Using KCCA for Japanese-English Cross-language Information Retrieval and Document Classification

Yaoyong Li* and John Shawe-Taylor†

Abstract

Kernel Canonical Correlation Analysis (KCCA) is a method of correlating linear relationship between two variables in a kernel defined feature space. A machine learning algorithm based on KCCA is studied for cross-language information retrieval. We apply the algorithm in Japanese-English cross-language information retrieval. The results are quite encouraging and are significantly better than those obtained by other state of the art methods. Computational complexity is an important issue when applying KCCA to large dataset as in information retrieval. We experimentally evaluate several methods to alleviate the problem of applying KCCA to large datasets. We also investigate cross-language document classification using KCCA as well as other methods. Our results show that it is feasible to use a classifier learned in one language to classify the documents in other languages.

1 Introduction

Recently there is a growing need for advanced information retrieval techniques to help people exploit a vast amount of information available through the Internets. Cross-language information retrieval enables us to retrieve information from other languages using a query written in the language we are familiar with. A cross-language information retrieval system can be built up via two approaches. One is to use machine translation to translate the query so that the problem is transformed into a monolingual information retrieval task where a variety of techniques can be employed (e.g. [11]). Another way is to first automatically induce a semantic correspondence between two languages by some automatic methods such as machine learning and then use it to project the inquiry into another language to accomplish cross-language information retrieval (e.g. [10], [12]). In [10] cross-language latent semantic indexing (CL-LSI) was proposed as a fully automatic method for cross-language information retrieval, which produced results comparable to (and sometimes better than) those obtained with machine translation systems. In [12] kernel canonical correlation analysis (KCCA) was used for cross-language information retrieval, which achieved significantly better performance than CL-LSI on an English-French corpus. The machine learning based method is interesting because its performance is comparable to the machine translation based method but its implementation is easier.

Canonical correlation analysis (CCA), proposed by H. Hotelling in [5], aims to find basis vectors for two sets of variables such that the correlation between the projections of the variables onto these basis vectors are mutually maximised. CCA can be seen as using complex labels as a way of guiding feature selection toward the underlying semantics. CCA makes use of two views of the same semantic object to extract the representation of the semantics. Here semantics refers to the content of an object (e.g. document) and different views are the different representations of the object (i.e. the document's text in different

*Department of Computer Science, The University of Sheffield
†ISIS Group, School of Electronics and Computer Science, University of Southampton

languages). In an attempt to increase the flexibility of the feature selection, kernelisation of CCA (KCCA) has been applied to map the data to a higher-dimensional feature space.

KCCA is particularly suitable for applications where the semantics of the object with two or more views are crucial. Two such problems are cross-language information retrieval and multimedia content-based retrieval. In the former the semantics refers to the content of a document and the texts of document in different languages represent different views. In the latter the semantics is the content of the multimedia object and the different media representations form different views. Actually KCCA has achieved state of the art results for the two problems. In [4] KCCA with a Gaussian kernel was applied to a collection of images with attached text to extract the semantic correspondence between image and text, which was then used to perform content-based image retrieval from a text query. In [12] KCCA with a linear kernel was used successfully to infer a semantic representation from an English-French bilingual corpus, as shown by the experimental results for cross-language information retrieval and text categorisation using the derived semantics.

In this paper we investigate KCCA for Japanese-English cross-language information retrieval and other issues. We also study another interesting topic using KCCA — cross-language document classification where a classifier learned in one language is used to classify documents in a second language. The paper has two novelties. One is that we applied the KCCA to two very different languages Japanese and English for cross-language information retrieval. Previously the KCCA was successfully used for English and French cross-language information retrieval. Another one is cross-language document classification. Previously we had considered using the semantic space induced by KCCA from a bilingual corpus to do classification on another monolingual corpus (see [12]), but not using the classifier trained in one language for documents in the other, as what we study in this paper.

The paper is organised as the following. In Section 2 we formulate kernel canonical correlation analysis in the context of cross-language text applications. In the following sections we present the experimental results using KCCA for Japanese-English cross-language information retrieval and document classification. As KCCA has been shown to be successful for cross-language English-French information retrieval, in Section 3 we investigate whether it could perform as well for two very different languages like English and Japanese. Section 4 addresses an important problem with KCCA — how to deal with large datasets. More training data for KCCA may result in a better semantic representation. On the other hand, as KCCA leads to a generalised eigenvalue problem, the computation time for KCCA may be very long if the training set is very large. We experimentally evaluate several methods which were proposed to help KCCA to alleviate the problem caused by large datasets. Finally Section 5 presents the methods and experimental results for cross-language document classification.

2 KCCA for cross-language text applications

KCCA induces a set of basis vectors in feature space from a collection of bilingual documents. Those vectors can be regarded as a semantic representation of the bilingual corpus. Here the semantic representation means that a basis vector of KCCA corresponds to one theme or several mixed themes of the corpus, which are represented by the typical terms about the themes in two languages. Figure 1 shows two examples of the basis vector obtained from a Japanese-English patent collection (see Section 3 for the detailed explanations of this collection). Each of such basis vector has more than 150 thousands components, of which we only list the first 10 largest components (the values and the corresponding terms) respectively for Japanese and English. It looks that the vector in the left of Figure 1 represents three mixed themes: a natural farming method, stepping motor and a new device for photo development, while the vector in the right is mainly for one theme, stepping motor. Since the KCCA extracts distinct themes from a text collection and repre-

Figure 1: The semantic representation of KCCA basis vector: ten terms respectively in Japanese and English which correspond to the largest components of vector. Note that the English terms are the stemmed words.

sents the themes respectively in two languages, we could represent a document in one or another language as some combination of the themes and use this kind of semantic representation for cross-language text applications such as information retrieval and document classification. For example, we can first obtain the semantic representations of a query in one language and some documents in another language by projecting them onto the KCCA basis vectors and then retrieve relevant documents for the query by comparing the semantic representations (see Section 3 for more details). In the following we will show how KCCA infers a set of basis vectors from a bilingual corpus as the semantic representation.

