Mach Learn (2010) 79: 105–121
DOI 10.1007/s10994-009-5151-5

A co-classification approach to learning from multilingual corpora

Massih-Reza Amini · Cyril Goutte

Received: 26 February 2009 / Revised: 14 September 2009 / Accepted: 23 September 2009 /
Published online: 29 October 2009

Abstract We address the problem of learning text categorization from a corpus of multilingual documents. We propose a multiview learning, co-regularization approach, in which we consider each language as a separate source, and minimize a joint loss that combines monolingual classification losses in each language while ensuring consistency of the categorization across languages. We derive training algorithms for logistic regression and boosting, and show that the resulting categorizers outperform models trained independently on each language, and even, most of the times, models trained on the joint bilingual data. Experiments are carried out on a multilingual extension of the RCV2 corpus, which is available for benchmarking.

Keywords Text categorization · Multilingual data · Logistic regression · Boosting

1 Introduction

In this paper we consider the problem of boosting the performance of multiple monolingual document categorizers by using a corpus of multilingual documents. In addition, we investigate the more specific situation where Machine Translation is used to produce a parallel corpus.

In many contexts, people are confronted with documents available in more than one language. This is a typical situation in many multilingual regions of the world, including many regions of Europe and, for example, most legal and regulatory documents in Canada. However, document categorization models are mostly developed in a monolingual context, typically for resource-rich languages such as English.

The situation we are addressing is when documents are available in two (or more) languages and share the same set of categories. In that case, it is obviously possible to train

Editors: Nicolo Cesa-Bianchi, David R. Hardoon, and Gayle Leen.

M.-R. Amini (✉) · C. Goutte
Interactive Language Technologies group, National Research Council Canada, 283, boulevard
Alexandre-Taché, Gatineau, QC J8X 3X7, Canada
e-mail: Massih-Reza.Amini@cnrc-nrc.gc.ca

independent monolingual categorizers on each part of the corpus. However this approach ignores the potentially richer information available from another language, and may be impractical when the number of available documents in the different languages is very uneven. The challenge is actually to develop a method which is able to leverage the multilingual data in order to produce performance that is higher than what one gets from the independent monolingual categorizers alone.

We focus on the situation where each document is available in two languages, i.e. two linguistic sources. In the particular regulatory context of Canada, this is a very common situation. It seems likely to be equally typical in many national or supra-national contexts (such as the European Union). We propose to learn statistical categorizers by optimizing a joint loss with an additional constraint on the divergence between the outputs produced on the different languages. We therefore minimize the classification loss for both classifiers under the constraint that their outputs are as similar as possible on documents and their translations. We show that this produces a significant increase in performance over independent categorizers trained on monolingual data, and over bilingual categorizers trained on the concatenated views.

We insist on the fact that our goal is to obtain a number of categorizer that each work on *monolingual* data. Although we assume that the training set contains parallel documents, at test time, we only use monolingual documents. In addition, although our method relies on a *parallel corpus*, we later relax this assumption to the more general case of a partly parallel or even *comparable corpus*, i.e. a set of texts that deal with the same topics (e.g. Adeva et al. 2005), without being translations of each other.

As an aside, we also investigate the use of Machine Translation (MT) in a multilingual categorization setting. In principle, a translation is supposed to contain the same information as the original, and therefore may not be very helpful to improve categorization. However, in the context of usual categorization models, which typically rely on bag-of-words or similar feature spaces with short-range dependencies, translation offers the possibility to enrich and disambiguate the text, especially for short documents. There have been renewed expectations regarding the usefulness of MT lately. We show that although a long way from being totally fluent, state-of-the-art statistical MT can indeed be used in a document categorization context and improve the categorization decision. In order to encourage further work on multilingual text categorization, we also release publicly a preprocessed version of the 186K multilingual Reuters documents and translations that were used in our experiments.

The following section relates our work to various previously described methods. In Sect. 3, we introduce the model and the estimation procedure. Section 4 will present the experimental settings and the results. We discuss these results and conclude in Sects. 5 and 6.

2 (Un)related work

Let us first position our work with respect to various existing methods.

First, there are several contexts in which multilingual document categorization may be invoked, which are in fact very different from the problem we address here. One such setting is *cross-language* text categorization (CLTC, Bel et al. 2003), the categorization analogue to *cross-language information retrieval* (CLIR, Oard and Diekema 1998). In CLTC, a large monolingual corpus is available in one resource-rich (typically English) language, and a document in another language must be categorized in the same set of categories. For example, a news item written in Maltese must be categorized in a news hierarchy learned from

English documents. This is a very exciting problem, however, in our setting, we assume that documents are available in both languages, and we are interested in learning improved monolingual categorizers in each of the different languages, i.e. categorizers that predict categories directly from a monolingual document.

Another totally unrelated problem that is sometimes referred to as multilingual text categorization is the task of finding which language a text is *written in* (e.g. English, Maltese, Afrikaans, etc.) (Cavnar and Trenkle 1994). This is more properly known as *language identification* or *language guessing* and is widely considered a solved problem except in fringe situations (such as very short sentences). In our context we usually know which languages our documents are written in, and the target categories are not language related. In fact, categories are by definition of the problem identical across all languages.

Multiview learning has become a popular research topic in the past years. Two important classes of techniques are the multiple kernel learning approach (e.g. Bach et al. 2004), and techniques relying on (kernel) Canonical Correlation Analysis (CCA, e.g. Farquhar et al. 2005; Kakade and Foster 2007). Multiple kernel learning typically assumes that all views of an example are available during training and testing, for example the same object viewed from different angles in object recognition. By contrast, we address a problem where multiple views (i.e. original or translation) may or may not be available in the training set, and we are interested in learning classifiers that can predict based on one language version of a document, i.e. without requiring to translate documents before testing.

CCA identifies matching maximally-correlated subspaces in the two views, that may be used to project data before learning is performed, or integrated with learning (Farquhar et al. 2005). The CCA subspaces may then be used at test time in either, or both, views. Note that CCA does not explicitely attempt to enforce agreement in the outputs of the classifiers obtained on each subspaces. Multiview Fisher Discriminant Analysis (Diethe et al. 2008) addresses that issue, and seems closer to the approach we follow here. However we have not considered this technique here due to concerns about the computational complexity and scalability to large document collections.

