Reuters`-21578 collection contains Reuters news articles from 1987 [15]. We selected documents in the collection that are assigned to at least one topic. Each document in the corpus can have multiple labels, but in practice more than 80% of articles are associated to a single topic. In addition, for multiply-labeled documents, only the first topic from the <TOPIC> field was retained.

We only considered documents associated with the 10 most frequent topics, which resulted in 9509 documents, each with a unique label. We carried out the same pre-

Table 1: The Reuters topics used in our experiments.

Topic	# of documents	Proportion (%)
earn	3972	41.77
acq	2423	25.48
money-fx	682	7.17
crude	543	5.71
grain	537	5.64
trade	473	4.98
interest	339	3.56
ship	209	2.89
money-su	177	1.87
sugar	154	1.67

processing as for the Ohsumed data set. The distribution of the number of relevant documents per topic, given in table 1, varies from 1.67% to 41.77%. These relatively populous topics will allow us to test the behavior of bipartite ranking algorithms when relevant documents are gradually removed from well-represented topics.

3.2 Evaluation Criteria

In order to compare the performance of the algorithms we used a set of standard ranking measures.

As the learning criterion (4) we used to train our model is related to the area under the ROC curve (AUC), we first compared the AUC measure of each algorithm on the test set. If a sample T contains p relevant and m irrelevant instances, the AUC of a scoring function h with respect to this sample represents the average number of relevant examples in T ranked higher than irrelevant ones [6]:

$$\texttt{AUC}(h,T) = \frac{1}{pm} \sum_{i:y_i=+1} \sum_{j:y_j=-1} [\![h(x_i) > h(x_j)]\!]$$

In addition, we computed the mean average uninterpolated precision (mAUP) and the mean precision@50 (referred as mT9P [17] in the following) across topics on both datasets.

The average uninterpolated precision (AUP) of a given topic τ is defined as the sum of precision value of relevant documents in the r top ranked documents divided by the number of relevant documents for that topic, $\mathcal{R}(\tau)$. Hence, relevant documents which do not appear in the top r ranked documents receive a precision score of 0:

$$\texttt{AUP}(\tau) = \frac{1}{\mathcal{R}(\tau)} \sum_{i=1;y_i=+1}^{r} \frac{|\{j \mid y_j = 1 \wedge rank(j) \leq rank(i)\}|}{rank(i)}$$

We used $r = 500$ on the Reuters dataset, and $r = 1000$ on the Ohsumed collection[17].

Finally, we plot the Precision/Recall curves [19] of different ranking algorithms for a fixed percentage of labeled-unlabeled documents in the training set. At each recall level, the precision score is averaged over all topics.

Each reported performance value is the average over the 10 random splits.

4. EXPERIMENTAL RESULTS

These experimental results test how unlabeled data affect the ranking performance of the proposed approach, vs. both of the semi-supervised classification algorithms.

4.1 The effect of unlabeled data

We start our evaluation by analyzing the impact of unlabeled data on mean average uninterpolated precision (mAUP) and mean precision@50 (mT9P) for a fixed number of labeled and unlabeled examples in the training set of each topic in Ohsumed and Reuters datasets. In order to effectively study the role of unlabeled data on the ranking behavior we begin our experiments with very few labeled training examples. For Ohsumed, the size of the labeled training sets is hence fixed to 180 documents per topic: 3 relevant and 177 irrelevant. For Reuters, we use 90 documents per topic: 9 relevant and 81 irrelevant documents. The remaining documents from the collection are used as unlabeled data. The lower proportion of relevant documents in our Ohsumed experiments reflect the higher imbalance between relevant and irrelevant documents in that collection. As discussed later in section 4.4, the value of the discount factor λ which provided the best ranking performance for these training sizes is $\lambda = 1$. We therefore use that value in our experiments in the three coming sections.

Table 2 summarizes results obtained by RB, TSVM, ssLR and ssRB in terms of mAUPand mT9P. The RankBoost algorithm is trained over the labeled part of each training sets and performance of ssRB are shown for different number k of nearest neighbors of each labeled example which are assigned the same relevance judgment. We use bold face to indicate the highest performance rates. The symbol $\downarrow$ indicates that performance is significantly worse than the best result, according to a Wilcoxon rank sum test used at a p-value threshold of 0.01 [14].