Suppose we are given N pairs of documents in two languages, i.e. every document c_i $(i = 1, \ldots, N)$ in one language is a translation of document d_i in another language. After some preprocessing, we obtain a feature vector $x_i \in \mathcal{X}$ for every document c_i and a feature vector $y_i \in \mathcal{Y}$ for document d_i, where $\mathcal{X}$ and $\mathcal{Y}$ are the feature spaces of the two languages, respectively. By using canonical correlation analysis (CCA), we can find some directions $f_x \in \mathcal{X}$ and $f_y \in \mathcal{Y}$ in the two spaces such that the projections $\{(f_x, x_i)\}_{i=1}^N$ and $\{(f_y, y_i)\}_{i=1}^N$ of the feature vectors of documents from the two languages would be maximally correlated. Then we can find another maximally correlated directions in the two complementary subspaces of the one-dimensional subspaces f_x and f_y in the feature spaces $\mathcal{X}$ and $\mathcal{Y}$, respectively, and so on. If the features consists of content terms (i.e. the stemmed words excluding stop words) from the documents as in the experiments described in [12] and in this paper (which corresponds to linear kernel, see the discussions below), then the directions f_x and f_y may represent the terms about the most popular topics in the collection in two languages, respectively, as these terms are most common in the document pairs $(c, d) \in \mathcal{X} \times \mathcal{Y}$. Therefore, the pair of directions f_x and f_y may represents some of the most distinct themes in the document collection, which could be useful for cross-language applications.

Formally, CCA finds a canonical correlation ρ in the space $\mathcal{X} \times \mathcal{Y}$ which is defined as

$$\begin{aligned} \rho &= \max_{(f_x, f_y) \in \mathcal{X} \times \mathcal{Y}} \text{corr}((f_x, x_i), (f_y, y_i)) \\ &= \max_{(f_x, f_y) \in \mathcal{X} \times \mathcal{Y}} \frac{\sum_{i=1}^N (f_x, x_i)(f_y, y_i)}{\sqrt{\sum_i (f_x, x_i)^2 \sum_j (f_y, y_j)^2}} \end{aligned} \quad (1)$$

We search for f_x and f_y in the space spanned by the corresponding feature vectors, i.e.

$$f_x = \sum_l \alpha_l x_l, \quad f_y = \sum_m \beta_m y_m \quad (2)$$

This rewrites the numerator of (1) as

$$\sum_i (f_x, x_i)(f_y, y_i) = \sum_i \sum_{lm} \alpha_l \beta_m (x_l, x_i)(y_m, y_i) = \alpha^T G_x G_y \beta \tag{3}$$

where α is the vector with components α_l $(l = 1, ..., N)$ and β the vector with components β_m $(m = 1, ..., N)$ and G_x is the Gram matrix of $\{x_i\}_{i=1}^N$ and G_y the Gram matrix of $\{y_j\}_{j=1}^N$. The problem (1) can then be reformulated as

$$\rho = \max_{\alpha,\beta} \frac{\alpha^T G_x G_y \beta}{\sqrt{\alpha^T G_x^2 \alpha \cdot \beta^T G_y^2 \beta}} \tag{4}$$

In order to force non-trivial learning on the correlation, we introduce a regularisation parameter to penalise the norms of the associated weights (e.g. see [1]). By doing so, the problem (4) becomes

$$\rho = \max_{\alpha,\beta} \frac{\alpha^T G_x G_y \beta}{\sqrt{(\alpha^T G_x^2 \alpha + \kappa \alpha^T \alpha) \cdot (\beta^T G_y^2 \beta + \kappa \beta^T \beta)}} \tag{5}$$

Note that the new regularised equation is not affected by re-scaling of α or β, hence the optimisation problem is subject to the two constraints

$$\alpha^T G_x^2 \alpha + \kappa \alpha^T \alpha = 1 \tag{6}$$
$$\beta^T G_y^2 \beta + \kappa \beta^T \beta = 1 \tag{7}$$

The corresponding Lagrangian is as

$$L(\alpha, \beta, \lambda_\alpha, \lambda_\beta) = \alpha^T G_x G_y \beta - \frac{\lambda_\alpha}{2}(\alpha^T G_x^2 \alpha + \kappa \alpha^T \alpha - 1) - \frac{\lambda_\beta}{2}(\beta^T G_y^2 \beta + \kappa \beta^T \beta - 1)$$

Taking derivatives of the Lagrangian with respect to α and β and setting them to be zero, respectively, we have the equations

$$G_x G_y \beta - \lambda_\alpha (G_x^2 + \kappa I)\alpha = 0 \tag{8}$$
$$G_y G_x \alpha - \lambda_\beta (G_y^2 + \kappa I)\beta = 0 \tag{9}$$

