Bilingual Text Classification using the IBM 1 Translation Model

Jorge Civera and Alfons Juan-Císcar

ITI/DSIC, Universidad Politécnica de Valencia
jorcisai@iti.upv.es ajuan@dsic.upv.es

Abstract

Manual categorisation of documents is a time-consuming task that has been significantly alleviated with the deployment of automatic and machine-aided text categorisation systems. However, the proliferation of multilingual documentation has become a common phenomenon in many international organisations, while most of the current systems has focused on the categorisation of monolingual text. It has been recently shown that the inherent redundancy in bilingual documents can be effectively exploited by relatively simple, bilingual naive Bayes (multinomial) models. In this work, we present a refined version of these models in which this redundancy is explicitly captured by a combination of a unigram (multinomial) model and the well-known IBM 1 translation model. The proposed model is evaluated on two bilingual classification tasks and compared to previous work.

1. Introduction

Historically, the manual categorisation of documents has entailed a time-consuming and arduous task that has been significantly alleviated with the deployment of automatic and machine-aided text categorisation systems (Sebastiani, 2002; Hodge, 1998). However, nowadays the proliferation of multilingual documentation has become a common phenomenon in many international organisations, while most of the current systems has focused on the categorisation of monolingual text. Nonetheless there are notable exceptions in the field of cross-lingual information retrieval (CLIR) and text categorisation (CLTC) in which bilingual sources are employed (Grefenstette, 1998; Bel et al., 2003). In this paper, we present an application that differs from that of CLIR and CLTC, since we want to classify bilingual pairs of documents that are translations of each other. We believe that by doing so, we can fully exploit the word correlation across languages using a translation model in a more natural way than CLIR and CLTC do using external translation resources.

Here we introduce an evolution of the relatively simple bilingual multinomial models presented in (Civera and Juan, 2006a; Civera and Juan, 2006b) in order to exploit the structural information in word correlation in bilingual texts. To this purpose a novel model inspired in the combination of a unigram (multinomial) model and the well-known IBM 1 translation model is proposed. The resultant bilingual classifier was evaluated on the Traveller task and the BAF corpus, and compared to previous work.

2. Bilingual text classification

Generally speaking, the task of bilingual text classification consists in assigning unlabelled bilingual pairs of texts to a set of predefined categories. As stated before, every pair of texts has the peculiarity of being mutual translations. Formally, according to the Bayes decision rule, given a bilingual pair of texts (x, y) we will assign this pair to that category $\hat{c}$ that maximises the posterior probability:

$$\begin{aligned}\hat{c} &= \operatorname*{argmax}_c \; p(c \,|\, x, y) \\ &= \operatorname*{argmax}_c \; p(c)\, p(x, y \,|\, c)\end{aligned} \tag{1}$$

where $p(c)$ is the prior probability of category c usually estimated as the relative frequency of category c in the training set, and $p(x, y \,|\, c)$ is the category-conditional probability of (x, y) given that was generated by category c. The modelisation and estimation of $p(x, y \,|\, c)$ is presented in Sections 3. and 4..

3. The unigram-IBM1 model

The probability of a given pair $p(x, y)$[1] can be decomposed into a *target language* probability, $p(y)$, and a *translation* probability, $p(x \,|\, y)$:

$$p(x, y) = p(y)\, p(x \,|\, y) \tag{2}$$

The target language probability can be written in terms of individual, target-word probabilities as follows:

$$p(y) = \prod_{i=1}^{|y|} p(y_i \,|\, y_1^{i-1}) \tag{3}$$

by assuming that the probability of each target word to occur does not depend on any previous word,

$$p(y_i \,|\, y_1^{i-1}) := p(y_i) \tag{4}$$

we have the unigram language model:

$$p(y) = \prod_{i=1}^{|y|} p(y_i) \tag{5}$$

For the translation probability, as in conventional statistical machine translation, we introduce the alignment hidden variable $a = a_1 \cdots a_j \cdots a_{|x|}$ that connects each source word to exactly one target word $a_j = \{0, \cdots, i, \cdots, |y|\}$, being 0 the position of the NULL word:

$$p(x \,|\, y) = \sum_{a \in \mathcal{A}(x,y)} p(x, a \,|\, y) \tag{6}$$

Work supported by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01, the *Consellería d'Empresa, Universitat i Ciència - Generalitat Valenciana* under contract GV06/252, the *Universidad Politécnica de Valencia* with ILETA project and Ministerio de Educación y Ciencia.

