Journal of Artificial Intelligence Research 57 (2016) 151-185 Submitted 05/16; published 10/16

Lightweight Random Indexing for Polylingual Text Classification

Alejandro Moreo Fernández ALEJANDRO.MOREO@ISTI.CNR.IT
Andrea Esuli ANDREA.ESULI@ISTI.CNR.IT
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, IT

Fabrizio Sebastiani FSEBASTIANI@QF.ORG.QA
Qatar Computing Research Institute
Hamad bin Khalifa University
PO Box 5825, Doha, QA

Abstract

Multilingual Text Classification (MLTC) is a text classification task in which documents are written each in one among a set L of natural languages, and in which all documents must be classified under the same classification scheme, irrespective of language. There are two main variants of MLTC, namely *Cross-Lingual Text Classification* (CLTC) and *Polylingual Text Classification* (PLTC). In PLTC, which is the focus of this paper, we assume (differently from CLTC) that for each language in L there is a representative set of training documents; PLTC consists of improving the accuracy of each of the $|L|$ monolingual classifiers by also leveraging the training documents written in the other $(|L| - 1)$ languages. The obvious solution, consisting of generating a single polylingual classifier from the juxtaposed monolingual vector spaces, is usually infeasible, since the dimensionality of the resulting vector space is roughly $|L|$ times that of a monolingual one, and is thus often unmanageable. As a response, the use of machine translation tools or multilingual dictionaries has been proposed. However, these resources are not always available, or are not always free to use.

One machine-translation-free and dictionary-free method that, to the best of our knowledge, has never been applied to PLTC before, is *Random Indexing* (RI). We analyse RI in terms of space and time efficiency, and propose a particular configuration of it (that we dub *Lightweight Random Indexing* – LRI). By running experiments on two well known public benchmarks, Reuters RCV1/RCV2 (a comparable corpus) and JRC-Acquis (a parallel one), we show LRI to outperform (both in terms of effectiveness and efficiency) a number of previously proposed machine-translation-free and dictionary-free PLTC methods that we use as baselines.

1. Introduction

With the rapid growth of multicultural and multilingual information accessible on the Internet, how to properly classify texts written in different languages has become a problem of relevant practical interest. *Multilingual Text Classification* (MLTC) is a text classification task in which documents are written each in one among a set $L = \{l_1, \ldots, l_{|L|}\}$ of natural languages, and in which all documents must be classified under the same classification scheme, irrespective of the language. There are two main variants of MLTC, namely *Cross-Lingual Text Classification* (CLTC) and *Polylingual Text Classification* (PLTC).

CLTC is a task characterized by the fact that, for all languages in a subset $L_T \subset L$, there are no training documents; the task thus consists of classifying the unlabelled documents written in the languages in L_T (i.e., the *target* languages) by leveraging the training documents expressed in the other languages $L_S = L \backslash L_T$ (i.e., the *source* languages). CLTC is thus a *transfer learning* problem (Pan & Yang, 2010), where one needs to transfer the knowledge acquired by learning from the training data in L_S, to the task of classifying documents in L_T. Most previous work on MLTC indeed focuses on CLTC, and fewer efforts have been devoted to PLTC, which is instead the focus of this paper.

In PLTC, a representative set of training documents for all languages in L is assumed to be available. Therefore, a straightforward solution may consist in training $|L|$ independent monolingual classifiers, one for each language. However, such solution is suboptimal, as each classifier is obtained by disregarding the additional supervision that could be obtained by using the training documents written in the other $(|L| - 1)$ languages. PLTC thus consists of leveraging the training documents written in *all* languages in L to improve the classification accuracy that could be obtained by simply training the $|L|$ independent, monolingual classifiers.

However, PLTC entails a number of obstacles that work to the detriment of efficient representation. To see this, assume we generate a single polylingual vector space (hereafter, the *juxtaposed vector space*) by juxtaposing the monolingual vector spaces. The vector space for a monolingual dataset usually consists of tens or even hundreds of thousands of features; for the juxtaposed vector space of a polylingual dataset, this dimensionality gets roughly multiplied by the number of distinct languages under consideration. Such a substantial increase in the feature space would degrade the performance of many classification algorithms, because of the so-called "curse of dimensionality", and would also bring about a severe degradation in efficiency. Additionally, co-occurrence-based techniques tend to lose power when representations are polylingual, since terms belonging to different languages rarely co-occur, if at all (a problem usually referred to as *feature disjointness*).

As a response, some authors have proposed the use of machine translation (MT) tools as a device to simultaneously cope with both high dimensionality and feature disjointness in PLTC. The idea is to reduce the problem to the monolingual case (typically English). That is, non-English training documents are automatically translated into English, are added to the English training set, and a monolingual (English) classifier is trained. At classification time, non-English unlabelled documents are translated into English and are then classified. (Of course, this idea can also be used in CLTC; in this case, there are no training documents to translate.) However, these MT-based PLTC (and CLTC) techniques suffer from a number of drawbacks (Wei, Yang, Lee, Shi, & Yang, 2014): (i) automatically translated texts usually present different statistical properties with respect to human translations; (ii) MT systems are not always available for all language pairs; and (iii) training a statistical MT system from any of the free toolkits available requires collecting large corpora of parallel text in the domain of interest, which is not always easy.

Thesaurus-based and dictionary-based methods, on the other side, represent a lighter approach in MLTC. If a multilingual dictionary or thesaurus that encompasses the different languages is available, some kind of unification of the vector representation may be attempted. This is customarily done by replacing non-English words with their English equivalents in the dictionary, or by replacing all terms with thesaurus codes invariant

across languages (e.g., BabelNet synsets – Ehrmann, Cecconi, Vannella, McCrae, Cimiano, & Navigli, 2014). However, bilingual dictionaries or thesauri are not available for all language pairs, and automatically constructing a domain-dependent bilingual resource requires a suitable parallel corpus with sentence-level alignment.

1.1 Distributional Representations

For classification purposes, a textual document is usually represented as a vector in a vector space according to the bag-of-words (BoW) model, i.e., each distinct term corresponds to a dimension of the vector space. In the juxtaposed vector space, most of the columns in the document-by-term matrix are thus informative for only one of the languages.

Since each distinct term corresponds to a dimension of the vector space, the BoW model is agnostic with respect to semantic similarities among terms. That is, the dimension for term "governor" is orthogonal to the dimension for the related term "president", as it is to the dimension for the unrelated term "transport". The semantic relations among terms can be uncovered by detecting their co-occurrences, i.e., the contexts in which words tend to be used together. This idea rests on the *distributional hypothesis*, according to which words with similar meanings tend to co-occur in the same contexts (Harris, 1968). By detecting co-occurrences, it is possible to establish a parallelism between term meaning and geometrical properties in the vector space. *Distributed Semantic Models* (DSMs – sometimes also called "word space models" in Sahlgren, 2006) aim at learning continuous and compact distributed term representations, which have recently been called *word embeddings* (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013b). DSMs have gained a lot of attention from the machine learning community, delivering improved results in many natural language processing tasks (Bengio, Schwenk, Senécal, Morin, & Gauvain, 2006; Bullinaria & Levy, 2007; Collobert, Weston, Bottou, Karlen, Kavukcuoglu, & Kuksa, 2011). DSM-based methods can be categorised (see Pennington, Socher, & Manning, 2014; Baroni, Dinu, & Kruszewski, 2014) as belonging (a) to the class of *context-counting* models, which are often based on matrix factorization, e.g., Latent Semantic Analysis (LSA – Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Österlund, Ödling, & Sahlgren, 2015), or (b) to the class of *context-predicting* models, e.g., methods based on deep learning architectures (Bengio, 2009; Mikolov et al., 2013b).

However, in multilingual contexts huge quantities of plain text for each language should be processed in order to learn meaningful word representations, which incurs high computational costs. Trying to find such representations for a large multilingual vocabulary can thus become computationally prohibitive. Some attempts have recently been made in this direction, by leveraging multilingual external resources such as Wikipedia articles (Al-Rfou', Perozzi, & Skiena, 2013), or bilingual dictionaries (Gouws & Søgaard, 2015), or word-aligned parallel corpora (Klementiev, Titov, & Bhattarai, 2012), or sentence-aligned parallel corpora (Zou, Socher, Cer, & Manning, 2013; Hermann & Blunsom, 2014; Lauly, Boulanger, & Larochelle, 2014; Chandar, Lauly, Larochelle, Khapra, Ravindran, Raykar, & Saha, 2014), or document-aligned parallel corpora (Vulić & Moens, 2015). However, such external resources may not always be available for all language combinations and, when they are available (e.g., Wikipedia articles), they may be of uneven quality and quantity for languages other than English. Alternatively, other approaches require a computationally

expensive post-processing step to align word representations across languages (Mikolov, Le, & Sutskever, 2013a; Faruqui & Dyer, 2014).

In this article we discuss efficient representation mechanisms for PLTC that (i) are MT-free, (ii) do not require external resources, and (iii) do not incur high computational costs. In particular, we investigate the suitability of *Random Indexing* (RI – Kanerva, Kristofersson, & Holst, 2000; Sahlgren, 2005) as an effective representation mechanism of the original co-occurrence matrix in PLTC. RI is a context-counting model belonging to the family of *random projections* methods (Kaski, 1998; Papadimitriou, Raghavan, Tamaki, & Vempala, 1998), that produces linear projections into a nearly-orthogonal reduced space where the original distances between vectors are approximately preserved (Hecht-Nielsen, 1994; Johnson, Lindenstrauss, & Schechtman, 1986). RI is expected to deliver fast and semantically meaningful representations in a reduced space, and can be viewed as a cheaper approximation of LSA (Sahlgren, 2005). RI is such that each column from the polylingual matrix produced by it will not depend on any single specific language (as it does instead in the BoW representation). We hypothesize this could be advantageous in PLTC, since the entire new space becomes potentially informative *for all languages at once*, thus making the problem more easily separable if enough dimensions are considered. While RI has already been applied to bilingual scenarios (Gorman & Curran, 2006; Sahlgren & Karlgren, 2005), to the best of our knowledge it has not been tested on the PLTC case so far. In monolingual TC, RI was found to be competitive, but not superior, to BoW (Sahlgren & Cöster, 2004). In this article we demonstrate that RI outperforms the BoW model in PLTC.

The method we present in this article, that we dub *Lightweight Random Indexing* (LRI), is inspired by the works of Achlioptas (2001) and Li, Hastie, and Church (2006) on very sparse random projections, and goes one step further by pushing sparsity to the limit. LRI is designed so that the orthogonality of the projection base is maximized, which causes sparsity to be preserved after the projection. We empirically show that LRI helps Support Vector Machines (SVMs) to deliver better classification accuracies in PLTC with respect to many popular alternative vector space models (including the main random projection variants, LSA-based approaches, and polylingual topic models), while also requiring substantially less computation effort.