Our *co-classification* algorithm is in fact an instance of co-regularization (Sindhwani et al. 2005; Brefeld et al. 2006). One key difference is that instead of regularizing the disagreement between the classifiers in the two views by the squared error, we use the KL divergence. In addition to having a natural interpretation in terms of probabilistic classifier output, that allows us to naturally propose a boosting version of the co-classification approach. Our method for solving the co-regularized classification problem is also different from the co-regularized least squares described by Sindhwani et al. (2005) and Brefeld et al. (2006).

3 Model

We consider two input spaces $\mathcal{X}_1 \subset \mathbb{R}^d$ and $\mathcal{X}_2 \subset \mathbb{R}^p$, and an output space $\mathcal{Y}$. We take $\mathcal{Y} = \{-1, +1\}$ since we restrict our presentation to binary classification (we will address the extension to multiclass in Sect. 5.3). We assume that we have a set of m *independently identically distributed* labeled bilingual documents, $\{(x_i^{(1)}, x_i^{(2)}, y_i); i = 1 \ldots m\}$, sampled from a fixed but unknown distribution $\mathcal{P}$ over $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}$. Input vector $x^{(1)}$ is the feature vector representing a document in one language, while $x^{(2)}$ is the feature vector representing the same document in another language, and y is the class label associated to the document. The two versions of the same document are typically translations of each other, although which direction the translation goes is not important for our purpose.

Each language offers a different view on the same document, and we can form two monolingual training sets, $S_1 = \{(x_i^{(1)}, y_i); i = 1 \dots m\} \in (\mathcal{X}_1 \times \mathcal{Y})^m$ and $S_2 = \{(x_i^{(2)}, y_i); i = 1 \dots m\} \in (\mathcal{X}_2 \times \mathcal{Y})^m$. Note that for a given i, label y_i is the same in both sets as both versions of the same document cover the same topic.

The problem we address is to construct two classifiers $f_1 : \mathcal{X}_1 \to \mathcal{Y}$ and $f_2 : \mathcal{X}_2 \to \mathcal{Y}$ from S_1 and S_2 so that a test document written in either language may be classified as accurately as possible. Of course, it is possible to independently train f_1 on S_1 and f_2 on S_2. Our goal is therefore to propose an algorithm that results in classifiers that are more efficient than if they were trained separately on the monolingual data.

3.1 The co-classification algorithm

Our basic assumption is that a document and its translated version convey the same idea but in different ways. The difference is mostly due to the fact that the expression of an idea in each language makes use of different words. Our aim here is to take advantage of these two complementary *views* of the same information to train two different classifiers. In addition, as both views of a document have matching labels, we want the output of the classifiers working on either view to be in agreement. Our learning paradigm expresses this idea by relying on:

- A monolingual *misclassification* cost for each classifier in each language/view,
- A *disagreement* cost to constrain decisions to be similar in both languages.

More precisely, we look for functions f_1 and f_2 which not only achieve good performance on the training set in their respective language, but also agree with each other. In the following, we assume that classifiers f_1 and f_2 have corresponding underlying real-valued functions h_1 and h_2 (e.g. output of a SVM or probability for a generative model), and are obtained by thresholding using the `sign` function; $f_1 = \text{sign}(h_1)$ and $f_2 = \text{sign}(h_2)$.

Our framework relies on iteratively and alternately optimizing the classifier h from one view ($h = h_\ell, \ell \in \{1, 2\}$), view while holding the classifier from the other view ($h^* = h_{3-\ell}$) fixed. This is done by minimizing a monolingual classification loss in that view, regularized by a divergence term which constrains the output of the trained classifier to be similar to that of the classifier previously learned in the other view. Without loss of generality, let us now describe the stage where we optimize functions h from one view, while leaving the function from the other view, h^*, fixed. Following the principle stated above, we seek the function h that minimizes the following objective function:

$$\mathcal{L}(h, S, h^*, S^*, \lambda) = \mathcal{C}(h, S) + \lambda d(h, S, h^*, S^*). \tag{1}$$

Where $\mathcal{C}(h, S)$ is the (monolingual) cost of h on training set S, $d(h, S, h^*, S^*)$ measures the divergence between the two classifiers on the same documents in both views and λ is a discount factor which modulates the influence of the disagreement cost on the optimization.

For the monolingual cost, we consider the standard misclassification error:

$$\mathcal{C}(h, S) = \frac{1}{m} \sum_{i=1}^{m} [\![y_i h(x_i) \le 0]\!],$$

where $[\![\pi]\!]$ is equal to 1 if the predicate π is true, and 0 otherwise. We usually replace it with an appropriate convex and differentiable proxy instead. Following standard practice in

Machine Learning algorithms, we replace $[\![z \leq 0]\!]$ by the (convex and differentiable) upper bound $a\log(1+e^{-z})$, with $a=(\log 2)^{-1}$. The monolingual misclassification cost becomes:

$$\mathcal{C}(h,S)=\frac{1}{m}\sum_{i=1}^{m} a\log(1+\exp(-y_i h(x_i))).$$

Assuming that each classifier output may be turned into a posterior class probability, we measure the disagreement between the output distributions for each view using the Kullback-Leibler (KL) divergence. Using the sigmoid function $\sigma(z)=(1+e^{-z})^{-1}$ to map the real-valued outputs of our functions h and h^* into a probability, and assuming that the *reference* distribution is the output of the classifier learned on the other view, h^*, the disagreement $d(h,S,h^*,S^*)$ becomes

$$d(h,S,h^*,S^*)=\frac{1}{m}\sum_{i=1}^{m} kl(\sigma(h^*(x_i^*))||\sigma(h(x_i))),$$

where for two binary probabilities p and q, the KL divergence is defined as:

$$kl(p||q)=p\log\left(\frac{p}{q}\right)+(1-p)\log\left(\frac{1-p}{1-q}\right).$$

There are two reasons for choosing the KL divergence: First, it is the natural equivalent in the classification context of the l_2 norm used for regression in previous work on co-regularization (Sindhwani et al. 2005; Brefeld et al. 2006; Rosenberg and Bartlett 2007); Second, it allows the derivation of a boosting approach for minimizing the local objective function (1), as described in Sect. 3.2. That objective function now becomes:

$$\mathcal{L}(h,S,h^*,S^*,\lambda)=\frac{1}{m}\sum_{i=1}^{m}\{a\log(1+\exp(-y_i h(x_i)))+\lambda kl(\sigma(h^*(x_i^*))||\sigma(h(x_i)))\}. \quad (2)$$

In the case where h is a linear function, $h(x)=\langle\beta,x\rangle$, it can be verified using calculus that the derivative of $\mathcal{L}(h,S,h^*,S^*,\lambda)$ with respect to parameters β is:

$$\nabla_\beta\mathcal{L}=\frac{1}{m}\sum_{x\in S} x\left(ay(\sigma(yh(x))-1)+\lambda(\sigma(h(x))-\sigma(h^*(x^*)))\right). \quad (3)$$

From the form of the derivative, it becomes apparent that the gradient is intimately related to the difference in classifier outputs, i.e., a large deviation $(\sigma(h(x))-\sigma(h^*(x^*)))$ makes the gradient larger in either direction.