[1] We have simplified the notation by dropping the dependency on c to avoid repetition and ease the comprehension of the model.

where $\mathcal{A}(x,y)$ denotes the set of all possible alignments between x and y. Then,

$$p(x,a\,|\,y) = \prod_{j=1}^{|x|} p(x_j, a_j \,|\, x_1^{j-1}, a_1^{j-1}, y) = \prod_{j=1}^{|x|} p(a_j \,|\, x_1^{j-1}, a_1^{j-1}, y)\, p(x_j \,|\, x_1^{j-1}, a_1^{j}, y) \quad (7)$$

In order to define the well-known IBM model 1 (Brown and others, 1993), we make the following two assumptions:

$$p(a_j \,|\, x_1^{j-1}, a_1^{j-1}, y) := \frac{1}{|y|+1} \quad (8)$$

$$p(x_j \,|\, x_1^{j-1}, a_1^{j}, y) := p(x_j \,|\, y_{a_j}) \quad (9)$$

where in Eq. (8) the probability of aligning a source position to a target position is uniform and, in Eq. (9) the probability of translating a source word does only depend on the target word to which is aligned.
Finally the IBM model 1 is:

$$p(x\,|\,y) = \sum_a \prod_{j=1}^{|x|} \frac{1}{|y|+1}\, p(x_j \,|\, y_{a_j}) = \prod_{j=1}^{|x|} \sum_{a_j=0}^{|y|} \frac{1}{|y|+1}\, p(x_j \,|\, y_{a_j}) \quad (10)$$

Putting Eqs. (5) and (10) together we define the *unigram-IBM1* model:

$$p(x,y;\boldsymbol{\Theta}) = \prod_{i=1}^{|y|} p(y_i) \prod_{j=1}^{|x|} \sum_{a_j=0}^{|y|} \frac{1}{|y|+1}\, p(x_j \,|\, y_{a_j}) \quad (11)$$

where $\boldsymbol{\Theta}$ is its vector of parameters defined as:

$$\boldsymbol{\Theta} = \begin{cases} p(v) & v \in \mathcal{Y} \\ p(u\,|\,v) & u \in \mathcal{X}, v \in \mathcal{Y} \end{cases} \quad (12)$$

being $\mathcal{X}$ and $\mathcal{Y}$, source and target vocabularies, respectively. The actual value of this vector of parameters is computed according to the maximum likelihood estimation criterion.
Let $(x_1,y_1),\ldots,(x_N,y_N)$ be N independent samples from a unigram-IBM1 model of parameters $\boldsymbol{\Theta}$. The log-likelihood function of $\boldsymbol{\Theta}$ is

$$L(\boldsymbol{\Theta}) = \sum_n \log p(x_n, y_n; \boldsymbol{\Theta}) \quad (13)$$

Our goal is to estimate a vector of parameters $\boldsymbol{\Theta}$ that maximises Eq. (13). This maximisation cannot be performed directly since Eq. (13) contains missing data, that is, the alignment variables. Therefore we need to revert to the Expectation-Maximisation (EM) algorithm in order to estimate the vector of parameters $\boldsymbol{\Theta}$.
The EM algorithm consists of two basic steps applied iteratively. The E step computes the expected value of the missing data given the training data and the current parameters $\boldsymbol{\Theta}^{(k)}$. The M step finds a new vector of parameter values $\boldsymbol{\Theta}^{(k+1)}$ which maximises the complete version of Eq. (13) on the basis of the missing data estimated in the E step.
In our case, the E step computes the expected value of the alignment variable a_n for each sample (x_n,y_n) as follows:

$$a_{nji}^{(k)} = \frac{p(x_{nj}\,|\,y_{ni})^{(k)}}{\sum_{i'=0}^{|y_n|} p(x_{nj}\,|\,y_{ni'})^{(k)}} \quad (14)$$

That is, the expectation of word x_{nj} to be connected to y_{ni} is our current estimation of the probability of x_{nj} to be a translation of y_{ni} instead of any other word in y_n (including the NULL word).
In the M step, the computation of a new estimate for the parameter values is decomposed into a conventional solution for the unigram language model,

$$p(v)^{(k+1)} = \frac{\sum_n \sum_{i:y_{ni}=v} 1}{\sum_{v'} \sum_n \sum_{i:y_{ni}=v'} 1} \qquad \forall v \in \mathcal{Y} \quad (15)$$

and the standard update formula for the IBM Model 1,

$$p(u|v)^{(k+1)} = \frac{\sum_n \sum_{j:x_{nj}=u} \sum_{i:y_{ni}=v} a_{nji}^{(k)}}{\sum_{u'} \sum_n \sum_{j:x_{nj}=u'} \sum_{i:y_{ni}=v} a_{nji}^{(k)}} \quad \forall u \in \mathcal{X}, v \in \mathcal{Y} \quad (16)$$

Note that Eq. (15) is simply the relative frequency of occurrence of word v in the target texts and, hence, it does not change over successive iterations of the EM.