The contribution of this work is twofold. First, we conduct a comparative empirical study of several PLTC approaches in two representative scenarios: the first is when the training corpus is *comparable* at the topic-level (i.e., documents are not direct translations of each other, but are simply about similar topics; this is here exemplified by the RCV1/RCV2 dataset), and the second is when the training corpus is *parallel* at the document-level (i.e., each text is available in all languages thanks to the intervention of human translators; this scenario is exemplified by the JRC-Acquis dataset). We show that LRI yields the best results in both settings, in terms of both effectiveness and efficiency. As a second contribution, we present an analytical study that can be useful to better understand the nature of random mapping methods.

The rest of this paper is organized as follows. In Section 2 we discuss related work. In Section 3 we present the problem statement, describe the Random Indexing method in detail, and present our proposal. Section 4 reports the results of the experiments we have conducted. Section 5 presents an analytical study on computational efficiency, while Section 6 concludes.

2. Related Work

This section gives an overview of the main approaches to PLTC that have emerged in the literature. We distinguish three groups of methods, according to whether the problem is approached (i) by leveraging external resources, (ii) by combining the outcome of independent monolingual classifiers, or (iii) by reducing the dimensionality of the resulting multilingual feature space. This discussion also includes some references to CLTC techniques that we consider relevant to PLTC and to our approach.

2.1 Exploiting External Multilingual Resources

Multilingual text classification is a relatively recent area of research, and most previous efforts within it were devoted to the CLTC subtask. As in CLTC there is no labelled information for all languages, previous approaches typically relied on automatic translation mechanisms as a means to fill the gap between the source and the target languages. The main difference between CLTC and PLTC lies in the fact that PLTC exploits labelled documents belonging to different languages during learning. Despite this, the two tasks have a close-knit relation, since in both of them cross-lingual adaptation is generally carried out by means of external resources, such as parallel corpora, bilingual dictionaries, and statistical thesauri.

If a suitable (unlabelled) multilingual corpus containing short aligned pieces of texts is available, correlations among groups of words in the two languages could be explored. Cross-Lingual Kernel Canonical Correlation Analysis (CL-KCCA) was proposed by Vinokourov, Shawe-Taylor, and Cristianini (2002) as a means to obtain a semantic cross-lingual representation, by investigating correlations between aligned text fragments. CL-KCCA takes advantage of kernel functions in order to map aligned texts into a high-dimensional space in such a manner that the correlations between the mapped aligned texts are jointly maximized. This cross-lingual representation could then be used for classification, retrieval, or clustering tasks. CL-KCCA was investigated in combination with Support Vector Machines (SVMs) and applied to cross-lingual patent classification by Li and Shawe-Taylor (2007). Their method, called SVM_2k, learns two SVM-based classifiers by searching two linear projections in the original feature space of each language such that the distance of the projections (instead of the correlation of the projections) of two aligned texts is minimized.

In a similar vein, polylingual topic models (Mimno, Wallach, Naradowsky, Smith, & McCallum, 2009) have been proposed as an extension of Latent Dirichlet Allocation (LDA – Blei, Ng, & Jordan, 2003) to the polylingual case. LDA is a generative model which assigns probability distributions to documents over latent topics, and to latent topics over terms. These distributions can be viewed as compact representations for documents in a latent space. Since topics discovered by Polylingual LDA (PLDA) are aligned across all languages, documents are represented in a common vector space regardless of the language they are written in. However, PLDA (which we will use as a baseline in the experimental section) requires a parallel collection of documents aligned at the sentence level.

Bilingual dictionaries can be used in a straightforward manner to carry out a word-by-word translation of the feature space. However, dictionary-based translations suffer from several deficiencies, e.g., context-unaware translations might perform poorly when handling polysemic words; dictionaries might suffer from a substantial lack of coverage of novel terms

and domain-dependent terminology; and dictionaries might not be available for all language pairs, or not be free to use. As a response to these drawbacks, the automatic acquisition of statistical bilingual dictionaries has been proposed. Wei et al. (2014) explored a co-occurrence-based method to measure the polylingual statistical strength of the correlation among words in a parallel corpus. These correlations are then taken into account to reinforce the weight of each feature in order to select the most important (highly weighted) ones. Gliozzo and Strapparava (2006) experimented with bilingual dictionaries and, more interestingly, provided a means to automatically obtain a Multilingual Domain Model (MDM), a natural extension of domain models to multiple languages, when no additional multilingual resources are available. A *domain model* defines soft relations between words and domain topics. In the absence of a multilingual dictionary, a MDM could be automatically obtained from a comparable corpus by performing Latent Semantic Analysis (explained in more detail below).

It has been argued that words that are shared across languages play an important role when searching the semantic latent space. Accordingly, Steinberger, Pouliquen, and Ignat (2004) exploit language-independent tokens which are shared across the languages, and propose a simple method to link documents with existing external resources such as thesauri, nomenclatures, and gazetteers. Finally, de Melo and Siersdorfer (2007) use ontologies to map original features onto synset-like identifiers, so that the documents are translated into a language-independent feature space.

MT tools, on the other side, provide more elaborated translations of texts, and represent a promising research field for multilingual tasks. Unfortunately, the above-mentioned problems regarding availability, accessibility, and performance still hold in this case. The effect of different translation strategies on CLTC has been investigated by Bel, Koster, and Villegas (2003), Rigutini, Maggini, and Liu (2005), and Wei, Lin, and Yang (2011).

Even when available, MT tools may be expensive resources. For this reason, in their experiments Prettenhofer and Stein (2010) restrict the use of an MT tool to a limited budget of calls. Their Structural Correspondence Learning (SCL) method, initially proposed for domain adaptation, was indeed applied to CLTC. The key idea of the method consists of discovering cross-lingual correspondences between pairs of terms (dubbed *pivot features*) that are later used to bridge across the two languages. Pivot features play an important role in bilingual tasks, since they establish pairs of words that behave similarly in the source and target languages, allowing one to find cross-language structural correspondences. One such special type of pivot features are obviously the words shared across languages, such as proper nouns, technical terms, not yet lexicalized terms, or stemmed forms of etymologically related terms. Nastase and Strapparava (2013) found that etymological ancestors of words do actually add useful information, allowing to transcend cross-lingual boundaries. This method however depends on the availability of etymological thesauri (such as Wikipedia's Wiktionary, or Etymological WordNet), and remains restricted to historically interrelated languages.

In sum, the applicability of the multilingual methods discussed in this section is usually constrained by the availability of external resources. With the aim of overcoming these limitations, we will restrict our investigations to dictionary-free, MT-free multilingual methods.

2.2 Monolingual Classifiers and Multiview Learning

Given the availability of a representative set of labelled documents for each language, a simple baseline, known as the *naïve polylingual classifier*, could be obtained by delegating the classification process to individual monolingual classifiers, each built upon separate monolingual data. Such a solution is sub-optimal, as each classifier does not exploit labelled information from the other languages, a type of information that might provide insights or different perspectives on the semantics of the classes.

García Adeva, Calvo, and López de Ipiña (2005) compared different naïve strategies, considering one single polylingual classifier, i.e., a classifier that works on the juxtaposed representation (1C), vs. various monolingual ones (NC), and one language-independent preprocessor (1P) vs. various language-specific ones (NP), using various learning methods in a bilingual Spanish/Basque benchmark. In their experimentation the combinations NP-NC and NP-1C, which we will consider here as baselines, yielded the best results in terms of running time, memory usage, and accuracy.

Even though training separate language-specific classifiers is a simple way to approach the PLTC task, there are some strategies that could improve the final accuracy by better merging the outcomes of each classifier. Multiview learning (Xu, Tao, & Xu, 2013) for TC deals with parallel texts, i.e., with the case when each document is available in all languages, where each language is considered as a separate source. It was shown by Amini, Usunier, and Goutte (2009) that a multiview majority voting algorithm, which returns the label output by the highest number of language-specific classifiers, outperforms both the naïve polylingual classifier and a multiview Gibbs classifier, which bases its predictions on the mean prediction of each language-specific classifier. Amini and Goutte (2010) proposed a co-regularization approach for multiview text classification which minimizes a joint loss function that takes into account each language-specific classifier loss. However, the availability of a parallel corpus containing all the documents' "views" is a very strong restriction, that is usually alleviated by leveraging machine translation tools that automatically generate the missing documents' views.

2.3 Dimensionality Reduction for Multilingual Classification

One of the main challenges in the "juxtaposed vector space approach" to PLTC concerns the relevant increase in the number of features that represent the documents, i.e., the dimensionality of the vector space (Rigutini et al., 2005). *Feature selection* methods attempt to select a reduced subset of informative features from the original set F so that the size of this subset is much smaller than $|F|$ and so that the reduced set yields high classification effectiveness. In TC the problem is usually tackled via a "filtering" approach, which relies on a mathematical function meant to measure the contribution of each feature to the classification task. Yang and Pedersen (1997) showed that filtering approaches may improve the performance of classification, even for aggressive reduction ratios (e.g., removal of 90% of the features).

Another important dimensionality reduction technique is Latent Semantic Analysis (LSA – aka Latent Semantic Indexing), which originated from the information retrieval community (Deerwester et al., 1990), and has been later applied to cross-lingual classification (Gliozzo & Strapparava, 2006; Xiao & Guo, 2013) and cross-lingual problems in general

(Dumais, Letsche, Littman, & Landauer, 1997). LSA maps the original document-term matrix into a lower dimensional "latent semantic space" that attempts to capture the (linear) relations among the original features and the documents. This mapping is carried out by means of a singular value decomposition (SVD) of the original document-term matrix M. SVD decomposes M as $M = V\Sigma U^T$, where Σ is a diagonal matrix containing all the eigenvalues of M. The approximation $\hat{M}_k = V_k\Sigma_k U_k^T$ of the original matrix M can be computed by taking the k largest eigenvalues of Σ and setting the remaining ones to 0; $\hat{M}_k$ is then said to be "rank-k optimal" in terms of the Frobenius norm. V_k and U_k are orthogonal matrices that "explain" the relations among pairs of terms and pairs of documents, respectively.

Although LSA can successfully be used to discover hidden relations between indirectly correlated features, as is the case for terms belonging to different languages, it suffers from high computational costs. "Random mappings" arise as an alternative to LSA, as they perform comparably in different machine learning tasks by preserving some important characteristics of LSA, and by bringing about, at the same time, significant savings in terms of computational cost (Fradkin & Madigan, 2003). *Random Projections* (RPs – Papadimitriou et al., 1998) and *Random Mappings* (RMs – Kaski, 1998) are two equivalent formulations deriving from the Johnson-Lindenstrauss lemma (Johnson et al., 1986), which states that distances in a Euclidean space are approximately preserved if projected onto a lower-dimensional random space. These formulations are also based on the fundamental result of Hecht-Nielsen (1994), who proved that there are many more nearly orthogonal than truly orthogonal directions in high-dimensional spaces.