The gradient from (3) can be plugged into any gradient-based minimization algorithm in order to obtain the linear weights which minimize $\mathcal{L}(h,S,h^*,S^*,\lambda)$. In the next section, we present the optimization of the cost function (1) as the minimization of a Bregman distance and show how this problem can be solved by a boosting-like algorithm.

Once the classifier h has been learned, we reverse the roles of h and h^* (as well as S and S^*), and optimize $\mathcal{L}(h^*,S^*,h,S,\lambda)$. This alternating optimization of partial cost functions bears similarity with the block-coordinate descent technique (Bertsekas 1999). At each iteration, block coordinate descent splits variables into two subsets, the set of the active variables and the set of inactive ones, then minimizes the objective function along active dimensions

Algorithm 1: The co-classification algorithm

```
Input  : Two labeled sets $S_1$ and $S_2$;
A discount factor $\lambda$.
Initialize: $t \leftarrow 1$;
$h_1^{(0)} \overset{def}{=} \text{argmin}_h\, \mathcal{C}(h, S_1)$;
$h_2^{(0)} \overset{def}{=} \text{argmin}_h\, \mathcal{C}(h, S_2)$;
repeat
    Learn $h_1^{(t)} = \text{argmin}_h\, \mathcal{L}(h, S_1, h_2^{(t-1)}, S_2, \lambda)$;
    Learn $h_2^{(t)} = \text{argmin}_h\, \mathcal{L}(h, S_2, h_1^{(t)}, S_1, \lambda)$;
    $t \leftarrow t + 1$;
until Convergence of $\Delta(h_1^{(t)}, S_1, h_2^{(t)}, S_2, \lambda)$ (eq. 4) to a local minimum ;
Output : $f_1 = \text{sign}(h_1^{(t)})$ and $f_2 = \text{sign}(h_2^{(t)})$
```

while inactive variables are fixed at current values. In our case, the global objective function is:

$$\Delta(h_1, S_1, h_2, S_2, \lambda) = \underbrace{\mathcal{C}(h_1, S_1) + \mathcal{C}(h_2, S_2)}_{\text{misclassification}} + \lambda \underbrace{D(h_1, S_1, h_2, S_2)}_{\text{disagreement}}. \tag{4}$$

Where $D(h_1, S_1, h_2, S_2) = d(h_1, S_1, h_2, S_2) + d(h_2, S_2, h_1, S_1)$ is the symmetrized KL divergence, measuring the corpus-level disagreement.

Notice that the symmetrized KL divergence is a convex function, with respect to the actual distributions on which the divergence is measured, but not necessarily with respect to the parameters of these distributions. Notice also that our algorithm is not *exactly* a block-coordinate descent technique: because of the asymmetry in the KL divergence used in (2), we only minimize an approximate version of the global loss at each iteration.

Algorithm 1 summarizes our overall training strategy, which we call *co-classification*. Each monolingual classifier is first initialized on the monolingual cost alone, then we alternate optimization of either h_1 or h_2 while keeping the other function constant, until $\Delta(h_1, S_1, h_2, S_2, \lambda)$ has reached a (possibly local) minimum.

We notice that this algorithm is reminiscent of the co-training algorithm (Blum and Mitchell 1998) in the sense that (1) we alternate between two views, and (2) the classifier that is learned in one view is affected by the output of the classifier learned in the other view, through the disagreement cost. Note however that each classifier does not change the *labeling* of examples, which is assumed to be fixed. However, by similarity with the alternating iterative process of learning a classifier on the basis of the decisions of another classifier, we refer to our proposed approach as co-classification.

In the following section, we extend the framework proposed by Collins et al. (2000) for learning h with a boosting-like algorithm which optimizes (2).

3.2 A boosting-like algorithm to learn the view-specific classifiers

In this section we present the loss-minimization of

$$\mathcal{R}(h, S, h^*, S^*, \lambda) = \frac{1}{m}\sum_{i=1}^{m}\{a \log(1 + \exp(-y_i h(x_i))) + \lambda kl(\sigma(h^*(x_i^*))||\sigma(h(x_i)))\} \tag{5}$$

as the minimization of a Bregman distance. This equivalence will allow us to employ the boosting-like parallel-update optimization algorithm proposed by Collins et al. (2000) to learn a linear classifier $h : x \mapsto \langle \beta, x\rangle$ minimizing (5).

A Bregman distance B_F of a convex, continuously differentiable function $F : \Omega \to \mathbb{R}$ on a set of closed convex set Ω is defined as

$$\forall p, q \in \Omega, \quad B_F(p||q) \overset{def}{=} F(p) - F(q) - \langle \nabla F(q), (p - q)\rangle.$$

One optimization problem arising from a Bregman distance is to find a vector $p^* \in \Omega$, closest to a given vector $q_0 \in \Omega$ with respect to B_F, under the set of linear constraints $\{p \in \Omega | p^t M = \tilde{p}^t M\}$, where, $\tilde{p} \in \Omega$ is a specified vector and M is a $n \times d$ matrix, with n the number of examples in the training set and d the dimension of the problem.[1]

Defining the Legendre transform as:

$$L_F(q, M\beta) \overset{def}{=} \underset{p\in\Omega}{\operatorname{argmin}}(B_F(p||q) + \langle M\beta, p\rangle),$$

the dual optimization problem can be stated as finding a vector q in the closure $\bar{Q}$ of the set $Q = \{L_F(q, M\beta) | \beta \in \mathbb{R}^p\}$, for which $B_F(\tilde{p}||q)$ is the lowest, under the set of linear constraints $\{q \in \Omega | q^t M = \tilde{p}^t M\}$.

It has been shown that both of these optimization problems have the same unique solution (Topsoe 1979; Csiszár 1995; Lafferty et al. 1999). Moreover, Collins et al. (2000) have proposed a single parallel-update optimization algorithm to find this solution in the dual form. They have further shown that their algorithm is a general procedure for solving problems which aim to minimize the exponential loss, like in Adaboost, or a log-likelihood loss, like in logistic regression. Indeed, they showed the equivalence of these two loss minimization problems in terms of Bregman distance optimization.