4. The unigram-IBM1 mixture model

Eq. (11) is a relatively simple parametric model for distributions of bilingual pairs of texts. Then, it is a good choice to describe simple distributions, but it might not be so good to approximate complex distributions, such as those comprising topically-unrelated groups of bilingual pairs. To deal with such cases, we will use the idea of mixture modelling and replace our simple model by a finite mixture.
Let us assume that bilingual pairs come from T different *topics*. Then, the probability function (p.f.) of a given pair can be appropriately described as a *finite mixture:*

$$p(x,y) = \sum_{t=1}^{T} p(t)\, p(x,y\,|\,t) \quad (17)$$

where t is the topic variable and, for each topic t, $p(t)$ is its *prior* or *coefficient* and $p(x,y\,|\,t)$ is its *topic-conditional p.f.* It can be seen as a generative model that first selects the tth topic with probability $p(t)$ and then generates (x,y) in accordance with $p(x,y\,|\,t)$.
We can further factorised the term $p(x,y\,|\,t)$ in a similar manner to Section 4., but including the topic variable:

$$p(x,y\,|\,t) = p(y\,|\,t)\, p(x\,|\,y,t) \quad (18)$$

where $p(y\,|\,t)$ and $p(x\,|\,y,t)$ are topic-dependent versions of Eq. (3) and (6).

We assume that each topic prior $p(t)$ is given by a parameter $p(t)$, and that each topic-conditional p.f. $p(x, y \mid t)$ can be approximated using a topic-conditional unigram-IBM1 model. Thus, our finite mixture model (17) for $p(x, y)$ is

$$p(x, y; \mathbf{\Theta}) = \sum_{t=1}^{T} p(t)\, p(x, y \mid t; \mathbf{\Theta}_t) \tag{19}$$

where

$$p(x, y \mid t; \mathbf{\Theta}_t) = \prod_{i=1}^{|y|} p(y_i \mid t) \prod_{j=1}^{|x|} \sum_{a_{jt}=0}^{|y|} \frac{1}{|y|+1}\, p(x_j \mid y_{a_{jt}}, t) \tag{20}$$

The global vector of parameters $\mathbf{\Theta}$ is:

$$\mathbf{\Theta} = (p(1), \ldots, p(T); \mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_T)^t \tag{21}$$

where each component has its own vector of parameters:

$$\mathbf{\Theta}_t = \begin{cases} p(v \mid t) & v \in \mathcal{Y} \\ p(u \mid v, t) & u \in \mathcal{X}, v \in \mathcal{Y} \end{cases} \tag{22}$$

The estimation of the parameters of the model is performed using the EM algorithm, as we did in Section 3..
In this case, the E-step reduces to compute a topic-dependent alignment hidden variable:

$$a_{njit}^{(k)} = \frac{p(x_{nj} \mid y_{ni}, t)^{(k)}}{\sum_{i'} p(x_{nj} \mid y_{ni'}, t)^{(k)}} \tag{23}$$

where $a_{njit}^{(k)}$ is the posterior probability of the source position j to be aligned to the target position i in the tth component for the nth sample (x_n, y_n).
In the M step, we obtain a new vector of parameters. The component priors:

$$p(t)^{(k+1)} = \frac{1}{N} \sum_{n} z_{nt}^{(k)} \quad \forall t \tag{24}$$

an update equation for the topic-dependent unigram:

$$p(v \mid t)^{(k+1)} = \frac{\sum_{n} z_{nt}^{(k)} \sum_{i:y_{ni}=v} 1}{\sum_{v'} \sum_{n} z_{nt}^{(k)} \sum_{i:y_{ni}=v'} 1} \quad \forall t, v \in \mathcal{Y} \tag{25}$$

and an update equation for the topic-dependent IBM 1:

$$p(u \mid v, t)^{(k+1)} = \frac{\sum_{n} z_{nt}^{(k)} \sum_{j:x_{nj}=u} \sum_{i:y_{ni}=v} a_{njit}^{(k)}}{\sum_{u'} \sum_{n} z_{nt}^{(k)} \sum_{j:x_{nj}=u'} \sum_{i:y_{ni}=v} a_{njit}^{(k)}} \tag{26}$$

for all t, $u \in \mathcal{X}$ and $v \in \mathcal{Y}$.