The solution (α, β) of the equations (8) and (9) is the solution of the optimisation problem (5) with the constraints (6) and (7). Taking α^T times equation (8) we have

$$\alpha^T G_x G_y \beta - \lambda_\alpha \alpha^T (G_x^2 + \kappa I)\alpha = 0$$

which together with (6) implies that

$$\lambda_\alpha = \alpha^T G_x G_y \beta$$

Similarly taking β^T times equation (9) together with the constraint (7) we have

$$\lambda_\beta = \beta^T G_y G_x \alpha$$

The above two equations imply that

$$\lambda_\alpha = \lambda_\beta = \alpha^T G_x G_y \beta \tag{10}$$

Letting $\lambda = \lambda_\alpha = \lambda_\beta$, we can rewrite the equations (8) and (9) as a generalised eigenvalue problem

$$B\xi = \lambda D\xi \tag{11}$$

where λ is the canonical correlation ρ between the projections (f_x, x_i) and (f_y, y_i) $(i = 1, ..., N)$, and

$$B = \begin{pmatrix} 0 & G_x G_y \\ G_y G_x & 0 \end{pmatrix}, \quad D = \begin{pmatrix} G_x^2 + \kappa I & 0 \\ 0 & G_y^2 + \kappa I \end{pmatrix}, \quad \xi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{12}$$

Finally, note that the vectors α and β of eigenvector ξ would be used to deduce the basis vector f_x and f_y in two languages, respectively, and the regularisation might cause the two basis vectors being differently scaled. Hence, using the equations (2) we should rescale the vectors α and β such that

$$\alpha^T G_x^2 \alpha = 1 = \beta^T G_y^2 \beta \tag{13}$$

Considering the constraints (6) and (7), the rescaling can be achieved by

$$\alpha = \alpha / \sqrt{1 - \kappa \alpha^T \alpha}, \quad \beta = \beta / \sqrt{1 - \kappa \beta^T \beta} \tag{14}$$

Therefore, the optimisation problem of the CCA has been transformed into a generalised eigenvalue problem (11), where the eigenvectors with the largest eigenvalues represent the maximally correlated directions in feature space. In other words, the eigenvector $\xi_1 = (\alpha_1^T, \beta_1^T)^T$ with the largest eigenvalue λ_1 forms the maximally correlated directions f_x and f_y in the feature spaces $\mathcal{X}$ and $\mathcal{Y}$ by using the equations (2). The eigenvector $\xi_2 = (\alpha_2^T, \beta_2^T)^T$ with the second largest eigenvalue λ_2 forms the maximally correlated directions in the complementary subspaces of the subspaces f_x and f_y in the feature spaces $\mathcal{X}$ and $\mathcal{Y}$, respectively, and so on.

We can see that, either in the optimisation problem (5), (6) and (7) or in the eigenproblem (11), the training points $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ are involved only through the Gram matrix G_x and G_y. Therefore, the so-called "kernel-trick" can be used to introduce extra flexibility into CCA. Kernelisation of CCA means that the training points $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ are mapped to another (some high-dimensional) feature space by a kernel function (see e.g. [2]) and the canonical correlation is then computed in the new feature space. This can be done easily by replacing the Gram matrices with the corresponding kernel matrices in the optimisation formulation (5), (6) and (7) and in the eigenproblem (11). A Gaussian kernel was employed in [4] for text-image content based retrieval. The experiments in [12] showed that the linear kernel was quite good for cross-language applications of KCCA (As a matter of fact, [6] also showed that the linear kernel performed similarly with other types of kernel for the monolingual document categorisation). Moreover, linear kernel is simpler and leads to faster learning algorithm than other kernels. Hence, the linear kernel was used in our experiments as well. Using the terms (i.e. stemmed words) as features together with linear kernel means that the feature space is basically vocabularies. Precisely every dimension of the feature space corresponds to a term (i.e. a stemmed word). Also, we used the same value of regularisation parameter as in [12], i.e. $\kappa = 1.5$ (also see [12] for a detailed discussion of the regularisation parameter).

3 Using KCCA for cross-language information retrieval

Cross-language information retrieval with KCCA. In the previous section we have shown that KCCA leads to a generalised eigenvalue problem. The eigenvectors with the largest eigenvalues correspond to the maximally correlating directions in the feature spaces, which constitute some kind of semantic basis vectors. These basis vectors represent semantic correspondence between the training documents of the two languages, i.e. every vector represents a theme or several mixed themes of training documents in two languages and a theme is represented by a distribution among the vectors (also see Figure 1). These basis vectors provides a framework for performing cross-language information retrieval where, given a

query in one language, we try to find out the relevant documents in another language. Here we adopt the procedure described in [12] for cross-language information retrieval using KCCA. We first pick a number d of eigenvectors with largest eigenvalues from the solution of (11) for two languages A and B, and compute the corresponding maximally correlated directions in the feature spaces which represents the most distinct themes of the collection in the two language. Then we represent query in language A as a combination of themes by projecting the query onto the language A part of the basis vectors, and also represent some documents in language B by the same themes by projecting them onto the language B part of the basis vectors. Finally we compare the semantic representations of the query and the documents to select the relevant documents in language B for the query in language A. Formally, to process a query q we represent q as a feature vector $\tilde{q}$ and project it onto the d canonical correlation directions in feature space

$$\tilde{q}_d = A^T Z^T \tilde{q} \tag{15}$$

where A is $N \times d$ matrix whose columns are the first or the second half (depending on which language was used in the query) of eigenvectors of (11) with the largest d eigenvalues, and each column of Z is a training vector in the same language as the query. Similarly, we represent the documents for retrieval in another language as d-dimensional vectors by projecting them onto the d-dimensional canonical correlation directions. Then the documents with the shortest distances to the query in the d-dimensional space are regarded as being relevant to the query.