RP-like methods can be formalized in terms of the projection of the original document-term matrix M by means of a random matrix Λ, i.e., $\hat{M}_{|D|\times n} = M_{|D|\times|F|} \cdot \Lambda_{|F|\times n}$, where $\Lambda\Lambda^T$ approximates the identity matrix, $|D|$ and $|F|$ indicate the number of documents and terms in the collection, and n stands for the reduced dimensionality, which is typically chosen in advance. The definition of the random-projection matrix Λ is a fundamental aspect of the method; Achlioptas (2001) demonstrated that any random distribution with zero mean and unit variance satisfies the Johnson-Lindenstrauss lemma, and proposed two simple distributions for the definition of the elements $\Lambda_{ij} = \{\lambda_{ij}\}$ of the random projection matrix, by setting the parameter distribution s of Equation 1 to either $s = 2$ or $s = 3$:

$$\lambda_{ij} = \sqrt{s} \times \begin{cases} +1 & \text{with probability } \frac{1}{2s} \\ 0 & \text{with probability } 1 - \frac{1}{s} \\ -1 & \text{with probability } \frac{1}{2s} \end{cases} \quad (1)$$

Achlioptas proved that the configuration in which $s = 3$ can be used to speed up computation, since in this case only 1/3 of the data is non-zero (*sparse random projection*), and therefore 2/3 of the computations can be skipped. Similarly, Li et al. (2006) set $s = \sqrt{|F|}$ and $s = |F|/\log|F|$ (*very sparse random projections*) to significantly speed up the computation while still preserving the inner distances.

Random Indexing (RI), first proposed by Kanerva et al. (2000), is an equivalent formulation of RPs that also accommodates Achlioptas' theory. Sahlgren (2001) defines RI as an approximate alternative to LSA for semantic representation. RI maintains a dictionary of random index vectors for each feature in the original space. Each random index vector consists of an n-dimensional sparse vector with k non-zero values, randomly distributed across $+1$ and -1 (the method is explained in detail in Section 3). In the work of Gorman and

Curran (2006) different weighting criteria for random index vectors in the dictionary were proven useful for improving the matrix representation. RI has been tested in different tasks, such as search (Rangan, 2011), query expansion (Sahlgren, Karlgren, Cöster, & Järvinen, 2002), image and text compression (Bingham & Mannila, 2001), and event detection (Jurgens & Stevens, 2009). Fradkin and Madigan (2003) showed that, since in RI distances are approximately preserved, distance-based learners such as k-Nearest Neighbours (k-NN) and SVMs are preferable when learning from randomly indexed instances. Accordingly, Sahlgren and Cöster (2004) applied RI to (monolingual) text classification using SVMs, and suggested that the random indexing representation (there dubbed *Bag of Concepts* – BoCs in Sahlgren & Cöster, 2004) performed comparably to the BoW representation. The performance of RI has also been tested by Sahlgren and Karlgren (2005) and Gorman and Curran (2006) in the realm of automatic bilingual lexicon acquisition.

The above-discussed works indicate that RI is a promising dimensionality reduction technique for representing polylingual data. Our proposal is inspired by the works of Achlioptas (2001) and Li et al. (2006) on sparse projections by taking the level of sparsity to the extreme, and extends the application of RI in TC (Sahlgren & Cöster, 2004) to PLTC, which, to the best of our knowledge, has never been done so far. In the following section we will first describe the method in detail, and then propose a particular setting aimed at overcoming certain obstacles that could arise in the polylingual setting.

3. Lightweight Random Indexing for Polylingual Text Classification

Text Classification (TC) can be formalized as the task of approximating an unknown *target function* $\Phi : \mathcal{D} \times \mathcal{C} \rightarrow \{-1, +1\}$, that indicates how documents ought to be classified, by means of a function $\hat{\Phi} : \mathcal{D} \times \mathcal{C} \rightarrow \{-1, +1\}$, called *the classifier*, such that Φ and $\hat{\Phi}$ coincide as much as possible in terms of a given evaluation metric. Here $\mathcal{D}$ denotes the domain of documents, $\mathcal{C} = \{c_1, c_2, ..., c_{|\mathcal{C}|}\}$ is a set of predefined *classes*, while values $+1$ and -1 indicate membership and non-membership of the document in the class, respectively. We will here consider "multilabel" classification, that is, the setting in which each document could belong to zero, one, or several classes at the same time; we will consider the "flat" version of the problem, in which no hierarchical relations among classes exist. We adopt the *1 vs. all* strategy, according to which the multilabel classification problem is solved as $|\mathcal{C}|$ independent binary classification problems.

A document collection D can be represented via a matrix $M_{|D|\times|F|}$

$$M = \begin{pmatrix} \vec{d_1} \\ \vec{d_2} \\ \vdots \\ \vec{d}_{|D|} \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1|F|} \\ w_{21} & w_{22} & \cdots & w_{2|F|} \\ \vdots & \vdots & \ddots & \vdots \\ w_{|D|1} & w_{|D|2} & \cdots & w_{|D||F|} \end{pmatrix} \tag{2}$$

where $|D|$ and $|F|$ are the number of documents and features in the collection, and real values w_{ij} represent the weight of feature f_j in document d_i, which is usually determined as a function of the frequency of the feature in the document and in the collection.

Polylingual Text Classification adds one fundamental aspect to TC, i.e., different documents may belong to different languages. Let $\Psi : \mathcal{D} \rightarrow L$ return the language in which

a given document is written, where $L = \{l_1, l_2, \ldots, l_{|L|}\}$ is the pool of languages, $|L| > 1$. Let $F = \bigcup_{i=1}^{|L|} F_i$ denote the vocabulary of the collection, that can be expressed as the union of the language-specific vocabularies F_i. The polylingual setting assumes that the distribution $P(\Psi(d) = l_i)$ across the training set is approximately uniform, that is, there is a representative quantity of labelled documents for each language.

There is usually only a small amount of shared features across languages (e.g., proper nouns)[1], and this implies that $\langle \vec{d'}, \vec{d''} \rangle \approx 0$ if $\Psi(d') \neq \Psi(d'')$, where $\langle \cdot, \cdot \rangle$ denotes the dot product. (Incidentally, this means that a direct similarity comparison among documents expressed in different languages, e.g., using cosine-similarity, would be doomed to fail.) It is thus possible, for any language l_i, to perform a reordering of the rows and columns in the matrix that allows the polylingual matrix M to be expressed as $M = \begin{bmatrix} M_1 & M_2 & 0 \\ 0 & M_3 & M_4 \end{bmatrix}$, where $[M_1; M_2]$ is the $|\{d \in D : \Psi(d) = l_i\}| \times |F_i|$ monolingual matrix representation for language l_i, $\begin{bmatrix} M_2 \\ M_3 \end{bmatrix}$ is a $|D| \times \alpha$ matrix containing all the α words that are shared across two or more languages, and 0 denotes all-zero matrices.

3.1 Random Indexing

Random Indexing maps each observable problem feature into a random vector in a vector space in which the number of dimensions is not determined by the number of different unique features we want to map, but is instead fixed in advance. Originally, RI was proposed for performing semantic comparisons between terms. Each document was thus mapped into a random index vector that was then accumulated (via vector addition) into the term's row of a term-document matrix each time the term occurred in that document. In our case, we are instead interested in performing semantic comparisons between documents, not terms. Thus, each term f_i is assigned an n-dimensional random index vector, that is accumulated into the j-th row of a document-term matrix every time the term is found in document d_j.

Random index vectors are nearly-orthogonal, and comply with the conditions spelled out by Achlioptas (2001) (see Section 2.3), i.e., zero-mean distribution with unit variance, so as to satisfy the Johnson-Lindenstrauss lemma. A random index vector is created by randomly setting $k \ll n$ non-zero values, equally distributed between $+1$ and -1, in an n-dimensional vector where n is typically on the order of the thousands. Once n is fixed, a recommended choice of k in the literature is $k = n/100$. We dub this configuration $\text{RI}_{1\%}$, and will use it in our comparative experiments. As vectors in $\text{RI}_{1\%}$ are sparse, using sparse data structure representations could bring about memory savings. The $\hat{M}_{|D|\times n} = M_{|D|\times|F|} \cdot \Lambda_{|F|\times n}$ matrix multiplication (see Section 2.3) can be completely skipped, building $\hat{M}_{|D|\times n}$ "on-the-fly" by scanning each document and accumulating the corresponding random index vectors as each term is read. This also avoids the need to allocate the entire matrix $M_{|D|\times|F|}$ in memory.

According to Sahlgren (2005), the main advantages of RI can be summarized as follows: the method (i) is incremental, and provides intermediate results before all the data are read

1. Note that other formulations of the polylingual problem, e.g., the ones by Amini et al. (2009) and Prettenhofer and Stein (2010), do actually impose that if $i \neq j$ then $F_i \cap F_j = \phi$. This means that shared words across languages, such as proper nouns, are given multiple representations as language-specific features.

in; (ii) avoids the so-called "huge matrix step" (i.e., allocating the entire $M_{|D|\times|F|}$ matrix in memory), and (iii) is scalable, since adding new elements to the data does not increase the dimensionality of the space (e.g., new features are represented via a new random index, and not via a new dimension).

BoW matrices are typically weighted and normalized to better represent the importance of the word to each document and to avoid giving long documents more *a priori* importance, respectively. Weighting schemes could also be incorporated into the RI formalism in a simple manner; e.g., each time a random index is added to a document row, it can first be multiplied by the weight of that term in that document. That this brings about improved accuracy was shown by Gorman and Curran (2006); however, in the same work it was also shown that the incremental nature of the algorithm is sacrificed if non-linear weights are taken into account. In our experiments, as the weighting criterion we use the well-known *tfidf* method, expressed as

$$tfidf(d_i, f_j) = tf(d_i, f_j) \times \log \frac{|D|}{|d \in D : tf(d, f_j) > 0|} \tag{3}$$

where $tf(d_i, f_j)$ counts the number of occurrences of feature f_j in document d_i; weights are then normalized via cosine normalization, as

$$w_{ij} = \frac{tfidf(d_i, f_j)}{\sqrt{\sum_{f_k \in F} tfidf(d_i, f_k)^2}} \tag{4}$$

3.2 Lightweight Random Indexing

During preliminary experiments on the application of RI as a method for dimensionality reduction, we observed that SVMs required more time to train when the training set had been processed with RI, than with the original high-dimensional vector space (see Section 5.2). We also observed a correlation between training times and the choice of k, while the choice of n had a smaller impact on efficiency.