Three observations can be made from these results. First, all semi-supervised algorithms perform better on both collection, and according to both metrics, than the RB algorithm trained using only the labeled data alone. This shows empirically that semi-supervised algorithms are able to partially exploit the relevant information contained in the unlabeled examples. The second observation is that the hyperparameter k has a definitive (although not highly significant) impact on the performance of the ssRB algorithm. On Ohsumed, the best results for ssRB are obtained when only one unlabeled instance nearest to each labeled example is labeled ($k = 1$). On Reuters, the best results are obtained for $k = 2$. This may be due to the fact that the NN algorithm is less effective on the Ohsumed collection, where the dimension of the document space is about 7 times higher than on the Reuters dataset. Performance of the ssRB on both datasets is lowest for $k = 3$, suggesting that this value of k yields too many erroneous label assignment that ssRB is unable to overcome. Finally, both TSVM and ssLR, which have been shown to be effective on classification tasks [13, 2], are less competitive than ssRB on both ranking measures.

4.2 Ranking on unbalanced data

Table 3 gives the AUC performance of the ranking algorithms on each topic of the Reuters dataset. For ssRB, the NN parameter was fixed to $k = 2$, in agreement with results from the previous section. The number of labeled examples in the training set for each of the topics is the same as above. These results show that ssRB is significantly better than TSVM and ssLR on topics presenting a higher disproportion of relevant/irrelevant examples in the collection, and not significantly worse than the best on others.

Table 2: mAUP and mT9P measures on the Ohsumed and Reuters document collections. The number of labeled examples per topic is (a) 180 on the Ohsumed collection (3 relevant and 177 irrelevant documents), and (b) 90 on the Reuters collection (9 relevant document and 81 irrelevant documents). All remainig documents were used as unlabeled training set.

	Ohsumed collection		Reuters data set	
Algorithm	mAUP(%), $r = 1000$	mT9P(%)	mAUP(%), $r = 500$	mT9P(%)
RB	$23.5 \pm 0.3^{\downarrow}$	$33.6 \pm 0.2^{\downarrow}$	$40.85 \pm 0.6^{\downarrow}$	$64.51 \pm 0.4^{\downarrow}$
TSVM	$26.2 \pm 0.2^{\downarrow}$	$36.4 \pm 0.1^{\downarrow}$	$51.6 \pm 0.3^{\downarrow}$	$71.01 \pm 0.2^{\downarrow}$
ssLR	$25.3 \pm 0.1^{\downarrow}$	$35.2 \pm 0.2^{\downarrow}$	$50.23 \pm 0.5^{\downarrow}$	$70.83 \pm 0.1^{\downarrow}$
ssRB, $k = 1$	**30.3±0.2**	**40.4±0.4**	57.32 ± 0.3	74.82 ± 0.1
ssRB, $k = 2$	28.9 ± 0.1	38.6 ± 0.3	**59.36±0.4**	**76.57±0.5**
ssRB, $k = 3$	27.2 ± 0.5	36.5 ± 0.2	57.6 ± 0.2	73.21 ± 0.3

Referring back to table 2, we can also see that on mAUP and mT9P measures, the relative margin between the performance of the best ssRB and TSVM is greater on Ohsumed than on Reuters. We recall that the proportion of relevant documents per topic is sizably higher on Reuters than on Ohsumed. For example, considering mT9P scores, the difference in percentage between ssRB (for $k = 1$) and TSVM on the Ohsumed collection is 4% which represents 58.8% of the interval length between the worse and best mT9P performance [33.6, 40.4]. While, the difference on performance between the best ssRB (for $k = 2$) and TSVM on the Reuters dataset is 5.56% which represents 46.1% of the interval length [64.5, 76.5].