The dataset used for the experiments. The dataset we used was from the NTCIR-3 patent retrieval test collection[1]. The collection includes about 1.7 million Japanese patent abstracts and their English translations, spanning five years (1995–99). Only the 336,929 documents from 1995 (referred to as the 1995 collection hereafter) was used in the experiments we did. First of all, we collected the terms and computed the idf (inverse document frequency) for every term from the 1995 collection. The English terms were collected in the usual way, i.e. down-casing the alphabetic characters, removing the stop words, replacing every non-alphabetic character with a blank space, stemming words using the Porter stemmer, and finally removing the terms which appear less than 3 times in the corpus. We preprocessed the Japanese documents using a Japanese morphological analysis software Chasen[2] version 2.3.3, as was done in [11]. From the documents processed by the Chasen, we picked up as our terms those words whose part of speech tags were either noun (but not dependent noun, proper noun or number noun), or independent verb, or independent adjective, or unknown. We also removed the Japanese terms appearing less than three times in the documents of the 1995 collection. By doing so, 61583 English terms and 90055 Japanese terms were obtained, respectively. Then we computed the $tf * idf$ feature vectors for the Japanese patent abstracts and the corresponding English translations in the usual way (e.g. see [6]) and finally normalised the feature vectors.

Mate retrieval. We first conducted experiments for mate retrieval. In mate retrieval a document in one language was treated as a query and only the mate document in another language was considered as relevant. A mate document was considered to be retrieved if it is most close to the query document in the semantic space. We applied KCCA to the first 1000 Japanese documents and the English translations of the 1995 collection. For comparison, we also implemented LSI for cross-language information retrieval (see [10]) under the same experimental settings, since the results of LSI on the collection we used was not available from other people.

The results presented in the upper part of Table 1 is for 1000 training documents as queries. The lower part of Table1 shows the results for the 2000 test documents used as queries. These results are consistent with those on the English-French documents (see

[1] See http://research.nii.ac.jp/ntcir/permission/perm-en.html

[2] See http://chasen.aist-nara.ac.jp/

[12]). That is, KCCA can achieve quite good performance using a fraction of eigenvectors (say 200) while LSI achieved the same results only when using the full 1000 eigenvectors. We can also see that the KCCA significantly outperformed LSI on the test documents.

Table 1: Mate retrieval (1000 training documents): the accuracy rates averaged over all the training documents and over other 2000 test documents, respectively. Different numbers of the eigenvectors were used and KCCA was compared with LSI. E→J means using English query to retrieve Japanese documents and J→E means Japanese query and English documents.

#Eigenvectors	**5**	**10**	**50**	**100**	**200**	**300**	**400**	**500**	**1000**
Training docs as queries									
KCCA(E-J)	0.675	0.898	0.973	0.988	0.993	0.994	0.993	0.992	0.995
KCCA(J-E)	0.661	0.876	0.973	0.979	0.987	0.988	0.988	0.985	0.998
LSI(E-J)	0.093	0.328	0.769	0.898	0.949	0.960	0.965	0.966	0.996
LSI(J-E)	0.091	0.264	0.652	0.827	0.923	0.946	0.952	0.959	0.996
Test docs as queries									
KCCA(E-J)	0.050	0.154	0.402	0.466	0.528	0.519	0.496	0.471	0.338
KCCA(J-E)	0.084	0.174	0.368	0.449	0.462	0.423	0.388	0.356	0.233
LSI(E-J)	0.037	0.095	0.296	0.376	0.431	0.431	0.417	0.393	0.247
LSI(J-E)	0.029	0.079	0.212	0.294	0.362	0.355	0.329	0.304	0.170

Pseudo query retrieval. We also carried out experiments for pseudo query retrieval. We generated a short query consisting of the five most probable words for each test document. And the relevant document is the mate of the document in another language. Table 2 shows the relative number of correctly retrieved documents in each experimental setting. Once again, we present the results for the queries from the 1000 training documents and the 2000 test documents, respectively. The retrieval accuracy of KCCA is high and is better than those using LSI when a short query was generated from training document.

The experimental results have shown that KCCA outperformed LSI consistently for cross-language information retrieval. We can also see that similar results were obtained for the English-Japanese bilingual corpus as that reported for English-French documents in [12], despite that English is much more different from Japanese than from French. Therefore, the KCCA provides a very encouraging way for cross-language information retrieval.

We can also see from the above results that, while the retrieval accuracy was quite high with training documents as queries, the retrieval accuracy became low when the documents not in training set were used as queries. This may be due to the small number of training documents. KCCA extracted a semantic correspondence between two languages from the training documents. If the training set is too small to be representative, then the semantic correspondence may not have a good coverage for documents not in training set.

More training documents. We expected that the KCCA have better generalisation performance when the training set became larger. To verify it, we added another 1000 documents into the training set and then repeated the above experiments with the enlarged training set. In the case of training documents as queries, the results for 2000 training documents were similar to those for 1000 training documents. The results for the 2000 other test documents as queries are presented in Table 3. Comparing with the corresponding results in Table 1 and Table 2, we can see from Table 3 that the generalisation performance has indeed improved when using more training documents.

It is possible that the generalisation performance of KCCA will become better if we use yet more training documents. However, we are unable to use a very large training set for KCCA because the computation time becomes very long when using for example 50,000

Table 2: Pseudo query retrieval (1000 training documents): the accuracy rates averaged over all the training documents and over 2000 test documents, respectively. Different numbers of eigenvectors were used and KCCA was compared with LSI. E→J means using English query to retrieve Japanese documents and J→E means Japanese query and English documents.