In order to apply Algorithm 2, we have to define a continuously differentiable function F such that by properly setting Ω, $\tilde{p}$, q_0 and M, the Bregman distance $B_F(0||L_F(q_0, M\beta))$ is equal to (5). Similarly to Collins et al. (2000), we choose:

$$\forall p \in \Omega = [0, 1]^n, \quad F(p) = \sum_{i=1}^{n} \alpha_i \left(p_i \log p_i + (1 - p_i)\log(1 - p_i)\right),$$

where α_i are non-negative real-valued weights associated to examples x_i. This yields:

$$\forall p, q \in \Omega \times \Omega, \quad B_F(p||q) = \sum_{i=1}^{n} \alpha_i \left(p_i \log\left(\frac{p_i}{q_i}\right) + (1 - p_i)\log\left(\frac{1 - p_i}{1 - q_i}\right)\right), \tag{6}$$

and

$$\forall i, \quad L_F(q, v)_i = \frac{q_i e^{-\frac{v_i}{\alpha_i}}}{1 - q_i + q_i e^{-\frac{v_i}{\alpha_i}}}. \tag{7}$$

[1] We have deliberately set the number of examples to n as in our equivalent rewriting of the minimization problem the latter is not exactly m.

Algorithm 2: The parallel-update optimization algorithm described in Collins et al. (2000)

```
Input      : Matrix $M \in [-1, 1]^{n \times d}$ .
Initialize: Let $\beta = 0$
for $t = 1, 2, \ldots$ do
    Let $q_t$ be the solution of $L_F(q_0, M\beta_t)$;
    for $j = 1, \ldots, d$ do
        $W^+_{t,j} = \sum_{i:\mathrm{sign}(M_{ij})=+1} q_{t,i} |M_{ij}|$;
        $W^-_{t,j} = \sum_{i:\mathrm{sign}(M_{ij})=-1} q_{t,i} |M_{ij}|$;
        $\delta_{t,j} = \frac{1}{2} \log \left( \frac{W^+_{t,j}}{W^-_{t,j}} \right)$;
    end
    $\beta_{t+1} = \beta_t + \delta_t$;
end
Output    : The sequence $\beta_1, \beta_2, \ldots$ verifying
```

$$\lim_{t \to \infty} B_F(0||L_F(q_0, M\beta_t)) = \inf_{\beta \in \mathbb{R}^d} B_F(0||L_F(q_0, M\beta_t))$$

Using (6) and (7), and setting $q_0 = \frac{1}{2}\mathbf{1}$, the vector with all components set to $\frac{1}{2}$, and M the matrix such that $\forall i, j, M_{ij} = \alpha_i y_i x_i^j$,[2] we have:

$$B_F(0||L_F(q_0, M\beta)) = \sum_{i=1}^{n} \alpha_i \log(1 + e^{-y_i \langle \beta, x_i \rangle}). \tag{8}$$

By developing (5), we get:

$$\begin{aligned}\mathcal{R}(h, S, h^*, S^*, \lambda) = \frac{1}{m} \sum_{i=1}^{m} \big\{ &\big(a + y_i \lambda \sigma(h^*(x_i^*)) + \lambda [\![y_i = -1]\!]\big) \log(1 + e^{-y_i h(x_i)}) \\ &+ \lambda([\![y_i = 1]\!] - y_i \sigma(h^*(x_i^*))) \log(1 + e^{y_i h(x_i)}) \big\} + K,\end{aligned} \tag{9}$$

where K does not depend on h.

In order to make (9) identical to (8) (up to a constant), we create, for each example (x_i, y_i), a new example $(x_i, -y_i)$ (which makes $n = 2m$), and set the weight as follows: for each example (x_i, y_i), take $\alpha_i = \frac{1}{m}(a + y_i \lambda \sigma(h^*(x_i^*)) + [\![y_i = -1]\!] \lambda)$, while for its counterpart $(x_i, -y_i)$, we set $\alpha_i = \frac{\lambda}{m}([\![y_i = 1]\!] - y_i \sigma(h^*(x_i^*))$.

As a consequence, minimizing (5) is equivalent to minimizing $B_F(0||q)$ over $q \in \bar{Q}$, where

$$Q = \{q \in [0, 1]^{2m} \mid q_i = \sigma(y_i \langle \beta, x_i \rangle), \beta \in \mathbb{R}^d\}.$$

This equivalence allows us to use Algorithm 2 for alternately optimizing each classifier within the general framework of Algorithm 1.

[2] All vectors $\forall i \in \{1, \ldots, n\}, \alpha_i y_i x_i$ should be normalized in order to respect the constraint $M \in [-1, 1]^{n \times d}$.

Table 1 Distribution of the number of documents and size of the vocabulary of the Reuters RCV2 data used in our experiments, across languages and categories

	Class proportions (%)						# docs	Voc. size
	C15	CCAT	E21	ECAT	GCAT	M11		
French	18.7	18.7	8.3	18.7	18.7	16.6	26,648	24,893
German	16.7	16.7	16.7	16.7	16.7	16.6	29,953	34,279
Italian	12.4	20.8	18.5	20.8	11.8	15.6	24,039	15,506
Spanish	5.9	17.0	6.7	17.5	12.2	40.5	12,342	11,547

4 Experiments

We conducted a number of experiments aimed at illustrating the effectiveness of our approach. These results will show how additional translated corpora can help to learn an efficient classifier under our multiview framework.

We first describe the data on which we ran the experiments, as well as the evaluation framework.

4.1 Data set

We conducted our experiments on a subset of the Reuters RCV2 collection (Reuters 2000). We used newswire articles written in 4 languages, French, German, Italian and Spanish and focused on 6 relatively populous classes: C15, CCAT, E21, ECAT, GCAT, M11 which are represented in all considered languages.

For each language and each class, we sampled up to 5000 documents from RCV2. Documents belonging to more than one of our 6 classes were assigned the label of their smallest class. This resulted in 12-30K documents per language (see Table 1), with between 728 and 5000 documents per category. We reserved a test split containing 75% of the documents (respecting class and language proportions) for testing. Each document from the corpus was translated to English using a state-of-the-art Statistical Machine Translation system developed at NRC (Ueffing et al. 2007), in order to produce 4 bilingual, parallel corpora on which we ran our experiments. Each parallel corpus contains documents with two views: the original document and its translation.