We further investigate the effect on the AUC measure of having less and less relevant documents in a labeled training pool where the number of irrelevant examples is kept fixed. Figure 1 illustrates this effect by showing the evolution on AUC scores of the three semi-supervised algorithms when relevant documents are gradually removed from the labeled part of the acq training set. This topic was, with earn, one of the two topics in Reuters on which ssRB did worse than the two other semi-supervised algorithms. The initial number of relevant/irrelevant documents in the labeled training set was set to respectively 9 and 81.

These curves show that the decrease rate of the AUC measure of ssRB is lower than the two other semi-supervised classifiers. The loss in AUC for ssRB is less than 9% when the proportion of relevant/irrelevant documents falls from 1/9 to 1/27. This drop is about 16% for TSVM and ssLR.

Figure 2, top, shows precision/recall curves on Ohsumed and Reuters collections using the same number of relevant/irrelevant documents in the respective training sets than what was used in previous experiments. In order to have an empirical upper-bound on these results we also plotted the precision/recall curves of a rankboost algorithm trained over all labeled and unlabeled examples, plus their true labels, in the different training sets. We refer to this model as RB-Fully supervised. These results confirm the previous ones as for different precision levels, the relative margin between ssRB and TSVM (or ssLR) is higher on Ohsumed than on Reuters specially for low recall rates. An explanation of these findings is that as ssRB learns a scoring function which output is supposed to rank relevant documents higher than irrelevant ones in both labeled and unlabeled training sets, its performance is less affected by fewer relevant documents.

4.3 The exponential value of labeled data

We finally report on the behavior of different ranking algorithms for growing number of labeled data in the training sets. Figure 2 down, illustrates this behavior on the mAUP measures for both of datasets. The addition of new

Table 3: AUC measure on the 10 largest topics of the Reuters dataset for RB, TSVM, ssLR and ssRB. Semi-supervised algorithms use 90 labeled examples per topic. The remaining documents in the training set are all unlabeled.

	RB	TSVM	ssLR	ssRB
earn	$85.6 \pm 0.7^{\downarrow}$	**95.9±0.1**	95.4 ± 0.3	94.8 ± 0.1
acq	$81.3 \pm 0.6^{\downarrow}$	91.6 ± 0.2	**92.1±0.3**	91.5 ± 0.3
money-fx	$83.8 \pm 0.4^{\downarrow}$	$90.3 \pm 0.5^{\downarrow}$	$89.7 \pm 0.4^{\downarrow}$	**92.8±0.7**
crude	$83.4 \pm 0.5^{\downarrow}$	94.5 ± 0.4	93.5 ± 0.4	**95.5±0.2**
grain	$84.5 \pm 0.4^{\downarrow}$	$91.1 \pm 0.6^{\downarrow}$	$92.4 \pm 0.3^{\downarrow}$	**93.1±0.1**
trade	$84.9 \pm 0.6^{\downarrow}$	$91.2 \pm 0.3^{\downarrow}$	$90.4 \pm 0.4^{\downarrow}$	**92.4±0.5**
interest	$79.9 \pm 0.6^{\downarrow}$	$87.6 \pm 0.7^{\downarrow}$	$88.3 \pm 0.2^{\downarrow}$	**90.5±0.4**
ship	$81.2 \pm 0.2^{\downarrow}$	$85.1 \pm 0.4^{\downarrow}$	$!84.3 \pm 0.4^{\downarrow}$	**89.7±0.3**
money-su	$80.2 \pm 0.3^{\downarrow}$	$86.2 \pm 0.3^{\downarrow}$	$87.3 \pm 0.1^{\downarrow}$	**91.3±0.2**
sugar	$78.6 \pm 0.1^{\downarrow}$	$86.6 \pm 0.2^{\downarrow}$	$85.3 \pm 0.6^{\downarrow}$	**90.3±0.4**