#Eigenvectors	**5**	**10**	**50**	**100**	**200**	**300**	**400**	**500**	**1000**
Training docs as queries									
KCCA(E-J)	0.042	0.132	0.484	0.711	0.906	0.966	0.965	0.959	0.949
KCCA(J-E)	0.061	0.144	0.379	0.608	0.813	0.966	0.972	0.952	0.946
LSI(E-J)	0.062	0.170	0.415	0.561	0.734	0.785	0.829	0.862	0.911
LSI(J-E)	0.048	0.128	0.244	0.317	0.433	0.495	0.528	0.539	0.548
Test docs as queries									
KCCA(E-J)	0.024	0.068	0.167	0.200	0.219	0.228	0.227	0.223	0.197
KCCA(J-E)	0.029	0.061	0.136	0.162	0.166	0.156	0.149	0.139	0.107
LSI(E-J)	0.028	0.077	0.152	0.186	0.203	0.212	0.220	0.211	0.172
LSI(J-E)	0.023	0.061	0.114	0.137	0.140	0.140	0.133	0.126	0.093

Table 3: Results of experiments with the 2000 training documents: the accuracy rates averaged over 2000 test documents. Different numbers of the eigenvectors were used. E→J means using English query to retrieve Japanese documents and J→E means Japanese query and English documents.

#Eigenvectors	**5**	**10**	**50**	**100**	**200**	**300**	**400**	**500**	**1000**
Mate retrieval									
KCCA(E-J)	0.134	0.245	0.565	0.609	0.638	0.642	0.628	0.607	0.469
KCCA(J-E)	0.160	0.287	0.525	0.573	0.591	0.569	0.537	0.499	0.351
Pseudo query retrieval									
KCCA(E-J)	0.063	0.101	0.209	0.249	0.274	0.287	0.297	0.326	0.273
KCCA(J-E)	0.054	0.089	0.169	0.194	0.207	0.217	0.213	0.202	0.166

documents for training. In the next section we will discuss several methods to help KCCA deal with a large training set. In the rest of this section, we discuss two problems raised from the above results.

One problem we can see from the results of our experiments is that the accuracy of retrieving English documents from a Japanese query (from Japanese to English) is lower than from English to Japanese in almost all cases. On the other hand, when applying KCCA to English-French corpus for cross-language information retrieval in [12], the results were very similar when using English document as query for retrieving French documents or using French document for retrieving English documents. Note that the main difference between processing English and Japanese documents was in the procedure of collecting the terms. The English (or French) terms were basically the stemmed words. However, in Japanese, unlike in English or French, there is no delimiter between words in a sentence. Hence we had to employ some procedure to segment Japanese sentence into a sequence of words and then to select Japanese terms according to the POS tags. The procedure of collecting Japanese terms may introduce more errors than that of collecting English term. Therefore, we think that the lower accuracy of using Japanese query for retrieving English documents may due to the fact that the quality of the Japanese terms we collected was not as good as that of English terms. Searching a better method of collecting Japanese terms than the one we used would be part of the future work.

Another problem is how to choose the optimal number of KCCA eigenvectors. First,

we can see from the above tables that the performance is not very sensitive to the number of KCCA eigenvectors. For example, in most cases, the number of eigenvectors which was not far (say 100) from the optimal one gave similar results. In the application we may use some empirical methods for choosing a good number of eigenvectors. Actually determining the optimal values of parameters in a learning algorithm is an important research problem in machine learning. Several empirical methods such as n-fold cross-validation have been studied and work well in some applications (see e.g. [7] or the machine learning textbooks).

4 Methods for KCCA to deal with large training sets

As shown above, KCCA's performance was improved when we used more training examples. Since KCCA is a kind of unsupervised learning algorithm, we can easily collect a large (unlabeled) training set for it. Hence we can use a large training set for KCCA to achieve better performance for cross-language information retrieval. However, it is difficult to apply KCCA directly to a large training set because of its computational complexity. A naive implementation of KCCA would scale as $O(N^3)$, a computational complexity with cubic growth in the number of data points N. So we have to use some method to help KCCA handle large training sets. To this end, we consider two strategies. One is to only use the salient examples from the training set. The partial Gram-Schmidt orthogonalisation of the training examples (or equivalently the incomplete Cholesky decomposition of the kernel matrix) is one example using this kind of strategy. Another strategy is to split the training set into small groups and compute KCCA for each group. In [12] the large training set was randomly split in order to alleviate the problem of large datasets. Here we propose a further method — cluster the training set before applying KCCA.

Incomplete Cholesky decomposition of the Gram matrix was used in [1] to reduce the computational complexity of a similar algorithm. A positive semidefinite $N \times N$ matrix K, such as the Gram matrix, can always be factored as GG^T, where G is an $N \times N$ matrix as well. This factorisation can be found via Cholesky decomposition. Incomplete Cholesky decomposition is to find a matrix $\tilde{G}$ of size $N \times M$, for small M, such that the difference $K - \tilde{G}\tilde{G}^T$ has norm less than a given precision. As shown in [1], this kind of approximation can reduce the computational complexity to be quadratic in the size of the training set (i.e. $O(N^2)$).

The partial Gram-Schmidt orthogonalisation was explored for KCCA in [4]. The Gram-Schmidt orthogonalisation algorithm was basically to determine a subset of examples with a pre-defined size, which were furthest from each other in the feature space and could be regarded as the most salient examples of training set. See [3] for more detail about the algorithm. By the partial Gram-Schmidt orthogonalisation, a semantic space was formed as a span of a subset of training examples which were selected by performing Gram-Schmidt orthogonalisation procedure on the training vectors in the feature space. The partial Gram-Schmidt orthogonalisation is equivalent to an incomplete Cholesky decomposition as the latter is the dual implementation of the former.

Another way for KCCA to deal with large training set is to split the training set into some relatively small subsets and apply KCCA to each subset independently and then integrate the solutions of the KCCAs from the subsets into a general semantic correspondence between two languages. One obvious way was to split the training set randomly. Another possible approach is to cluster the training set into small groups. We hypothesis that clustering a large training set may be a better way for KCCA to handle large dataset than splitting it randomly into small groups. Clustering a large dataset not only results in small training sets for KCCA, it can also put together documents with similar contents so that the semantic correspondence extracted by KCCA from the cluster could be a good semantic representation of the cluster.