Optimizing the choice of k in RI can be though of as a means to achieve two main goals: (i) being able to encode a large number of different features in a reduced space, and (ii) increasing the chance that two random index vectors are orthogonal.

With respect to (i), it is easy to show that, if we want to assign a different n-dimensional index vector with k non-zero values to each original feature, RI could encode a maximum of $C(n,k) = \binom{n}{k}2^k$ features (*representation capacity*). $C(n,k)$ grows rapidly as a function of either n or k; just as an example, $C(5000, 50) \approx 2.5 \cdot 10^{135}$. Such a huge capacity clearly exceeds the representation requirements imposed by any current or future dataset. However, even with small values of k the capacity becomes large enough to encode any reasonable dataset, e.g., $C(5000,2)$=49,990,000 distinct features.

With respect to (ii), random-projection-based algorithms rely on the Hecht-Nielsen (1994) lemma to find nearly orthogonal directions in a reduced space. Two vectors $\vec{u}$ and $\vec{v}$ in an inner product space are said to be *orthogonal* whenever $\langle \vec{u}, \vec{v} \rangle = 0$, where $\langle \vec{u}, \vec{v} \rangle = \sum_i u_i v_i$ is the dot product. Random indexes are chosen so as to be sparse in order to increase the probability that the dot product equals zero, with non-zero products evenly distributed between $+1$ and -1, leaving the expected value of the outcome close to zero. By

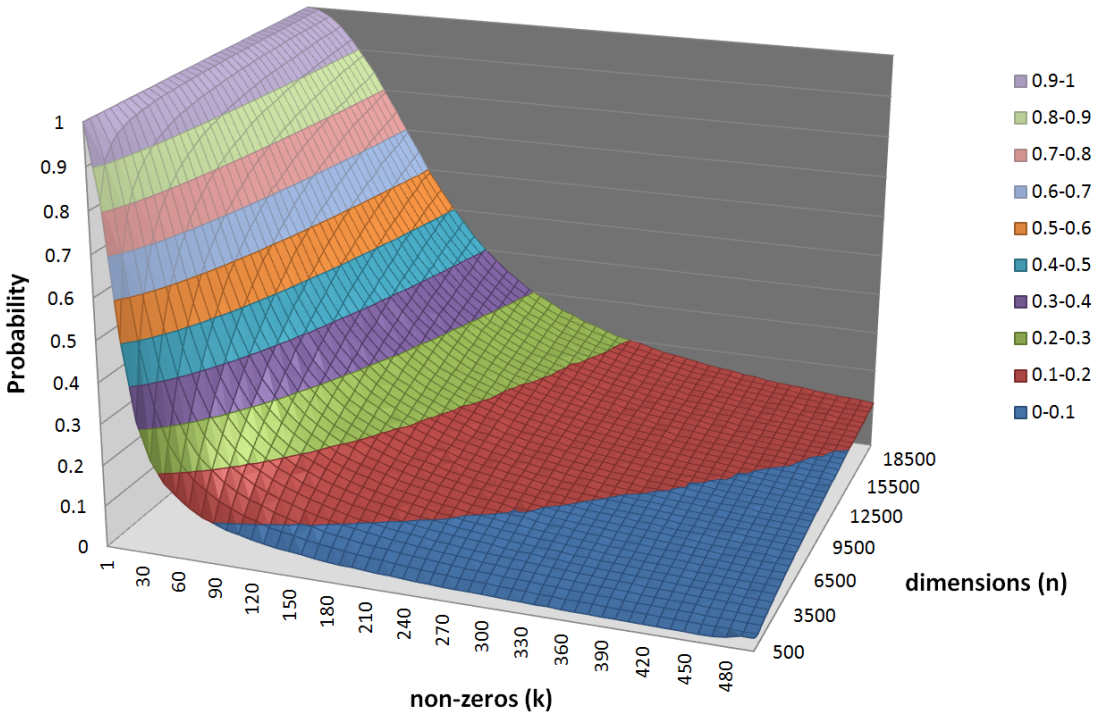


Figure 1: Probability of orthogonality of two random index vectors as a function of k and n.

means of a Monte Carlo algorithm, we estimated the probability of orthogonality between any two randomly generated vectors for a grid of sample values for n and k. The results, plotted in Figure 1, reveal that smaller values of k are the main factor in favouring the orthogonality of two random index vectors, while n has a smaller impact.

If many random index vectors lack orthogonality, the information conveyed by the original distinct features, which are predominantly pair-wise semantically unrelated, gets mixed up, causing the learner to have more difficulty in learning meaningful separation patterns from them. The orthogonality of random index vectors plays an even more important role for features that are shared across languages. As shown in work by Gliozzo and Strapparava (2005), these shared words play a relevant role in bringing useful information across languages. If their corresponding random index vectors are orthogonal with respect to all the other vectors, the information they contribute to the process is maximized, instead of being diluted by other less informative features.

Following the observations above, we propose the use of Random Indexing with a fixed $k = 2$; we dub this configuration *Lightweight Random Indexing* (LRI). Our hypothesis is that this setting could be advantageous as a mechanism to reduce dimensionality (so as to mitigate the problem of feature disjointness in PLTC), since it is sufficient in order to represent large feature vocabularies while also preserving vector orthogonality. Note that choosing $k = 1$, when $n = |F|$, would be equivalent to performing a random permutation of

Output: Dictionary;
// Generate a random index vector for each feature
1 **for** $i = 0$ *to* $(|F| - 1)$ **do**
  // We choose the 1st dimension sequentially
2 $dim_1 \leftarrow (i \textbf{ mod } n) + 1$;
  // We choose the 2nd dimension uniformly at random
  // from the dimensions not chosen in Line 2
3 $dim_2 \leftarrow rand(\{1, ..., n)\} \setminus \{dim_1\})$;
  // We assign the 1st non-zero value uniformly at random
4 $val_1 \leftarrow rand(\{\frac{+1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\})$;
  // Same for the 2nd non-zero value
5 $val_2 \leftarrow rand(\{\frac{+1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\})$;
  // We create the sparse random index vector
6 random_index_vector $\leftarrow [(dim_1, val_1), (dim_2, val_2)]$;
  // We build the feature-vector mapping
7 Dictionary.map(f_{i+1}, random_index_vector) ;
8 **end**

Algorithm 1: Feature Dictionary for Lightweight Random Indexing.

feature indexes in a BoW representation; $k = 2$ is the minimum value for which an actual RI is performed.

Algorithm 1 formalizes the process of creating a dictionary, that is, of creating a mapping consisting of one random vector for each original feature; the mapping is created at training time and is then used for classifying the unlabelled documents (this means that, in Line 1, F is the set of features present in the training set). The value $1/\sqrt{2}$ is used instead of 1 in order to obtain vectors of length one. Note that the two dimensions are selected in a different manner, with the step at Line 2 ensuring that all latent dimensions are used approximately the same number of times, and the step at Line 3 ensuring that the dimension chosen in the previous step is not chosen twice.

Our proposal presents the following advantages with respect to standard $\text{RI}_{1\%}$ and, in general, with respect to any RI with $k > 2$:

- Each index vector has only two non-zero values. The mapping can be allocated in memory for any number of original features, and the projection is performed very quickly;
- Given a fixed value of n, it has a higher probability than any other instantiation of RI of generating truly pairwise orthogonal random vectors;
- Parameter k becomes a constant that needs no tuning.

4. Experiments

In this section we experimentally compare our Lightweight Random Indexing (LRI) method to other representation approaches proposed in the literature.

4.1 Baselines and Implementation Details

As the baselines against which to compare LRI we have chosen the following methods, that we group in three categories according to their common characteristics:

Orthogonal Mappings: methods using a canonical basis for the co-occurrence matrix:

- **PolyBow:** a classifier that operates on the juxtaposed BoW representation (PolyBow corresponds to the NP-1C setup in García Adeva et al., 2005).
- **FS:** Feature Selection on PolyBoW using Information Gain as the term scoring function and Round Robin (Forman, 2004) as the term selection policy.
- **Majority Voting:** a multiview voting algorithm that returns the label output by the highest number of language-specific classifiers (Amini et al., 2009).
- **MonoBoW:** a lower bound baseline that uses a set of naïve monolingual classifiers (MonoBoW corresponds to the NP-NC setup in García Adeva et al., 2005).
- **MT:** an upper bound baseline based on statistical machine translation, which translates all non-English training and test documents into English.

Random Mappings: dimensionality reduction methods relying on random projections:

- **RI$_{1\%}$:** Random Indexing with $k = n/100$ (Sahlgren & Cöster, 2004).
- **ACH:** Achlioptas mapping with ternary distribution obtained by setting $s = 3$ in Equation 1 (Achlioptas, 2001).

Non-Random Mappings: dimensionality reduction methods relying on mappings which are not random:

- **CL-LSA:** Cross-Lingual Latent Semantic Analysis (Dumais et al., 1997).
- **MDM:** Multilingual Domain Models (Gliozzo & Strapparava, 2005).
- **PLDA:** Polylingual Latent Dirichlet Allocation (Mimno et al., 2009).

We will here assume language labels are available in advance[2] for both training and testing documents. Note that RI methods and PolyBoW represent all documents in the same feature space, irrespective of their language label. Conversely, MonoBoW keeps a separate language-specific classifier for each language; the class label for a test document is then decided by the classifier associated to the document's language label. We test PLDA and Majority Voting only on the JRC-Acquis parallel corpus, since for all documents they require a separate view in all languages to be available. Majority Voting maintains a separate classifier for each distinct language (5 in our experiments); each test document is thus classified after using 5 classification decisions in voting, one for each language-specific

2. This assumption is fair, as current language identification models deliver accuracies very close to 100%

view. For singular value decomposition we have used the Rohde (2011) package. We have used the Haddow, Hoang, Bertoldi, Bojar, and Heafield (2016) implementation to generate a set of statistical translation systems trained on the sentence-aligned parallel data provided by the Europarl data release (Koehn, 2005). Note that, since we used the method described by Gliozzo and Strapparava (2005) to automatically obtain the bilingual model in MDM, MT is the only method using external knowledge. For PLDA we have used the Richardson (2008) implementation, which uses Gibbs sampling; we adhere to the common practice of fixing the budget of iterations to 1,000. We have implemented the LRI method and the other baseline methods as part of the Esuli, Fagni, and Moreo (2016) framework. We have used Support Vector Machines (SVMs) as the learning device in all cases, since it has consistently delivered state-of-the-art results in TC so far; for it we used the well-known Joachims (2009) implementation of Joachims (2005), with default parameters.