For each document, we indexed the text appearing in the title (*headline* tag), and the body (*body* tags) of each article. As preprocessing, we lowercased, mapped digits to a single digit token, and removed tokens with no alphanumeric characters. We also filtered out function words using a stop-list, as well as tokens occurring in less than 5 documents.

Documents were then represented as a bag of words, using a TFIDF weighting scheme based on BM25 (Robertson et al. 1994). The final vocabulary size for each language is given in the last column of Table 1 for the four source languages.

4.2 Evaluation criteria

In order to evaluate the classification performance of the various methods, we used the F_1 measure (Rijsbergen 1979). This measure combines Recall (Φ) and Precision (Π) in the following way:

$$\Phi(h) = \frac{\sum_{i;y_i=+1}[\![h(x_i) > 0]\!]}{\sum_i [\![y_i = +1]\!]}, \qquad \Pi(h) = \frac{\sum_{i;y_i=+1}[\![h(x_i) > 0]\!]}{\sum_i [\![h(x_i) > 0]\!]},$$

$$F_1(h) = \frac{2 \times \Phi(h) \times \Pi(h)}{\Phi(h) + \Pi(h)}.$$

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

4.3 Experimental results

We first evaluated the impact of the co-regularization training on the monolingual classification performance. As a baseline, we trained logistic regression classifiers on the monolingual data only (source language documents on one hand, English translation on the other hand), i.e. each view independently. This actually corresponds to the initialization stage in Algorithm 1, and is indicated as `logistic` in the following. We also trained the linear classifiers using the two co-classification algorithms described in the previous section, i.e. Algorithm 1 using either the gradient-based or the boosting-based approach for alternately learning each classifier. We refer to these two approaches as `cc-Logistic` (for co-classification logistic) and `cc-Boost` (for co-classification boosting), respectively. For each language, we also compared the result to a Support Vector Machine (SVM) trained on each view independently. In our experiments, we used the SVM$_{light}$ package (Joachims 1999). We used a linear kernel, and C was fixed to the default value of $C^{-1} = \frac{1}{m}\sum_{i=1}^{m} \|x_i\|^2$.

In a second stage, we compared our co-classification results to logistic and SVM models trained on the concatenated feature space obtained by joining the original and translated documents. This allows us to compare our results to an approach that uses information from both views. Note however that we are ultimately interested in learning classifiers that work on monolingual data, i.e. we do not want to require that an incoming document is first translated before it can be classified. The classifiers trained on the concatenated feature space are therefore useful for comparison, but they do not correspond to a viable alternative in our preferred use case.

All results presented below are averaged over 10 training/test splits of the initial collection.

Table 2 shows how the co-classification approach improves over the monolingual alternatives. It shows that `cc-Logistic` and `cc-Boost` always improve over the baseline `logistic`, and the difference is almost always statistically significant. In Table 2, a $^\downarrow$ symbol indicates that a result is significantly worse than the best, according to a Wilcoxon rank sum test used at a p-value threshold of 0.01 (Lehmann 1975). The co-classification also usually improves over the single-view SVM. The SVM gets the best classification performance for 4 combinations of language and class, but the difference is never significant. On the other hand, both `cc-Logistic` and `cc-Boost` get several top classification results (21 and 23, respectively), and the improvement over the SVM is usually significant. These results show that the additional translated view is able to provide additional information, and possibly some disambiguation, which our co-classification is able to leverage. This therefore supports the conclusion that the co-classification approach we propose is able to simultaneously exploit the relevant information contained in both collections.

Another observation that can be made from these results is that both co-classification algorithms behave similarly on all classes and languages. The difference in F-score is usually

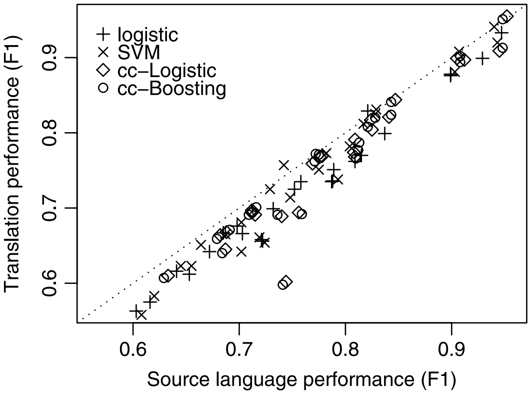
Table 2 F-measures of different learning algorithms on different classes and for all languages. The best result is in bold, and a ↓ indicates a result that is statistically significantly worse than the best, according to a Wilcoxon rank sum test with $p < .01$

		C15	CCAT	E21	ECAT	GCAT	M11
French	logistic	0.837	0.685↓	0.672↓	0.703↓	0.815↓	0.947
	SVM$_{light}$	0.828↓	0.687↓	0.664↓	0.702↓	0.817↓	0.940↓
	cc-Logistic	0.841	**0.712**	0.688	**0.715**	0.824	**0.952**
	cc-Boost	**0.843**	0.709	**0.691**	0.712	**0.828**	0.948
English$_f$	logistic	0.799↓	0.667↓	0.642↓	0.666↓	0.770↓	0.933↓
	SVM$_{light}$	**0.825**	0.665↓	0.651↓	0.642↓	0.812	0.941↓
	cc-Logistic	0.821	**0.694**	0.668	0.691	0.817	**0.955**
	cc-Boost	0.824	0.691	**0.671**	**0.694**	**0.820**	0.951
German	logistic	0.788↓	0.641↓	0.752↓	0.653↓	0.758↓	0.899
	SVM$_{light}$	0.775↓	0.645↓	0.748↓	0.655↓	0.742↓	0.903
	cc-Logistic	0.808	**0.687**	**0.776**	**0.682**	**0.778**	**0.912**
	cc-Boost	**0.812**	0.684	0.772	0.679	0.775	0.908
English$_g$	logistic	0.736↓	0.616↓	0.725↓	0.612↓	0.735↓	0.876↓
	SVM$_{light}$	0.751↓	0.623↓	0.714↓	0.623↓	0.757↓	0.881↓
	cc-Logistic	0.774	**0.645**	0.768	**0.664**	0.769	**0.897**
	cc-Boost	**0.777**	0.640	**0.772**	0.659	**0.771**	0.894
Italian	logistic	0.721↓	0.722↓	0.789↓	0.787↓	0.616↓	0.929↓
	SVM$_{light}$	0.719↓	0.724↓	0.793↓	0.782↓	0.620↓	0.943
	cc-Logistic	**0.740**	0.756	0.809	0.809	**0.633**	0.945
	cc-Boost	0.736	**0.759**	**0.813**	**0.810**	0.629	**0.948**
English$_i$	logistic	0.658↓	0.656↓	0.751↓	0.735↓	0.575↓	0.899↓
	SVM$_{light}$	0.661↓	0.654↓	0.738↓	**0.773**	0.583↓	**0.920**
	cc-Logistic	0.689	**0.694**	**0.791**	0.768	**0.610**	0.909
	cc-Boost	**0.691**	0.692	0.787	0.766	0.607	0.913
Spanish	logistic	0.698↓	0.809↓	0.603↓	0.732↓	0.821↓	0.899
	SVM$_{light}$	0.702↓	0.804↓	0.608↓	0.729↓	0.829↓	0.907
	cc-Logistic	0.712	**0.825**	**0.744**	0.769	**0.847**	0.905
	cc-Boost	**0.716**	0.821	0.741	**0.771**	0.843	**0.908**
English$_s$	logistic	0.676↓	0.762↓	0.563↓	0.699↓	0.829↓	0.878↓
	SVM$_{light}$	0.681↓	0.782↓	0.558↓	0.725↓	0.831↓	**0.908**
	cc-Logistic	0.697	0.804	**0.602**	0.759	**0.844**	0.899
	cc-Boost	**0.701**	**0.808**	0.598	**0.762**	0.841	0.902