5. Experimental results

The unigram-IBM1 model described in the previous section was assessed on two tasks: the *Traveller* dataset and the *BAF* corpus. The *Traveller* dataset comes from a *limited-domain* Spanish-English machine translation application for human-to-human communication situations in the front-desk of a hotel (Vidal and others, 2000). It was semi-automatically built from a small "seed" dataset of sentence pairs collected from traveller-oriented booklets by four persons. Each person had to cater for a (non-disjoint) subset of subdomains, and thus it can be considered as a different (multimodal) class of Spanish-English sentence pairs. Subdomain overlapping among classes foresees that perfect classification is not possible, although in our case, low classification error rates will indicate that our mixture model has been able to capture the multimodal nature of the data. Some statistics of this dataset are shown in Table 1.
The *BAF* corpus (Simard, 1998) is a compilation of bilingual "institutional" French-English texts ranging from debates of the Canadian parliament (Hansard), court transcripts and UN reports to scientific, technical and literary documents. This dataset is composed of 11 documents that are organised into 4 natural genres (Institutional, Scientific, Technical and Literary) trying to be representative of the types of text that are available in multilingual versions. The Institutional and Scientific classes comprises documents from the original pool of 11 documents, which were theme-related, but devoted to heterogeneous purposes or written by different authors. This fact provides the multimodal nature to the *BAF* corpus that can be adequately modelled by mixture models. As it can be seen in Table 1, this corpus is more complex than the *Traveller* dataset.

Table 1: *Traveller* and *BAF* corpora statistics.

	Traveller		*BAF*	
	Sp	En	Fr	En
sentence pairs	8000		18509	
average length	9	8	28	23
vocabulary size	679	503	20296	15325
singletons	95	106	8084	5281
running words	86K	80K	522K	441K

Several experiments were carried out to analyse the unigram-IBM1 classifier in terms of classification error rate as a function of the number of mixture components per class ($T = 1, 2, 5, 10, 20, 50, 100$). The results are shown in Figure 1, together with those of best monolingual (English-based) and the best unigram-based bilingual global classifier from (Civera and Juan, 2006b). Each plotted point is an average over values from 30 random training-test splits, as defined in (Civera and Juan, 2006b); 50%-50% (training-test) in *Traveller* and 80%-20% in *BAF*. From the results in Figure 1, we can see that the unigram-IBM1 classifier outperforms both classifiers, especially in the case of the *BAF* corpus. Therefore, the cross-lingual word correlation information provided by the IBM1 model helps to improve the accuracy of its associated classifier.
Table 2 presents a summary of error figures on the Traveller task and the BAF corpus for different classifiers, including support vector machines (SVM) and boosting techniques.
On the one hand, SVM were originally thought as binary classifiers, although there have been a generalisation of the 2-class problem (Crammer and Singer, 2002). In practise binary classifiers based on the one-against-one approach, among others, seem to be the most adequate (Hsu and Lin, 2002). This simple yet effective approach consists in defining as many binary classifiers as possible class pairs, then each binary classifier votes for a class and finally, we classify according to the majority voting criteria. In this pa-

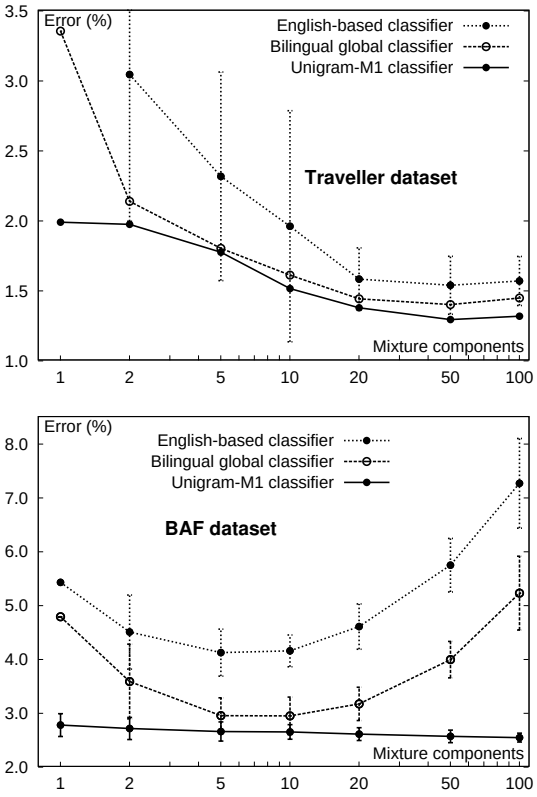

Figure 1: Competing curves: %Error vs. mixture components for *Traveller* and *BAF*.