We carried out the experiments to make a comparison among the methods described above. We put together the 101 documents, each of which is relevant to one of the first five

topics in the NTCIR-3 collection, as the training set (denoted as Set-A hereafter). In order to evaluate the scalability of the results, we also did experiments using a larger set of the 306 documents (denoted as Set-B hereafter) each of which was relevant to one of the first ten topics in the NTCIR-3 collection.

Four methods were compared in these experiments, i.e. the partial Gram-Schmidt orthogonalisation, the clustering, splitting the training set randomly, and applying KCCA directly. For the clustering method, as we know that a document in the training set is relevant to one of the five topics for Set-A (or ten topics for Set-B), we did not implement any clustering algorithm in our experiment. Instead we checked two cases. One is the perfect clustering, which meant we had five clusters for Set-A (or ten clusters for Set-B) and a cluster consisted of the documents relevant to one unique topic. Another is the clustering with 20% noise. In this case we had five (or ten) clusters again but on average only 80% of the documents in a cluster are relevant to a topic. Correspondingly, for the random split, we randomly divided the document set into five (or ten) partitions of the same size (the last partition may be larger than others so that the partitions contained all the documents).

For the partial Gram-Schmidt method, we had to specify an important parameter, i.e. the number of training examples to be selected (the dimension of the subset used). It can be determined automatically through a predefined precision parameter as shown in [4]. In our experiments, however, we tried several values of the parameter to see the effect of the parameter on the overall performance (see Table 4) and then used the optimal one.

The KCCA eigenvectors obtained were then used for a cross-language information retrieval task where an English document in the training set was regarded as a query and the Japanese documents which were on the same topic as the query were considered as relevant. The averaged precision was computed to measure the retrieval performance for one query. The means of the averaged precisions over all the English documents were used to evaluate the different methods.

In Table 4 we present the results for the partial Gram-Schmidt method with different dimensions of subsets used for two document sets, respectively. For the Set-A, we obtained the best result with the dimension 60 of the subset. For the Set-B, the dimension 120 of the subset was the best. We can see that the performances just decreased slightly for other dimensions not very different (say 20) from the optimal one for both sets, which means that the performance of Gram-Schmidt method for KCCA is stable with respect to the dimensional parameter. On the other hand, the optimal numbers are different for the two sets, which is not surprising because the two set has different numbers of documents and more importantly different numbers of topics. The optimal values of dimensional parameter (namely 60 and 120 for the Set-A and Set-B, respectively) would be used in the subsequent experiments.

Table 4: The results for Gram-Schmidt method with different dimensions of subset for two document sets, the Set-A and Set-B, respectively. The means of averaged precisions were used to measure the overall performances for cross-language document retrieval. Note that we cannot obtain a subset of dimension 120 or 140 for the set-A because the total number of data in Set-A is 101.

Dimension	5	10	20	40	60	80	100	120	140
Set-A	0.699	0.700	0.845	0.856	0.881	0.851	0.833	*	*
Set-B	0.445	0.668	0.689	0.737	0.846	0.866	0.876	0.894	0.873

Table 5 presents the results[3] for the methods we evaluated, i.e. the partial Gram-Schmidt orthogonalisation (with the optimal value of the dimensional parameter, 60 for the

[3]Note that we used the averaged precision as measure rather than the commonly used F_1 in the experiments, because the F_1 need a parameter to threshold the retrieved results but the averaged precision need not – it was computed from a ranked list of results and we just obtained a ranked list of documents from KCCA. See e.g. [9] for a detailed explanation of the averaged precision.

Set-A and 120 for the Set-B (see Table 4)), perfect clustering, noise clustering, randomly splitting the training set and applying the KCCA directly. The Gram-Schmidt method and the perfect clustering achieved the similar results, which was much better than both the noise clustering and the randomly split. They were even better than applying KCCA directly. Secondly, as we expected, the clustering based method outperformed the randomly split significantly. Thirdly, though the perfect clustering achieved similar performance to the Gram-Schmidt method, noisy clustering can badly decrease the overall performance. Note that in general we are currently unable to perform the perfect clustering in most cases. Therefore, at least for the moment, the clustering method may not be very helpful for KCCA. Hence, the Gram-Schmidt method appears the most practical way for KCCA to deal with large datasets.

We can also see that these methods had the similar behaviors on the Set-A and Set-B, showing that the results here are scalable. In addition, the optimal number of KCCA eigenvectors was 5 for the Set-A and 10 for the Set-B. As the Set-A contained 5 topics and the Set-B contained 10 topics, we may speculate that the optimal number of KCCA eigenvectors for cross-language applications could be around the number of topics in dataset. However, we need do more experiments to verify the speculation.

Table 5: The results of experiments for the Set-A of 5 topics and the Set-B of 10 topics, respectively: means of averaged precisions for cross-language document retrieval. The methods evaluated included the partial Gram-Schmidt, perfect lustering, noise clustering, Randomly split, and applying KCCA directly. Different numbers of eigenvectors were used. An English document was used as a query and all the Japanese documents on the same topic were considered as relevant. Note that we could not compute results for the Set-B for both clustering and randomly split methods in the case of using 5 eigenvectors, because we divided the Set-B into 10 subsets and took at least one eigenvector from every subset.