4.2 Evaluation Measures

As the effectiveness measure we use the well-known F_1, the harmonic mean of precision (π) and recall (ρ) defined as $F_1 = (2\pi\rho)/(\pi + \rho) = (TP)/(2TP + FP + FN)$ where TP, FP, and FN stand for the numbers of *true positives*, *false positives*, and *false negatives*, respectively. We take $F_1 = 1$ when $TP = FP = FN = 0$, since the classifier has correctly classified all examples as negative.

We compute both micro-averaged F_1 (denoted by F_1^μ) and macro-averaged F_1 (denoted by F_1^M). F_1^μ is obtained by (i) computing the class-specific values TP_r, FP_r, and FN_r, (ii) obtaining TP as the summation of the TP_r's (same for FP and FN), and then applying the F_1 formula. F_1^M is obtained by first computing the class-specific F_1 values and then averaging them across all classes. The fact that F_1^M attributes equal importance to all classes means that low-frequency classes will be as important as high-frequency ones in determining F_1^M scores; F_1^μ is instead more influenced by high-frequency classes than by low-frequency ones. High values of F_1^M thus tend to indicate that the classifier performs well also on low-prevalence classes, while high values of F_1^μ may just indicate that the classifier performs well on high-prevalence classes.

4.3 Datasets

We have performed our experiments on two publicly available corpora, RCV1/RCV2 (a comparable corpus) and JRC-Acquis (a parallel corpus).

4.3.1 RCV1/RCV2

RCV1 is a publicly available collection consisting of the 804,414 English news stories generated by Reuters from 20 Aug 1996 to 19 Aug 1997 (Lewis, Yang, Rose, & Li, 2004). RCV2 is instead a polylingual collection, containing over 487,000 news stories generated in the same timeframe in thirteen languages other than English (Dutch, French, German, Chinese, Japanese, Russian, Portuguese, Spanish, LatinoAmerican Spanish, Italian, Danish, Norwegian, Swedish). The union of RCV1 and RCV2 (hereafter referred to as RCV1/RCV2) is a corpus *comparable* at topic-level, as news stories are not direct translations of each other are but simply refer to the same or to related events in different languages. Since the cor-

pus is not parallel, each training document for a given language in general does not have a counterpart in the other languages.

From RCV1/RCV2 we randomly selected 8,000 news stories for 5 languages (English, Italian, Spanish, French, German) pertaining to the last 4 months (from 1997-04-19 to 1997-08-19), and we performed a 70%/30% train/test split, thus obtaining a training set of 28,000 documents (5,600 for each language) and a test set of 12,000 documents (2,400 for each language)[3]. In our experiments we have restricted our attention to the 67 classes (out of 103) with at least one positive training example for each of the five languages. The average number of classes per document is 2.92, ranging from a minimum of 1 to a maximum of 11; the number of positive examples per class/language combination ranges from a minimum of 1 to a maximum of 4,182.

We preprocessed the corpus by removing stop words and by stemming terms using the Porter stemmer for English, and the Snowball stemmer for the other languages. This resulted in a total of 123,258 stemmed terms, distributed across languages as shown in Table 1.

	English	Italian	Spanish	French	German	Appearing in	#
English	40,483	3,420	6,559	6,370	3,921	1 languages	106,182
Italian		14,762	3,752	3,300	1,929	2 languages	10,474
Spanish			30,077	6,139	3,014	3 languages	3,851
French				26,961	3,441	4 languages	1,923
German					38,232	5 languages	828

Table 1: Feature distribution across languages for the RCV1/RCV2 comparable corpus. In the leftmost part of the table, the cell in row i and column j represents the number of features that are shared across the i-specific and the j-specific sections of the dataset. (The table is symmetric, so for better clarity the entries below the diagonal have been omitted.) The rightmost part of the table indicates how many features are shared across x language-specific sections of the dataset.

4.3.2 JRC-Acquis

The JRC-Acquis corpus (version 3.0) is a version of the Acquis Communautaire collection of parallel legislative texts from European Union law written between the 1950s and 2006 (Steinberger, Pouliquen, Widiger, Ignat, Erjavec, Tufis, & Varga, 2006). JRC-Acquis is publicly available for research purposes, and covers 22 official European languages. The corpus is parallel at the sentence-level, i.e., each document exists in all 22 languages, as a sentence-by-sentence translation. The corpus is labelled according to the ontology-based EuroVoc thesaurus, which consists of more than 6,000 classes; for our experiments we have restricted our attention to the 21 classes in the top level of the EuroVoc hierarchy.

3. All the information required to replicate the experiments, e.g., IDs of the selected documents, assigned labels, etc., is publicly available (Moreo, 2016). The source code we used in our experiments is accessible as part of the Esuli et al. (2016) framework

	English	Italian	Spanish	French	German	Appearing in	#
English	150,866	77,878	80,220	89,573	98,740	1 languages	249,216
Italian		150,838	95,515	90,522	78,919	2 languages	42,566
Spanish			143,712	88,561	85,434	3 languages	33,305
French				147,077	86,905	4 languages	22,171
German					228,834	5 languages	59,676

Table 2: Feature distribution across languages for the JRC-Acquis parallel corpus; the meaning of the cells is the same as in Table 1. Note the high number of features (59,676) which appear in all five languages; this is due to the presence of proper names, which are the same in all languages. Note also the high number of features (228,834) which are unique to the German language: this is due to the presence of word compounds, a phenomenon present in the German language but not in the other four languages.

We have selected the 7,235 texts from 2006 for 5 languages (English, Italian, Spanish, French, and German) and removed documents without labels, thus obtaining 6,980 documents per language. We have taken the first 70% documents for training (24,430, i.e., 4,886 for each language) and the remaining 30% (10,470, i.e., 2,094 for each language) for testing. The average number of classes per document is 3.5, ranging from a minimum of 1 to a maximum of 10; the number of positive examples per class/language combination ranges from a minimum of 47 to a maximum of 2,011.

The same preprocessing as for RCV1/RCV2 was carried out on this dataset, obtaining 406,934 distinct features distributed across languages as shown in Table 2. Since the JRC-Acquis corpus is parallel, each language-specific document is guaranteed to have a counterpart in each of the other languages, which results in a relatively large number of terms (e.g., proper nouns) appearing in several languages. Note that, despite the fact that the dataset is parallel at the sentence level, we are interested in indexing entire documents as a whole, and thus disregard sentence order; we thus consider the corpus as parallel at the document level.

We use the JRC-Acquis corpus in order to test the performance of LRI in cases in which the co-occurrence matrix has been *compacted*, as defined in the work of Dumais et al. (1997). More precisely, the *compact representation* of $|L|$ translation-equivalent documents is a vector consisting of the concatenation of the $|L|$ vectors that each represent one (monolingual) such document. This is different from the juxtaposed representation used in the previous chapters, where the vector corresponding to one monolingual document has all zeros in the positions corresponding to the features of the other languages. The "compact matrix" can thus be obtained from the matrix resulting from the juxtaposed representations by compressing $|L|$ rows into a single (compact) row storing their sum.

4.4 Results

In this section we present the results of our experiments. We first compare LRI to a set of monolingual classifiers (Section 4.4.1), and then we explore the dimensionality reduction aspect of the polylingual problem (Section 4.4.2).

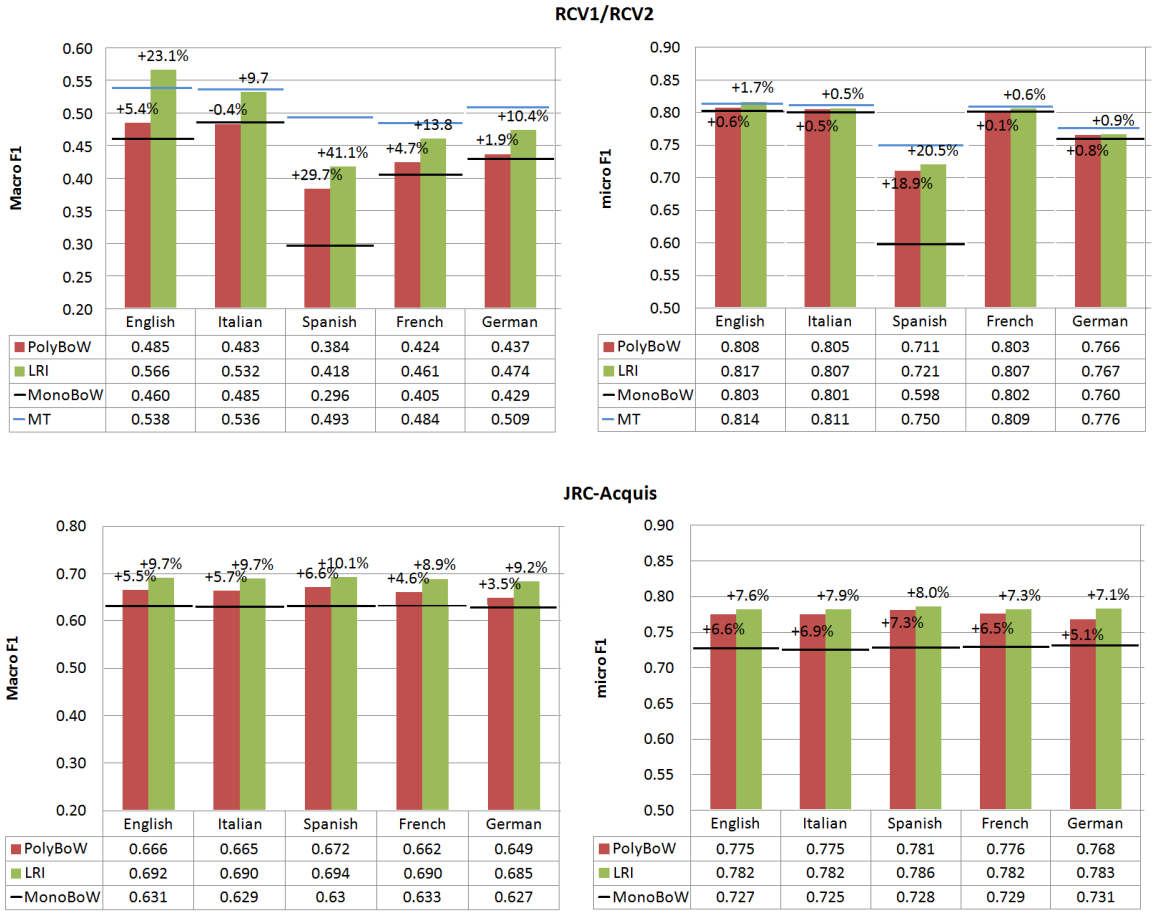
4.4.1 Polylingual Information

As a first case of study, we investigate how much the addition of polylingual information affects the accuracy of a monolingual classifier. In this scenario, we compare LRI and PolyBoW, which train on documents from all languages, with the lower bound MonoBoW, which trains only on documents of the same language of test documents, and with the upper bound MT, that first translates all training and test documents into English. Note that the MT baseline is not tested in the JRC-Acquis corpus because each of the documents is already available as a direct translation in all languages. In this experiment the vector space is not being reduced, i.e., we set $n = |F|$ for LRI so that the vector spaces for PolyBoW and LRI have the same number of dimensions. Values for LRI were averaged after 10 runs.