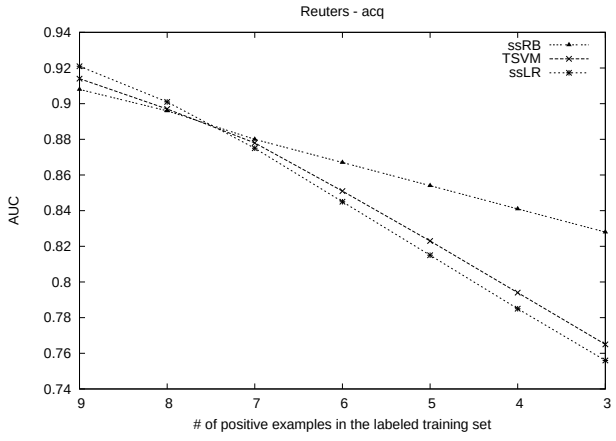

Figure 1: AUC on Reuters category acq with respect to the number of relevant documents in the labeled training set. The number of irrelevant documents is fixed kept to 81.

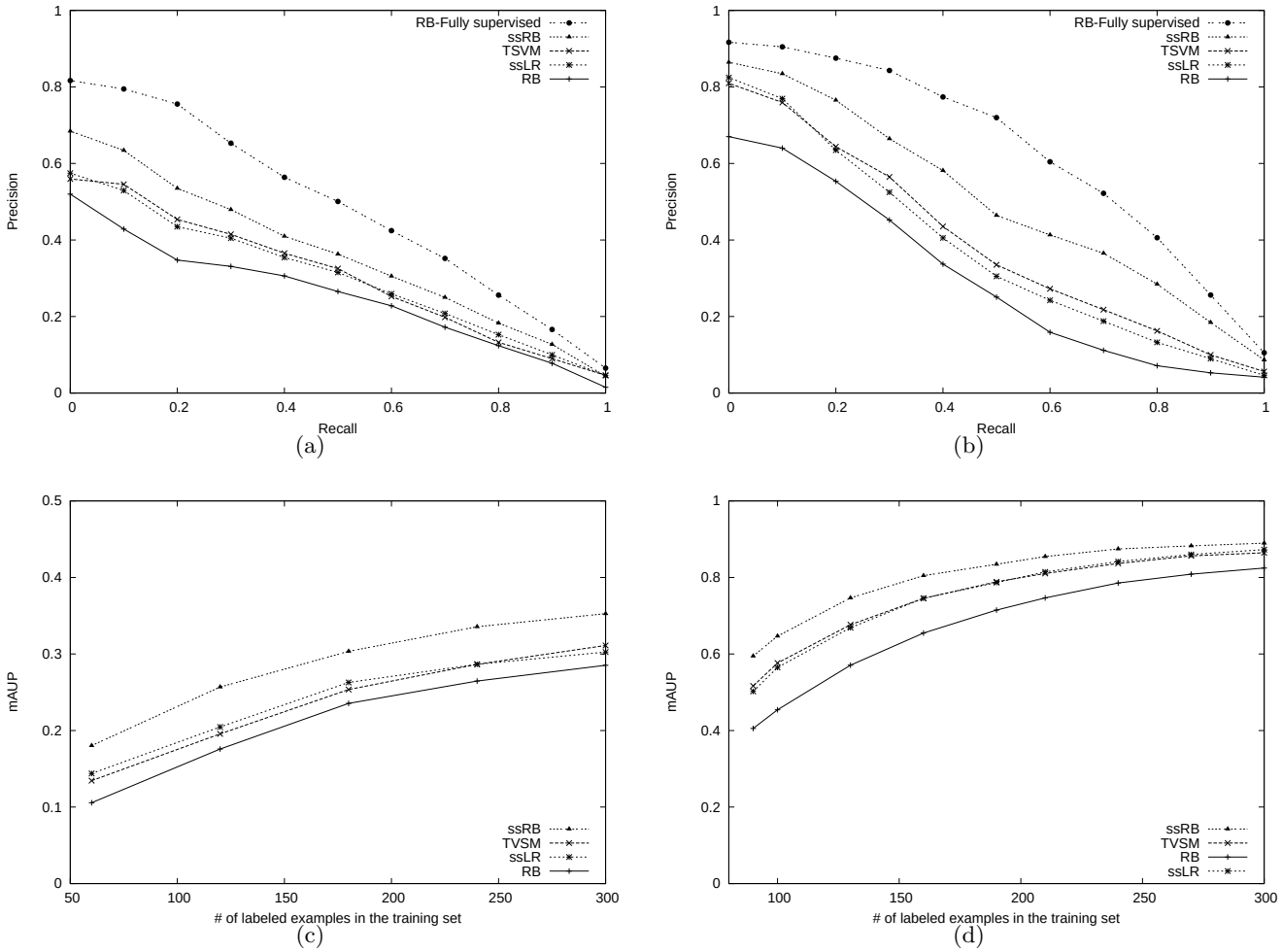

Figure 2: (Top) Precision recall curves on (a) `Ohsumed` and (b) `Reuters` collections. The number of labeled examples per topic is fixed to 180 on the `Ohsumed` collection with a ratio of 1 relevant document for every 59 irrelevant ones, and 90 on the `Reuters` data set with a ratio of 1 relevant document for every 9 irrelevant ones. (Bottom) `mAUP` on (c) `Ohsumed` and (d) `Reuters` with respect to different labeled training size.

labeled data on each training sets respects the initial relevant/irrelevant proportion of documents on these sets. All performance curves increase monotonically with respect to the additional labeled data. The convergence rate of `mAUP` scores on the `Reuters` dataset is however faster. We expect that this is because training sets per topic on this collection contain more relevant information than `Ohsumed` topics.

4.4 The effect of the discount factor λ

Another interesting observation here is that unlabeled examples become relatively less important as more labeled data are available. This observation is confirmed with results on figure 3 which show that for growing number of labeled training size on the `Ohsumed` collection, the discount factor λ for which a maximum is reached on `mAUP` moves away from 1. We recall that for $\lambda = 1$, unlabeled data play the same role in the training of the scoring function than labeled data.

5. CONCLUSIONS