Table 2: Competing error figures for different classifiers on the Traveller task and the BAF corpus.

	1gm	1g1gm	1gM1m	SVM^{light}	BoosTexter
Traveller	1.5	1.4	1.3	1.5	1.2
BAF	4.1	3.0	2.5	9.0	5.8

per, all the SVM experiments were carried out with the SVM^{light} toolkit (Joachims, 1999) adopting the approach to the multi-class problem commented above . On the other hand, the idea behind boosting methods is to find a highly accurate classification rule by combining many weak hypotheses, each of which may be only moderately accurate. The implementation of the boosting algorithm employed in this paper is BoosTexter (Schapire and Singer, 2000).

As we can observe in Table 2, the unigram-M1 mixture model (1gM1m) supersedes the other two unigram mixture models, monolingual (1gm) and bilingual global (1g1gm), being statistically significant better in the case of the BAF corpus, but not being so for the Traveller task. The unigram-M1 mixture model obtains similar performance to SVM and boosting methods in the Traveller task, and statistically significantly better in the BAF corpus. These experiments show the benefits of learning word correlation across languages in bilingual text classification.

6. Conclusions and future work

We have presented a novel model for bilingual text classification in which the crosslingual structure is incorporated by using the well-known IBM model 1. Doing so, we outperform the accuracy of the simple bilingual global classifier that considers each language separately.

As shown in the results on the *T*raveller and *B*AF corpora, the unigram-IBM1 model statistically significantly surpasses the bilingual unigram classifiers.

Apart from the model presented in this paper, we studied the performance of a bilingual classifier when upgrading from IBM model 1 to IBM model 2 in the unigram-IBM1 model. IBM model 2 provides a non-uniform alignment p.f. between source and target sentence positions refining the uniform alignment distribution assumed by IBM model 1. Despite this refinement, the unigram-IBM2 model suffered from severe data sparseness, and its performance was worse than that of the unigram-IBM1 model.

As a future work we plan to explore the combination of smooth n-gram models with IBM model 1 in the powerful framework of mixture modelling. Moreover, the incorporation of bilingual classes (Och, 1999) is an interesting approach to control the model complexity in the presence of data scarcity problems, specifically the number of parameters in modelling topic-dependent statistical dictionaries by adjusting the number of word classes. Another appealing issue for future work is the automatic estimation of the number of components in the mixture using model selection methods such as, variational EM, BIC or MDL.

7. References

N. Bel, C.H.A. Koster, and M. Villegas. 2003. Cross-lingual text categorization. In *ECDL'03*, pages 126–139.

P. F. Brown et al. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

J. Civera and A. Juan. 2006a. Bilingual Machine-Aided Indexing. In *Proc. of LREC'06*, pages 1302–1305.

J. Civera and A. Juan. 2006b. Multinomial Mixture Modelling for Bilingual Text Classification. In *Proc. of PRIS'06*, pages 93–103.

K. Crammer and Y. Singer. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *Jour. Mach. Learn. Research*, 2:265–292.

G. Grefenstette. 1998. *Cross-Language Information Retrieval*. Kluwer Academic Publishers, USA.

G. Hodge. 1998. CENDI agency indexing system descriptors: A Baseline Report. Technical report, IIA, Inc.

Ch-W. Hsu and Ch-J. Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

T. Joachims. 1999. Making large-scale support vector machine learning practical. *Advances in kernel methods: support vector learning*, pages 169–184.

F. J. Och. 1999. An efficient method for determining bilingual word classes. In *Proc. of EACL'99*, pages 71–76.

R. E. Schapire and Y. Singer. 2000. Boostexter: A boosting-based systemfor text categorization. *Machine Learning*, 39(2-3):135–168.

F. Sebastiani. 2002. Machine learning in automated text categorisation. *ACM Computing Surveys*, 34(1):1–47.

Michel Simard. 1998. The BAF: A Corpus of English-French Bitext. In *Proc. of LREC'98*, pages 489–496.

E. Vidal et al. 2000. Example-Based Understanding and Translation Systems. ESPRIT project 20268 report.