The results illustrated in Figure 2 show that the simple addition of examples in different languages (PolyBoW) brings about an improvement in accuracy with respect to the monolingual solution (MonoBoW). This improvement is likely achieved thanks to the words shared across languages. However, LRI clearly outperforms PolyBoW. The improvements of PolyBoW over MonoBoW range from -0.4% to +29.7%, while LRI achieves improvements ranging from +9.7% to +41.1%; when LRI obtains its smallest improvement over MonoBoW in terms of F_1^M (on Italian, +9.7%), PolyBoW performs slightly worse than MonoBoW (-0.4%). The improvements are more marked for F_1^M than for F_1^μ, indicating that the improvements especially take place in the more infrequent classes, which have a substantial impact on F_1^M but not on F_1^μ.

In general, training on documents coming from all languages (PolyBoW, LRI, and MT) seems to be preferable to training from language-specific documents only (MonoBoW). This is particularly so for the MT baseline, which obtained the best results in all cases with the sole exception of English, where LRI obtained the best result. This exception might be explained by the fact that automatically translated documents tend to exhibit different statistical properties with respect to documents written by humans, which means that the English test documents (which are not translations) might not be in tune with the training documents (which are mostly the result of automatic translation).

The language-specific classification performance is much more homogeneous in JRC-Acquis than in RCV1/RCV2. This can be explained by the fact that JRC-Acquis is a parallel corpus, and therefore each language benefits from the very same information. There is no significant difference in performance among the different languages, which means that the effects due to the different difficulty of the various languages are minor. Instead, differences in RCV1/RCV2 can be explained by the different amount of information that the training sets carry on the corresponding test sets. For example, the Spanish classifier is the worst performer, and is the one that obtains the best benefit (with respect to the MonoBoW baseline) from the addition of polylingual information (as in PolyBoW, LRI, and MT).



RCV1/RCV2 — Macro F1

	English	Italian	Spanish	French	German
PolyBoW	0.485	0.483	0.384	0.424	0.437
LRI	0.566	0.532	0.418	0.461	0.474
MonoBoW	0.460	0.485	0.296	0.405	0.429
MT	0.538	0.536	0.493	0.484	0.509

RCV1/RCV2 — micro F1

	English	Italian	Spanish	French	German
PolyBoW	0.808	0.805	0.711	0.803	0.766
LRI	0.817	0.807	0.721	0.807	0.767
MonoBoW	0.803	0.801	0.598	0.802	0.760
MT	0.814	0.811	0.750	0.809	0.776

JRC-Acquis — Macro F1

	English	Italian	Spanish	French	German
PolyBoW	0.666	0.665	0.672	0.662	0.649
LRI	0.692	0.690	0.694	0.690	0.685
MonoBoW	0.631	0.629	0.63	0.633	0.627

JRC-Acquis — micro F1

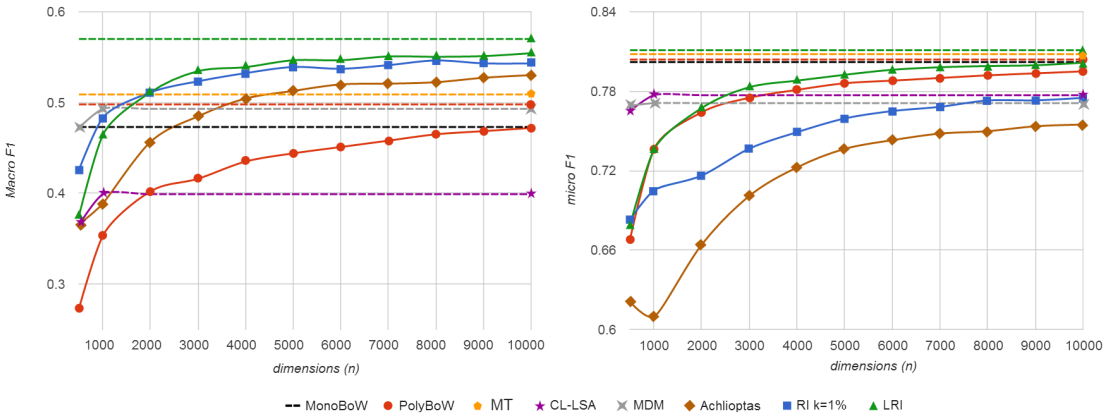
	English	Italian	Spanish	French	German
PolyBoW	0.775	0.775	0.781	0.776	0.768
LRI	0.782	0.782	0.786	0.782	0.783
MonoBoW	0.727	0.725	0.728	0.729	0.731

Figure 2: Monolingual classification on RCV1/RCV2 (top) and JRC-Acquis (bottom), using F_1^M (left) and F_1^μ (right) as the evaluation measure.

Note that in this experiment the matrices that PolyBoW and LRI feed to the learning algorithm are of the same size. The difference between the two methods, which is the likely cause for their difference in effectiveness, is that in PolyBoW the useful dimensions for a specific language are "packed" in a specific portion of the vector space, while LRI spreads them across the entire vector space, causing all dimensions to become potentially useful for all languages. Note that this substantial increase in the number of useful dimensions available for each language allows the model to create more easily separable representations. We further discuss this aspect in Section 5.3.

4.4.2 Dimensionality Reduction

In the PolyBoW setup the dimensionality of the vector space is substantially increased when more languages are considered during training. The following experiments explore the dimensionality reduction aspect of the problem, and address a realistic polylingual scenario, where both training and test data contain examples for each language.

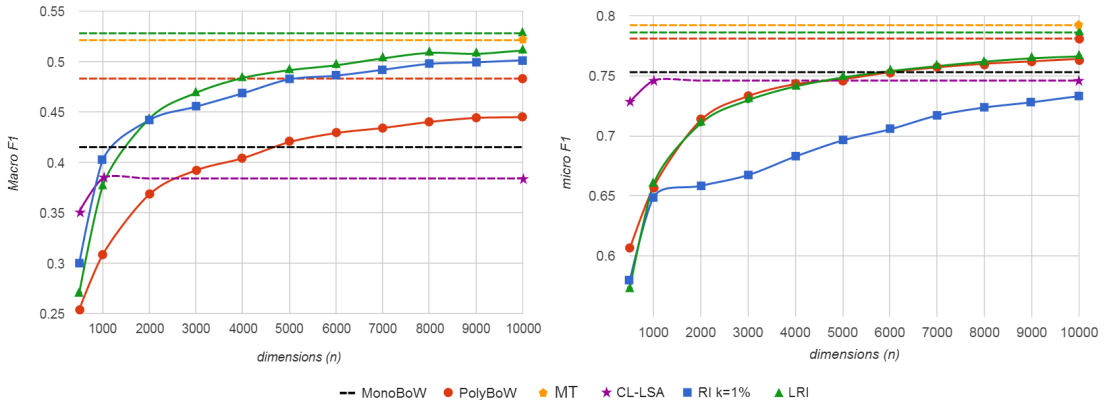


	F_1^M					F_1^μ				
n	**500**	**1,000**	**5,000**	**10,000**	**full**	**500**	**1,000**	**5,000**	**10,000**	**full**
MonoBoW	-	-	-	-	0.473	-	-	-	-	0.802
PolyBoW	0.273	0.353	0.444	0.472	0.498	0.668	0.736	0.786	0.795	0.804
MT	-	-	-	-	0.509	-	-	-	-	0.808
CL-LSA	0.365	0.399	-	-	-	0.765	**0.777**	-	-	-
MDM	**0.472**	**0.493**	-	-	-	**0.769**	0.771	-	-	-
ACH	0.366	0.389	0.513	0.530	-	0.621	0.610	0.736	0.755	-
RI$_{1\%}$	0.426	0.483	0.539	0.543	-	0.683	0.705	0.756	0.775	-
LRI	0.375	0.464	**0.547**	**0.554**	**0.570**	0.679	0.736	**0.792**	**0.802**	**0.811**

Figure 3: Effects of dimensionality reduction on RCV1/RCV2 (English and Italian). Dotted lines indicate reference values, e.g., green and red lines represent the performance of LRI and PolyBoW, respectively, when dimensionality is not reduced. Values in bold highlights the best performing method for each dimension.

We first run a sample bilingual experiment on RCV1/RCV2 (as the language other than English we have picked Italian). The total amount of features in this dataset is 51,828. Restricting the experiment to two languages allows us to compare LRI (i) against methods that were proposed for bilingual representations (MDM), and (ii) against methods that would be too computationally expensive if considering more languages (such as ACH, see below). We explore the effect of dimensionality reduction, with the number of selected features ranging from 500 to 10,000 (Figure 3). We adhere to the common practice that establishes a number of dimensions ranging from 500 to 1000 in LSA and MDM. Results for random projection methods (ACH, RI$_{1\%}$, and LRI) are averaged after 10 runs.