between 0.002 and 0.004. This is not surprising as both the gradient approach and boosting are solving the same optimization problem. Their average performances are almost identical.

We also observe that the performance on the source language data (French, German, Italian and Spanish) is overall slightly higher than on the translated (English) documents, as illustrated in Fig. 1. We attribute this to the imperfect translation provided by the Statistical Machine Translation model. Note however that the difference is between 2.5 and 3.5

Fig. 1 Performance on the translation vs. source language, for all combinations of class and language, and all models. As expected, the performance is generally slightly lower on the translation



F-score points on average, which suggest that the translation, although imperfect,[3] is clearly sufficient for document categorization purposes.

Along the same lines, let us note that the *gain* in performance provided by the co-classification approach is larger on the English data. This again makes sense: given that the translated data is slightly degraded, it is not surprising that we gain more by considering source data as an additional view to the translations than when we add translated data as additional view to the source documents.

Finally, note that on class `M11`, where the performance of the baseline classifier is already high for all languages, using both views in the learning stage did not alter the performance very much. This again makes sense: as classifier performance gets higher, it gets more difficult to improve upon it.

Although the `cc-Logistic` and `cc-Boost` outperform both `logistic` and SVM in these results, the comparison is not entirely fair as the co-classification approach has access to both views, and therefore more information than the monolingual baselines. In order to address that issue, we also trained logistic and SVM classifiers on the concatenated feature space containing both views. For each class and source language, we index both the source document and the translation, and train the classifier on that. Note that this corresponds to a different use case than the co-classification approach, where the goal is to obtain a classifier that operates on a monolingual document, i.e. the bilingual data is used only during training.

The results are reported in Table 3. We see that the performance on the concatenated views increases slightly over the monolingual baseline, but stays below the co-classification approach, except for three contexts (class `C15` for `logistic` on `Spanish` and class `M11` for SVM$_{light}$ on `German` and `Spanish`), where the increase is not statistically significant. In fact, the performance is typically between what we observed on source language documents and what we obtain on translations. We analyse that as a sign that simply concatenating the features adds a lot of redundant and noisy information. Although it may improve over the monolingual classifier obtained from the lower quality view (English translations), it usually degrades the performance obtained on the higher quality view using co-classification.

Finally, we analyse the influence of the discount factor λ on the performance. Figure 2 shows how the performance varies depending on the value of λ for 4 classes (for clarity

[3] On the actual translation task, the difference between the Machine Translation output and human-quality text, as measured for example by the Translation Edit Rate, is typically around 30–40% on our reference data. It's impossible to assess on the Reuters data for lack of reference data.

Table 3 F-measures for the logistic regression and SVM for all classes, obtained on the concatenated feature space using both views

		C15	CCAT	E21	ECAT	GCAT	M11
French+English$_f$	logistic	0.817	0.673	0.672	0.672	0.790	0.938
	SVM$_{light}$	0.829	0.691	0.668	0.705	0.820	0.942
German+English$_g$	logistic	0.792	0.653	0.745	0.646	0.750	0.901
	SVM$_{light}$	0.780	0.646	0.749	0.658	0.761	**0.919**
Italian+English$_i$	logistic	0.731	0.698	0.786	0.756	0.621	0.926
	SVM$_{light}$	0.672	0.665	0.751	0.790	0.588	0.925
Spanish+English$_s$	logistic	**0.720**	0.811	0.620	0.721	0.846	0.880
	SVM$_{light}$	0.705	0.805	0.609	0.737	0.840	**0.910**

we omit CCAT and M11 from the graphs). The graphs suggest that the performance is only mildly influenced by the precise setting of the discount factor. However, it also shows that the optimal value of λ varies depending on the condition. On French, for example (top left graph in Fig. 2), lower values are best for classes C15 and ECAT, while higher values are preferable for classes E21 and GCAT.

5 Discussion

We discuss some of the interesting features of our co-classification approach, how to relax the parallel corpus assumption, and present a number of natural extensions of this work.

5.1 The virtues of co-classification

Our co-classification framework relies on a co-regularization, multiview learning approach which may be applied to various document classifiers. Our experimental results suggest that this is an effective way to train monolingual classifiers while leveraging the availability of multilingual data with the same category structure. Our results also suggest that Machine Translation may be an effective way to provide useful additional views on which the co-classification framework may be applied.

One key feature as opposed to multiple kernel learning is that after having learned from multiple views, we obtain one classifier per view, and we can therefore classify examples for which only one view is available, without having to generate additional views (using MT for example).

Another interesting feature is that we can use any monolingual classifier as long as it can be trained with a regularized cost such as (1). This allows co-classification to be computational efficient when the base classifiers are trained by gradient descent or boosting, as presented here.

5.2 Relaxing the parallel corpus assumption

In our work and experiments, we have focused on the use of a parallel corpus of documents and their translations. However, our framework extends easily to the situation where a possibly much larger *comparable corpus* is available.