	#Eigenvectors	**5 (or 1×5)**	**10 (or 2×5 or 1×10)**	**20 (or 4×5 or 2×10)**
Set-A	Gram-Schmidt	0.881	0.775	0.617
	Perfect Clustering	0.880	0.694	0.552
	Noise Clustering	0.639	0.582	0.493
	Randomly Split	0.356	0.340	0.336
	KCCA only	0.834	0.731	0.573
Set-B	Gram-Schmidt	0.809	0.894	0.781
	Perfect Clustering	*	0.900	0.659
	Noise Clustering	*	0.585	0.533
	Randomly Split	*	0.142	0.160
	KCCA only	0.755	0.859	0.761

5 Cross-language document classification

Cross-language document classification is about using a classifier learned from one language to classify documents in other languages, by exploiting the semantic correspondence between the languages. It is useful in the context of multi-lingual information management because by doing so we need not learn different classifiers for multi-lingual document classification (instead we just learn a single classifier and then use it to classify documents in all languages). In addition, the results of cross-language document classification can be used to check how good the semantic correspondence is, since successfully applying a classifier in another language requires a good semantic correspondence between two languages.

As the SVM gives state of the art results for text classification (see [6]), we used the SVM as cross-language document classifier in our experiments. Fortunately, the SVM

learned in one language can be used easily in another language if we are given pairs of the training documents in two languages — we can first train an SVM using documents in one language and then transform it into a new SVM for another language by substituting the training feature vectors in the dual form of the SVM by the mates in another language, since the SVM in dual form is a weighted sum of the training vectors in feature space. On the other hand, the semantic correspondence inferred by e.g. KCCA between the two languages could also be used as a basis to form the correspondence of feature vectors representing the documents in two languages.

We therefore proposed two methods to use the SVM for cross-language classification. The first one was to just exploit pairs of training documents in the two languages, i.e. $\{(x_i, y_i) : i = 1, ..., N\}$. If an SVM was trained from the training documents $\{x_i : i = 1, ..., N\}$ in one language, which can be represented in dual form as

$$h_x(\cdot) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i K(\cdot, x_i)\right) \tag{16}$$

then we can transform it into an SVM classifier in another language as

$$h_y(\cdot) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i K(\cdot, y_i)\right) \tag{17}$$

We call the new SVM classifier (17) *pSVM* since it just employs the semantic correspondence derived directly from the pairings of the training documents in two languages.

Note that this approach can only be applied if the training set is a paired dataset, though one could envisage using the approach by first training an SVM in one language and then only translating the so-called support documents for which the dual variable $\alpha_i > 0$. Typically this only holds for a small subset of the full training set.

Another method used the semantic correspondence derived using KCCA. Given a training set containing pairs of documents in both languages, projecting the training documents onto the semantic space of KCCA resulted in pairs of semantic feature vectors, exactly as we obtained in Section 3 for cross-language information retrieval. These pairs of semantic vectors can then be used to project an SVM classifier from one language into another language, just as was done for the *pSVM*. We call this kind of classifier *kcca_SVM*. Note that crucially the training set for KCCA may be different from that for the SVM. This implies that a large (unlabeled) training set can be used in KCCA to deduce a good semantic correspondence between the two languages and another labeled document set would be used to train the SVM. However, in the experiments described below, only one and the same training set was used for both KCCA and the SVM.

We also implemented an algorithm based on the generalised vector space model (GVSM) for comparison (see [4]). This uses as a semantic feature vector the vector of inner products between a document and the training documents in the same language. We call this kind of SVM classifier *gvsm_SVM*.

The dataset used in our experiment was also from the NTCIR-3 patent retrieval test collection. The collection includes 31 topics. For each topic some pairs of documents in Japanese and English were annotated as relevant or irrelevant. The annotated documents for one topic form a dataset for cross-language document classification, where a classifier can be learned from one language and then be tested in another language.

In the experiments we randomly split the dataset into two equal parts, one for training and another for test. We used the English part of the training documents to train an SVM classifier and then induced the *pSVM*, *kcca_SVM* and *gvsm_SVM* classifiers for the Japanese documents, respectively. In order to test these cross-language classifiers, we also trained an SVM directly using the Japanese training set and tested it on the Japanese test set. As

in the experiments presented in Section 4, We used averaged precision[4] to evaluate the performances of all the SVM classifiers on the Japanese test set (see e.g. [9] for a detailed explanation of the averaged precision). We ran the experiments 10 times for one topic and then the statistical measures $mean$ and std were computed for the averaged precisions from the 10 runs.

Table 6 shows the results for six topics, Topic 01, 02, 03, 07, 12 and 14 in the NTCIR collection. It also lists the numbers of relevant and irrelevant documents for each topic. These topics were selected such that they were variable in respect of the ratio of relevant and irrelevant documents (the ratios for the six topics were from 0.019 to 0.84). For the *kcca_SVM* we present the results with different numbers of eigenvectors derived from KCCA. First, not surprisingly, the SVM learned directly from Japanese training documents achieved the best results on the Japanese test documents. However, the *pSVM* was comparable to the SVM and the performance of *kcca_SVM* did not drop much from that of the SVM, which shows that the cross-language classification is feasible, in particular by using the *pSVM*. Secondly, the *pSVM* and *kcca_SVM* outperformed *gvsm_SVM* on all the six topics, showing that the semantic correspondences of the former two are better than that of the last one. Finally, note that the results were various among the six topics but were consistency among the methods. The result for a topic were dependent upon the topic itself (whether it is hard for classification) as well as the training set (e.g. the number of relevant examples). Moreover, if we had used F_1 as the measure instead of the averaged precision, the differences of the results among the topics would become bigger (see Footnote 4).