We presented a new approach to learn a boosting based, inductive ranking function in a semi-supervised setting, when both labeled and unlabeled data are available. We showed that unlabeled data do help provide a more efficient ranking function and that their effect depends on the initial labeled training size. In the most interesting case, when there are few labeled data, we empirically illustrated that unlabeled data have a large effect on the `mAUP` measure.

We evaluated and validated our approach using two test collections. We showed through different ranking measures that the new model provides results that are superior to two semi-supervised classification algorithms on both collections. Among other things, these results confirm theoretical and empirical studies made previously in the supervised case, showing that the classification ability of classifiers is not necessarily a good indication of their ranking performance.

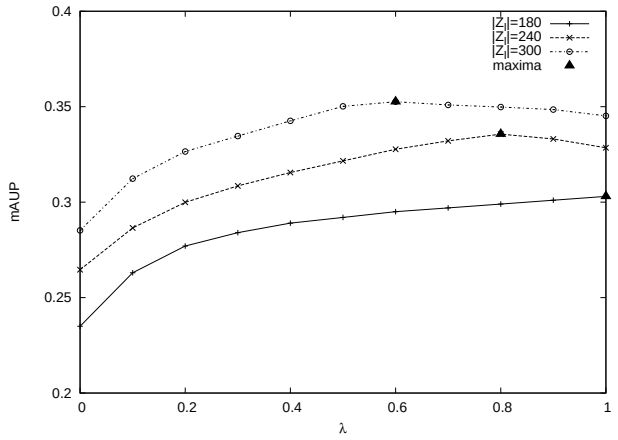

Figure 3: mAUP with respect to the discount factor λ for different labeled training sizes on Ohsumed.

In future work, we will make the discount factor λ depend on each unlabeled example in the training set using a continuous function. The key of that study would be the choice of the function with necessary conditions allowing to take into consideration information contained on unlabeled data as best as possible. Another promising direction to explore would be the optimization of other ranking criterion than the modified AUC for learning ranking functions.

6. ACKNOWLEDGMENTS

This work was supported in part by the IST Program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

7. REFERENCES

[1] S. Agarwal. Ranking on Graph Data. In *Proceedings of the* 23^{rd} *International Conference on Machine Learning*, pages 25-32, 2006.