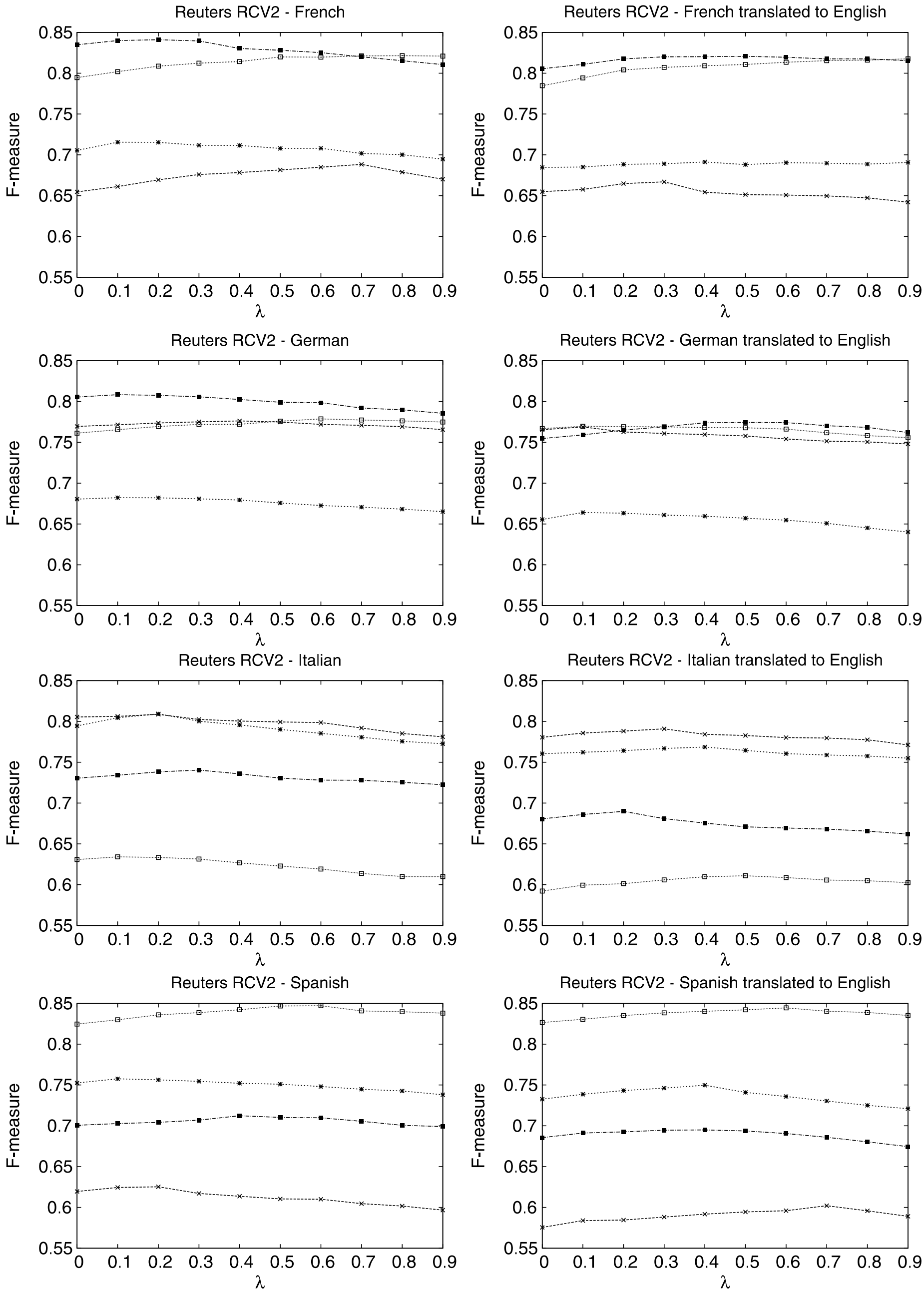


Fig. 2 F-measures of cc-Logistic with respect to the discount factor λ for French, German, Italian and Spanish (*Left*) and their translations to English (*Right*) on C15 (■), ECAT (∗), E21 (×) and GCAT (□)

A comparable corpus contains documents in both language that, roughly speaking, "talk about the same thing". It is usually argued that comparable corpora are easier to obtain, and in larger quantities, than parallel corpora. Not only do documents need not be translation of each other, the number of source and translated documents may be quite different. Theoretical as well as empirical results (Amini et al. 2009) suggest that using such comparable corpora as additional resource potentially improves the classification accuracy.

In the context of our co-classification framework, let us assume that we have an additional comparable corpus containing m_1 and m_2 documents, respectively. Let us consider each monolingual side of the corpus,[4] $T_1 = \{(x_i^{(1)}, y_i^{(1)}), i = m+1, \ldots, m+m_1\}$ and $T_2 = \{(x_i^{(2)}, y_i^{(2)}), i = m+1, \ldots, m+m_2\}$. We can take this into account during training by adding these documents to the monolingual cost. For binary classification:

$$\mathcal{C}(h_1, S_1, T_1) = \underbrace{\sum_{i=1}^{m} [\![y_i h_1(x_i^{(1)}) \le 0]\!]}_{\text{parallel corpus cost}} + \underbrace{\sum_{i=m+1}^{m+m_1} [\![y_i^{(1)} h_1(x_i^{(1)}) \le 0]\!]}_{\text{comparable corpus cost}} \qquad (10)$$

and similarly for the monolingual cost on the English side.

The divergence between the classifiers is unchanged from Sect. 3 in that case, and is still evaluated on the parallel corpus alone. The modification to Algorithm 1 is straightforward. In addition, note that we actually do not use the labels in the divergence term. The parallel corpus may therefore be entirely unlabeled. The monolingual costs may then use the labeled, comparable data, while the divergence use unlabeled parallel data from the same domain.

5.3 Extensions of co-classification

Let us describe two straightforward extensions of our co-classification framework: the multiclass, multilabel setting, and the use of non-symmetric losses.