Table 6: Result for cross-language classification for six topics: the *mean* and *std* of the averaged precisions over 10 experiments of the SVM classifiers on Japanese test set. The SVM classifiers were learned from English training set and then induced *pSVM*, *kcca_SVM* and GVSM classifiers in Japanese. KCCA_100 means using the 100 eigenvectors with largest eigenvalues from KCCA and KCCA_full using all the eigenvectors. The last column presents the results of the SVM learned directly from the Japanese training documents.

Topic	#docs of relevant/irrelevant	GVSM	pSVM	KCCA_100	KCCA_full	SVM
01	26/811 (0.032)	0.511± 0.090	0.594± 0.123	0.561± 0.90	0.603± 0.087	0.666± 0.087
02	17/912 (0.019)	0.550±0.210	0.711± 0.143	0.603±0.107	0.684±0.139	0.730±0.128
03	33/500 (0.066)	0.111±0.025	0.167±0.038	0.133±0.042	0.131±0.033	0.188±0.050
07	102/264 (0.386)	0.669±0.055	0.749±0.056	0.757±0.042	0.760±0.035	0.767±0.042
12	330/393 (0.840)	0.707±0.041	0.750±0.025	0.728±0.030	0.736±0.025	0.768±0.031
14	64/368 (0.174)	0.692±0.066	0.760±0.052	0.733±0.021	0.715±0.0.49	0.809±0.042

6 Conclusions

We described a method for fully automated cross-language information retrieval in which no query translation was required. The method was based on KCCA, a method of finding out the maximally correlating relationship between documents in two languages. We used KCCA for cross-language Japanese-English information retrieval. The experimental results were quite encouraging and were better than those obtained by another state of the art method CL-LSI. As the computational complexity issue became serious when applying KCCA to large datasets, we investigated several methods to alleviate the computational

[4]We did not use the F_1, a commonly used measure in information retrieval research, to measure the performance, because of another reason rather than the one for the experiments in Section 4 (see Footnote 3). The F_1 is dependent on the bias b of the SVM solution but the average precision is not. It is known that the SVM would learn a poor bias if the number of positive training patterns is very small and the bias can be improved by some algorithms (see [8] and [7]). But our purpose here is to compare different algorithms rather than achieving high value of F_1. Therefore, we think that the averaged precision is a better measure than F_1 for the experiments.

problem with KCCA. Our experiments showed that the partial Gram-Schmidt orthogonalisation was a practical way to help KCCA deal with large datasets.

We also presented two methods for cross-language document classification. The *pSVM* projected the SVM classifier learned in one language onto another language directly through the pairs of training documents in two languages. The *KCCA_SVM* induced an SVM classifier in another language by using the semantic correspondence inferred by KCCA. We tested the methods using an English training set and a Japanese test set. Both methods obtained promising results and the *pSVM* performed better than the *KCCA_SVM*. In comparison with the SVM classifier learned directly from Japanese training documents, the *pSVM* achieved similar performance and the *KCCA_SVM* did not deteriorate much on the six various topics.

Note that we have investigated the capability of KCCA for cross-language information retrieval by implementing some special tasks such as mate retrieval and pseudo query retrieval. Further work is required to implement the KCCA based method incorporating the partial Gram-Schmidt orthogonalisation for more practical information retrieval tasks such as those defined in the NTCIR-3 collection. In the experiments we just used the stemmed words as features for document, which means the feature space is basically vocabularies. It is interesting to incorporate linguistic information (such as the semantic features derived from e.g. the WordNet) into feature space for KCCA. Another interesting work is to investigate further the KCCA based method for cross-language document classification by using two different training sets one for KCCA and the second for the SVM to check if it is more helpful of the semantic representation which is inferred from a larger unlabeled training set. We are also interested in seeking a better approach than the one we used to collect the Japanese terms.

7 Acknowledgments

We would like to thank Alexei Vinokourov and Nello Cristianini for discussions and technical assistance in implementing KCCA. We would also thank Mitsuharu Makita for help in preprocessing Japanese document. We thank anonymous reviewers for detailed comments and valuable suggestions. We thank National Institute of Informatics (NII) for providing the NTCIR-3 patent retrieval test collection. The work described in this paper has been supported by the European Commission through the IST Programme under Contract IST-2000-25431 (KerMIT).

References

[1] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[2] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[3] N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent semantic kernels. *Journal of Intelligent Information System*, 18(2/3):127–152, 2002.

[4] D. R. Hardon, S. Szedmark, and J. Shawe-Taylor. Canonical correlation analysis: an overview with application to learning methods. Technical Report CSD-TR-03-02, Department of Computer Science, Royal Holloway, University of London, 2003.

[5] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:312–377, 1936.

[6] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of*

ECML-98, 10th European Conference on Machine Learning, number 1398 in Lecture Notes in Computer Science, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[7] D.D. Lewis, Y. Yang, T. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

[8] Y. Li and J. Shawe-Taylor. The SVM with uneven margins and Chinese document categorization. In *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*, pages 216–227, Singapore, Oct. 2003.

[9] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. Kandola. The Perceptron Algorithm with Uneven Margins. In *Proceedings of the 9th International Conference on Machine Learning (ICML-2002)*, pages 379–386, 2002.

[10] M. L. Littman, S. T. Dumais, and T. K. Landauer. Automatic cross-language information retrieval using latent semantic indexing. In G. Grefenstette, editor, *Cross language information retrieval*. Kluwer, 1998.

[11] M. Makita, S. Higuchi, A. Fujii, and T. Ishikawa. A system for Japanese/English/Korean multilingual patent retrieval. In *Proceedings of Machine Translation Summit IX (online at http://www.amtaweb.org/summit/MTSummit/papers.html)*, Sept. 2003.

[12] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances of Neural Information Processing Systems 15*, 2002.