LRI obtains good results on both macro- and micro-averaged F_1, while the other methods exhibit alternating performance on the two measures. RI$_{1\%}$ obtains comparable results in terms of F_1^M but performs poorly on F_1^μ; in contrast, PolyBoW performs comparably in terms of F_1^μ but worse in terms of F_1^M. A two-tailed t-test on paired examples reveals that the difference in terms of F_1^M between LRI and RI$_{1\%}$ is not statistically significant,



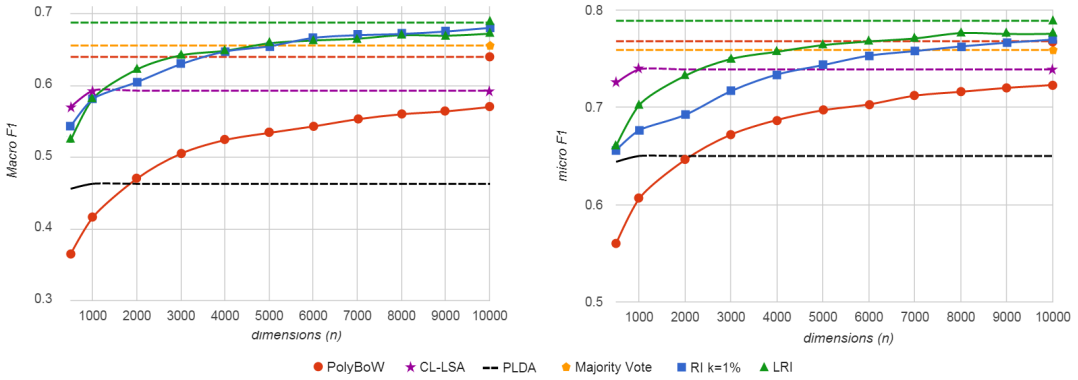
	F_1^M					F_1^μ				
n	**500**	**1,000**	**5,000**	**10,000**	**full**	**500**	**1,000**	**5,000**	**10,000**	**full**
MonoBoW	-	-	-	-	0.415	-	-	-	-	0.753
PolyBoW	0.254	0.308	0.420	0.445	0.483	0.606	0.657	0.747	0.764	0.781
MT	-	-	-	-	0.521	-	-	-	-	**0.793**
CL-LSA	**0.351**	0.384	-	-	-	**0.728**	**0.746**	-	-	-
RI$_{1\%}$	0.300	**0.402**	0.482	0.501	-	0.580	0.649	0.696	0.733	-
LRI	0.270	0.376	**0.491**	**0.511**	**0.528**	0.573	0.659	**0.749**	**0.766**	0.786

Figure 4: Accuracy of different PLTC methods on RCV1/RCV2 on 5 languages, for different levels of dimensionality reduction.

and that LRI significantly outperforms RI$_{1\%}$ in F_1^μ and the rest of dimensionality reduction methods for both evaluation measures, with $p < 0.001$. Surprisingly, CL-LSA and MDM perform worse than the naïve classifier (MonoBoW) with all features. However, it should be remarked that they outperform all other baselines with only 500 and 1000 dimensions. As will be seen in Section 5, apart from the drastic dimensionality reduction, these methods are affected by large computational costs that negatively impact on the run times and memory resources needed. Consistently with our previous observations (see Figure 2), LRI, PolyBoW, MonoBoW, and MT are comparable in terms of F_1^μ, but LRI outperforms all tested algorithms in terms of F_1^M.

To test the scalability of our method when several languages are involved, we extend the experiment to five languages (English, Italian, Spanish, French, German) in RCV1/RCV2 (Figure 4). Note that in this case not all algorithms were able to complete their execution due to memory constraints, hence the incomplete plots and table; concretely, ACH and the last iterations for RI$_{1\%}$ overflowed memory resources when trying to allocate a $28,000 \times 123,258$ matrix. More insights about space and time complexity are reported in Section 5. Results for RI$_{1\%}$ and LRI are the average of 10 runs that use different random seeds.

These results confirm the previous observations. RI$_{1\%}$ behaves similarly to LRI in terms of F_1^M (i.e., with no statistically significant difference) but worse in terms of F_1^μ (p <0.001),



	F_1^M					F_1^μ				
n	**500**	**1,000**	**5,000**	**10,000**	**full**	**500**	**1,000**	**5,000**	**10,000**	**full**
PolyBoW	0.365	0.416	0.534	0.570	0.640	0.560	0.606	0.697	0.723	0.768
CL-LSA	**0.570**	**0.593**	-	-	-	**0.725**	**0.739**	-	-	-
PLDA	0.456	0.463	-	-	-	0.644	0.650	-	-	-
Majority Vote	-	-	-	-	0.656	-	-	-	-	0.759
RI$_{1\%}$	0.543	0.581	0.655	**0.680**	-	0.656	0.676	0.743	0.770	-
LRI	0.524	0.581	**0.659**	0.672	**0.688**	0.660	0.702	**0.764**	**0.776**	**0.789**

Figure 5: Accuracy of different PLTC methods on JRC-Acquis on 5 languages, for different levels of dimensionality reduction.

while PolyBoW behaves in an opposite way, i.e., performs worse than LRI in terms of F_1^M ($p < 0.001$) and comparably in terms of F_1^μ. As a dimensionality reduction method, LRI thus outperforms the other methods when considering both F_1^M and F_1^μ; when no dimensionality reduction is applied, only the upper bound MT is comparable to LRI in both F_1^M and F_1^μ.

Finally, we used JRC-Acquis to reproduce one last polylingual scenario, namely, one in which texts are aligned at the document level. Even if this situation is not common in practice (exceptions include, say, proceedings of official events), this scenario is interesting since such a dataset may serve as a test bed for multiview learning methods (Amini et al., 2009). Since documents in JRC-Acquis were translated by humans, results are not affected by any noise MT tools might introduce. Figure 5 shows the results obtained considering the compacted matrix of JRC-Acquis (a $4,886 \times 406,934$ matrix), on which we also tested Majority Voting, which combines the classification decisions of the five independently trained MonoBoW classifiers on the parallel versions of the documents, and PLDA, that first defines the generative model based on polylingual topics and then trains and tests on the probability distributions over topics assigned to each document. We set the number of polylingual latent topics to 500 and 1000, respectively.

LRI is clearly superior to PolyBoW in this case. The difference in performance between LRI and RI$_{1\%}$ seems to be lower in this case, especially in terms of F_1^M; the t-test revealed

however that LRI is superior to $\text{RI}_{1\%}$ in a statistically significant sense ($p < 0.001$). However, it should be considered that LRI delivers its best performance without reducing the dimensionality of the polylingual matrix, while $\text{RI}_{1\%}$ is not able to accomplish the projection due to memory restrictions; this is something we will expand on in the following section. PLDA, in turn, succeeded in discovering polylingual topics that were aligned across languages, but proved less effective in terms of classification performance.

5. Analysis

During our experiments we observed substantial differences in terms of efficiency among some of the compared methods, particularly ACH, $\text{RI}_{1\%}$, and LRI. For example, $\text{RI}_{1\%}$ exhausted memory resources for $n \geq 10,000$, while LRI was able to represent even the full-sized $|D| \times |F|$ matrix (see Figure 4). Given the strong relationship between the two methods, we would have expected they delivered similar performance. This anomaly prompted us to investigate the issue more in depth. This section presents an analytical study in terms of efficiency of the methods discussed in the previous section.

5.1 Space Efficiency

Data samples in ML are usually represented as a co-occurrence matrix. In TC this matrix suffers from high-dimensionality, but luckily enough it is also sparse. A sparse, low-density matrix suggests the use of a non-exhaustive data-structure, in which zero values are not stored explicitly.

The random projection has a direct impact on sparsity. For each feature contained in a document, k non-zero values are placed in the projected matrix. For ACH the situation is worse, since each feature is mapped, on average, into $n/3$ non-zero values. As an example, for $n = 5,000$ each feature will be mapped into 50 and 1,666 non-zero values in $\text{RI}_{1\%}$ and in ACH, respectively.

As an example, we have rerun our RCV1/RCV2 experiments with English and Italian as the only languages, and examined their matrix density (percentage of non-zero values over the total matrix size) and memory footprint (absolute number of non-zero values). The results are displayed in Figure 6.

LRI requires double the space with respect to standard BoW, but succeeds in preserving sparsity, while $\text{RI}_{1\%}$ drastically increases the matrix density and produces a large memory footprint. MDM, LSA, and ACH operate on dense matrices. However, since both MDM and LSA produce an extreme dimensionality reduction, the overall memory footprint remains much lower than that of $\text{RI}_{1\%}$ and, especially, of ACH. When $n = |F|$, LRI must allocate about $1,844 \cdot 10^3$ values (this is indicated as "LRI (full)" in Figure 6), while $\text{RI}_{1\%}$ ($n = 5000$) must allocate about $28,463 \cdot 10^3$ values (requiring 15.42 times more space); ACH ($n = 5000$) must allocate $55,998 \cdot 10^3$ values (30.35 times more space). Note that even though MDM and LSA reduce significantly the dimensionality (e.g., from 51,828 to 500, or 1,000), they need to allocate more values in memory than LRI (full).

As an example, let us suppose that each non-zero value is represented as a "double" (typically: 8 bytes in most modern programming languages); this means we roughly need 428MB for ACH and 218MB for $\text{RI}_{1\%}$, whereas LRI requires only 15MB. Although the difference is substantial, (even taking into account that the actual memory needed is higher

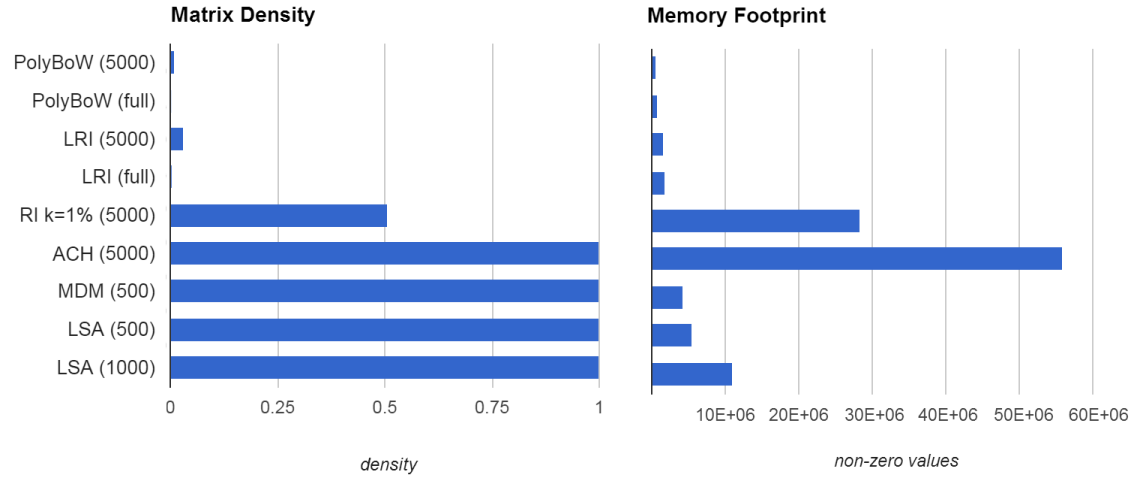


Figure 6: Matrix density (left) and memory footprint (right) in the RCV1/RCV2 English-Italian run ($11,200 \times 51,828$ full training matrix size).

if the values are indexed in a hash table) they still do not represent any real problem in terms of space for most modern computers. However, note that the matrix is not the only data structure we need to allocate in memory. Also the mapping dictionary, i.e., the data structure linking each original feature to its random index vector, should be allocated in memory. The dictionary will be queried as many times as there are terms in any document we want to classify. If the dictionary is small enough (which it is in LRI), we may be able to allocate it in cache in order to significantly speed up the indexing of new documents.