Although we have described our algorithms on binary classification, it is naturally possible to extend the framework to multiclass (both single- and multilabel). As the multiclass, multilabel situation may be seen as multiple binary classifications, described above, we will describe how the model can handle multiclass, single label classification. In that situation, $\mathcal{Y} = \{1, \ldots, K\}$. The monolingual cost $\mathcal{C}(\mathbf{h}, S)$ is then changed to reflect that. Assuming that the classifier $\mathbf{h}$ outputs a vector of h_k, $k = 1 \ldots K$, a multiclass extension of the misclassification cost used in (5) is:

$$\mathcal{C}(\mathbf{h}, S) = \sum_{i=1}^{m} \left[\!\!\left[\operatorname*{argmax}_k h_k(x_i) \neq y_i \right]\!\!\right]. \qquad (11)$$

The general shape of the global objective (see (4)) does not change, but the divergence between the classifier outputs is updated to handle multiple classes:

$$d(\mathbf{h}^{(1)}(x_i), \mathbf{h}^{(2)}(x_i)) = \sum_k \left(\sigma_k(\mathbf{h}^{(1)}(x_i)) - \sigma_k(\mathbf{h}^{(2)}(x_i))\right) \log \left(\frac{\sigma_k(\mathbf{h}^{(1)}(x_i))}{\sigma_k(\mathbf{h}^{(2)}(x_i))} \right) \qquad (12)$$

[4] In our notation, documents $x_i^{(1)}$ and $x_i^{(2)}$ are translations of each other, and have an identical label $y_i^{(1)} = y_i^{(2)} = y_i$, for $i = 1 \ldots m$, whereas for $i > m$, the documents are different and may have different labels $y_i^{(1)}$ and $y_i^{(2)}$.

where $\sigma(\mathbf{h}(x))$ is a "softmax" which transforms the numeric scores $\mathbf{h}(x)$ into output probabilities, e.g. $\sigma_k(\mathbf{h}(x)) = \exp(h_k(x))/\sum_j \exp(h_j(x))$.

The previous extension leverages the general form of the global cost (4), which allows both classifiers to be trained on different sets of documents. Notice that this can be pushed further by actually using different costs for each view, or even train different classifiers on each view, as long as both can produce probabilities as output, so that we can compute the divergence. On languages that are linguistically very different, this may actually be relevant. E.g. if one view is a language with a natural tokenization, a bag-of-word approach usually performs well. If the other language has no natural segmentation (e.g. Chinese), a classifier and cost working at the level of character sequences may be more appropriate.

6 Conclusion

In this paper we presented a strategy for learning to classify documents from multilingual corpora. Our approach takes into account the disagreement of classifiers on the parallel part of a corpus, where for each document there exists a translated version in the other language. We derived training algorithms for logistic regression and boosting, and show that the resulting categorizers outperform models trained independently on each language, as well as classifiers trained on the concatenation of both languages. Experiments were performed on four corpora extracted from Reuters RCV2, where each document was translated using a Statistical Machine Translation model. We are working towards making available our preprocessed bilingual corpora as a usable resource for the community. Our results suggest that multi-view learning is a promising framework for learning text categorizers from multilingual corpora. They also show that Machine Translation can help improve text categorization performance. There exist a numerous interesting directions to be explored under the framework of learning from multiple languages, making this paradigm ideal for further research.

Acknowledgements The authors would like to thank Nicolas Usunier for his helpful comments and also thank Reuters for making the RCV1/RCV2 data available and granting permission to distribute processed versions of it.

References

Adeva, J. J. G., Calvo, R. A., & de Ipiña, D. L. (2005). Multilingual approaches to text categorisation. *UPGRADE: The European Journal for the Informatics Professional*, *VI*(3), 43–51.

Amini, M.-R., Usunier, N., & Goutte, C. (2009). Learning from multiple partially observed views—an application to multilingual text categorization. *Advances in Neural Information Processing*, *23*.

Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. In *Proc. 21st international conference on machine learning*.

Bel, N., Koster, C. H., & Villegas, M. (2003). Cross-lingual text categorization. In *Proceedings ECDL 2003* (pp. 126–139).

Bertsekas, D. (1999). *Nonlinear programming* (2nd ed.). Belmont: Athena Scientific.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on computational learning theory* (pp. 92–100).

Brefeld, U., Gärtner, T., Scheffer, T., & Wrobel, S. (2006). Efficient co-regularised least squares regression. In *Proc. 23rd international conference on machine learning* (pp. 137–144).

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. In *Proceedings of the third annual symposium on document analysis and information retrieval*, Las Vegas, NV (pp. 161–175).

Collins, M., Schapire, R. E., & Singer, Y. (2000). Logistic regression, AdaBoost and Bregman distances. In *Proc. computational learning theory* (pp. 158–169).

Csiszár, I. (1995). Maxent, mathematics and information theory. In *Proceedings of the fifteenth international workshop on maximum entropy and Bayesian methods* (pp. 35–50).

Diethe, T., Hardoon, D. R., & Shawe-Taylor, J. (2008). Multiview Fisher discriminant analysis. In Hardoon, D. R., Leen, G., Kaski, S., & Shawe-Taylor, J. (Eds.), *NIPS workshop on learning from multiple sources*.

Farquhar, J. D. R., Hardoon, D. R., Meng, H., Shawe-Taylor, J., & Szedmak, S. (2005). Two view learning: SVM-2K, theory and practice. *Advances in Neural Information Processing*, *18*.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *International conference on machine learning* (pp. 200–209).

Kakade, S. M., & Foster, D. P. (2007). Multi-view regression via canonical correlation analysis. In *Proc. computational learning theory (COLT)*.

Lafferty, J. D., Della Pietrea, S., & Della Pietra, V. (1999). Statistical learning algorithms based on Bregman distances. In *Proceedings of the Canadian workshop on information theory*.

Lehmann, E. L. (1975). *Nonparametric statistical methods based on ranks*. New York: McGraw-Hill.

Oard, D. W., & Diekema, A. R. (1998). Cross-language information retrieval. *Annual Review of Information Science and Technology*, *33*.

Reuters (2000). *Reuters Corpus*, *Vol. 2*: *Multilingual, 1996-08-20 to 1997-08-19*.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., & Gatford, M. (1994). Okapi at TREC-3. In *Proc. 3rd text retrieval conference (TREC)*.

Rosenberg, D. S., & Bartlett, P. L. (2007). The Rademacher complexity of co-regularized kernel classes. In M. Meila & X. Shen (Eds.), *Proceedings of the eleventh international conference on artificial intelligence and statistics*.

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). A co-regularization approach to semi-supervised learning with multiple views. In *Proc. of the workshop on learning with multiple views at the 22nd ICML*.

Topsoe, F. (1979). Information theoretical optimization techniques. *Kybernetika*, *15*, 7–17.

Ueffing, N., Simard, M., Larkin, S., & Johnson, J. H. (2007). NRC's PORTAGE system for WMT 2007. In *ACL-2007 second workshop on SMT* (pp. 185–188).

van Rijsbergen, C. (1979). *Information retrieval*. London: Butterworths.