[2] M.-R. Amini and P. Gallinari. The Use of Unlabeled Data to Improve Supervised Learning for Text Summarization. In *Proceedings of the* 25^{th} *annual international ACM SIGIR*, pages 105-112, 2002.

[3] M.-R. Amini and P. Gallinari. Semi-Supervised Learning with Explicit Misclassification Modeling. In *Proceedings of the* 18^{th} *International Joint Conference on Artificial Intelligence*, pages 555-560, 2003.

[4] P. L. Bartlett, M. I. Jordan and Jon D. McAuliffe. Large Margin Classifiers: convex Loss, Low Noise and Convergence Rates. In *Advances in Neural Information Processing Systems 16*, pages 1173-1180, 2004.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with Co-training. In *Proceedings of the Workshop on Computational Learning Theory*, pages 92-100, Wisconsin, USA, 1998.

[6] A.P. Bradley. The use of the Area under the ROC curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, 30:1145-1159, 1997.

[7] R. Caruana and A. Niculescu-Mizil. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the* 10^{th} *ACM SIGKDD*, pages 69-78, 2004.

[8] O. Chapelle, B. Schölkopf and A. Zien. *Semi-Supervised Learning*, MIT Press, Cambridge, MA, 2006

[9] C. Cortes and M. Mohri. AUC optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems 16*, pages 313-320, 2004.

[10] Y. Freund, R. Iyer, R.E. Schapire and Y. Singer. An efficient Boosting Algorithm for Combining Preferences. *Journal of Machine Learning Research*, 4:933-969, 2004.

[11] E. Gaussier and C. Goutte. Learning with Partially Labelled Data - with Confidence. In *ICML'05 Workshop on Learning from Partially Classified Training Data (ICML'05-LPCT)*, pages 29-36, 2005.

[12] W. Hersh, C. Buckley, T. J. Leone and David Hickam. OHSUMED: an interactive Retrieval Evaluation and new Large Test Collection for Research. In *Proceedings of the* 17^{th} *annual international ACM SIGIR*, pages 192-201, 1994.

[13] T. Joachims. Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings of* 16^{th} *International Conference on Machine Learning*, pages 200-209, 1999.

[14] E.L. Lehmann. *Nonparametric Statistical Methods Based on Ranks*. McGraw-Hill, New York, 1975.

[15] D. D. Lewis. Reuters-21578, distribution 1.0 http://www.daviddlewis.com/resources /testcollections/reuters21578/. January 1997.

[16] K. Nigam, A.K. McCallum, S. Thrun and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning* 39(2/3):127-163, 2000.

[17] S. Robertson and D.A. Hull. The TREC-9 Filtering Track Final Report. In *Proceedings of the* 9^{th} *Text REtrieval Conference (TREC-9)*. pages 25-40, 2001.

[18] I. Soboroff and S. Robertson. Building a Filtering Test Collection for TREC 2002. In *Proceedings of the* 26^{th} *annual international ACM SIGIR*. pages 243-250, 2003.

[19] C. van Rijsbergen, *Information Retrieval*, Butterworths, London, 1979.

[20] J.-N. Vittaut, M.-R. Amini and P. Gallinari, Learning Classification with Both Labeled and Unlabeled Data. In *Proceedings of the* 13^{th} *European Conference on Machine Learning*. pages 468-476, 2002.

[21] C.C. Vogt and G.W. Cottrell, Fusion Via a Linear Combination of Scores. *Information Retrieval*, 1(3):151-173, 1999.

[22] J. Weston, R. Kuang, C. Leslie and W.S. Noble. Protein ranking by semi-supervised network propagation. *BMC Bioinformatics*, special issue, 2006.

[23] D. Zhou, J. Weston, A. Gretton, O. Bousquet and B. Schölkopf. Ranking on Data Manifolds. In *Advances in Neural Information Processing Systems 16*, pages 169-176, 2004.

[24] D. Zhou , C.J.C. Burges and T. Tao. Transductive link spam detection. In *Proceedings of the* 3^{rd} *international workshop on Adversarial information retrieval on the web*. pages 21-28, 2007.