Assuming a sparse representation, a random index vector can be described as a list of k pairs (d_i, v_i), where d_i indicates a latent dimension and v_i encodes its value. For example, for $k = 2$ the random vector $(0, 0, +1, 0, -1, 0, ...)$ could be represented as $[(3, 1), (5, 0)]$, where a bit set to 1 encodes '+1' and a bit set to 0 encodes '−1'. As from Equation 5, the space occupation for the dictionary of a random indexing method depends on (i) $|F|$, the number of indexes; (ii) k, the number of non-zero values for each index; and (iii) the number of bits needed to indicate one latent position and to encode all possible non-trivial values; that is,

$$Cost(RI_k) = O(|F| \cdot k\,(\log_2 n + \log_2 2)) \tag{5}$$

It turns out that, given that the expected number of non-zero values for ACH is $n/3$, using a dense representation for each index is cheaper. Each position thus indicates one of the three possible values for the index. The cost in terms of space of the ACH index dictionary is described by

$$Cost(ACH) = O(|F| \cdot n \cdot \log_2 3) \tag{6}$$

Method	Index type	Index size	Index cell	Memory required
LRI	sparse	2	$\log_2 n + \log_2 2$ bits to encode dim_i and val_i, resp.	1.39MB
$RI_{1\%}$	sparse	100	$\log_2 n + \log_2 2$ bits to encode dim_i and val_i, resp.	69.31MB
ACH	dense	10,000	$\log_2 3$ bits to encode λ_{ij}	768.87MB

Table 3: Memory occupation for the feature dictionary for different random mapping methods on the JRC-Acquis dataset ($|F| = 406,934$). The meanings of dim_i and val_i are as in Algorithm 1. The meaning of λ_{ij} is as in Equation 1.

Assuming the reduced dimensionality is set to a fixed percentage of the original dimensionality, i.e., $n = \alpha|F|$ with $0 < \alpha \leq 1$, the following hold:

$$\begin{aligned} Cost(RI) &= O(|F|^2 \log_2 |F|) > \\ Cost(ACH) &= O(|F|^2) > \\ Cost(LRI) &= O(|F| \log_2 |F|) \end{aligned} \tag{7}$$

However, the hidden constants play a key role in practice. As an example, we have computed the total amount of memory required for each method for storing the index dictionaries for $n = 10,000$ in JRC-Acquis, where $|F| = 406,934$; the resulting values are reported in Table 3. As it can be observed, for the index dictionary ACH requires 769MB, while the space required for the RI-based versions is one to three orders of magnitude smaller. In other words, the index dictionary for LRI could easily fit in current cache memories, while $RI_{1\%}$ and ACH need to resort to higher-capacity, and thus slower, storage devices.

5.2 Time Efficiency

It is usually the case that sparsity benefits not only space occupation, but also execution time. As an example, the computational cost of SVD is $O(|F|^2|D|)$ for a document-by-term matrix; however, the implementation SVDLIBC is specifically optimized for sparse matrices and requires $O(c|F||D|)$ steps, where c is the average number of non-zero values in a vector.

In Figure 7 we plot run times for the experiments on the bilingual (English-Italian) RCV1/RCV2 experiment by paying attention to the time required for (i) obtaining the transformed index for the training set, (ii) training the learning algorithm (SVM), (iii) obtaining the transformed index for the test set, and (iv) classifying the test documents. All the experiments were run on an Intel i7 64bit processor with 12 cores, running at 1,600MHz, and 24GBs RAM memory.

The results show that it takes about 3.5 minutes to generate and test the classifier that uses the BoW representation. Time is slightly reduced to about 3 minutes when only 5000 features are selected. The total time for LRI is roughly higher by a factor of 2, up to 7.3 (full) and 6.6 ($n = 5000$) minutes, respectively. Notwithstanding this, these figures are still low when compared to the other methods: both training and testing times grow very substantially for RI and ACH. Regarding latent methods, it should be pointed out that the time required for preparing the matrices also grow substantially, due to the large

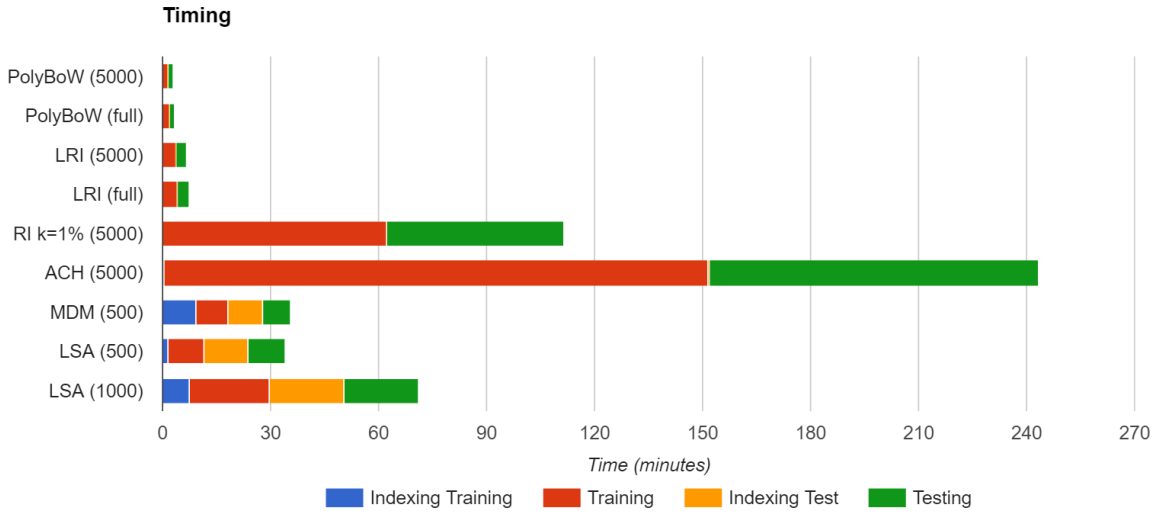


Figure 7: Run times on RCV1/RCV2 (English and Italian setting).

computational cost inherent in SVD and matrix multiplication, while in the case of random indexing methods these times are negligible.

By comparing the overall memory footprint (Figure 6, right) with execution times (Figure 7) it seems clear that there is a strong correlation between them. We have investigated this dependency in our experiments by computing the Pearson correlation between them. The *Pearson correlation* quantifies the degree of linear dependence between two variables, and ranges from -1, meaning perfect negative correlation, to $+1$, meaning perfect positive correlation, whereas 0 means that there is not any linear dependency. We found a linear Pearson correlation of $+0.988$ and $+0.998$ between the number of non-zero values in the matrix and times required for training and testing, respectively, which brings additional support to our observation: preserving sparsity during the projection favours execution times in PLTC.

5.3 The Effect of k in Random Indexing

Previous work in RI (see, e.g., Sahlgren & Karlgren, 2005; Sahlgren & Cöster, 2004) tend to set k to about 1% of the dimension of the vector; smaller values of k (about $k = 0.1\%$) have also been explored (Karlgren, Holst, & Sahlgren, 2008). Other works related to random projections (see, e.g., Achlioptas, 2001; Li et al., 2006) have noticed that sparse projection matrices help to speed up computation.

Besides run times, sparsity in the projection matrix also affects the orthogonality of the random projection, which in turn has an impact on the preservation of the relative distances. Two random vectors r_i and r_j are said to be orthogonal if the angle between them is 90 degrees. Although the probability that any two randomly picked vectors are orthogonal increases as the dimensionality of the vector space grows (Karlgren et al., 2008), most random projection approaches choose sparse random vectors, so as to maximize this probability.

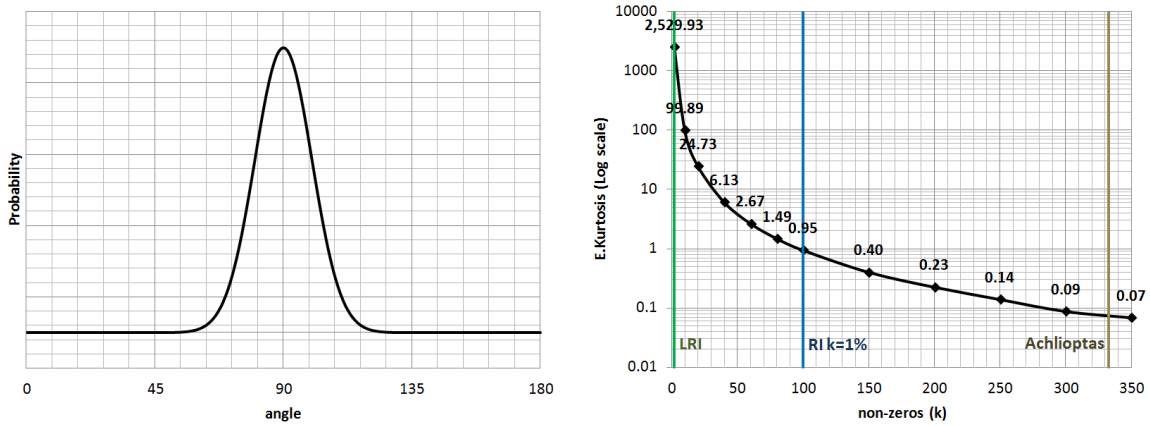


Figure 8: Probability distribution of the angle between any two arbitrary vectors in high-dimensional space (left), and excess kurtosis as a function of the non-zero values in a projection matrix of 10,000 dimensions (right).

We could thus establish a parallelism between the degree of orthogonality of any projection matrix and the probability distribution of the angle of any two of its random vectors. The more this probability distribution is skewed towards 90 degrees, the closer to orthogonal the projection base is, and the better distances are preserved. We propose to quantify the orthogonality by means of the excess kurtosis of the distribution of this angle[4]. To this aim, we have studied how the kurtosis of the angle distributions (as estimated via a Monte Carlo algorithm) varies as a function of the matrix sparsity k for any 10,000-dimension projection matrix (Figure 8, right).

Figure 8 shows that the orthogonality of the projection, for a fixed dimensionality, rapidly degrades as the density increases. LRI is thus expected to produce the most nearly orthogonal indexing, followed by RI and then by ACH.

We have further investigated the relation between orthogonality and PLTC accuracy. To this aim, we have run a series of experiments on the bilingual version of RCV1/RCV2, varying (from 2 to 100) the number k of non-zero values and (from 1,000 to 10,000) the reduced dimensionality n. Figure 9 shows the contour lines (equally valued points in the 3-dimensional representation) for classification performance (here measured in terms of F_1^{μ}), execution time, and probability of pairwise orthogonality (i.e., the probability that $\langle r_i, r_j \rangle = 0$ for any two randomly chosen random index vectors).

The following trends can be directly observed from the results: (i) accuracy improves as n increases and k decreases; (ii) run times tend to grow when both n and k increase, and (iii) the higher the dimensionality n and the smaller the parameter k, the higher the probability of finding two orthogonal random indexes.

4. The *excess kurtosis* of a random variable X is typically defined as its fourth standardized moment minus 3, i.e., $EKurt[X] = \frac{\mu_4}{\sigma^4} - 3$.

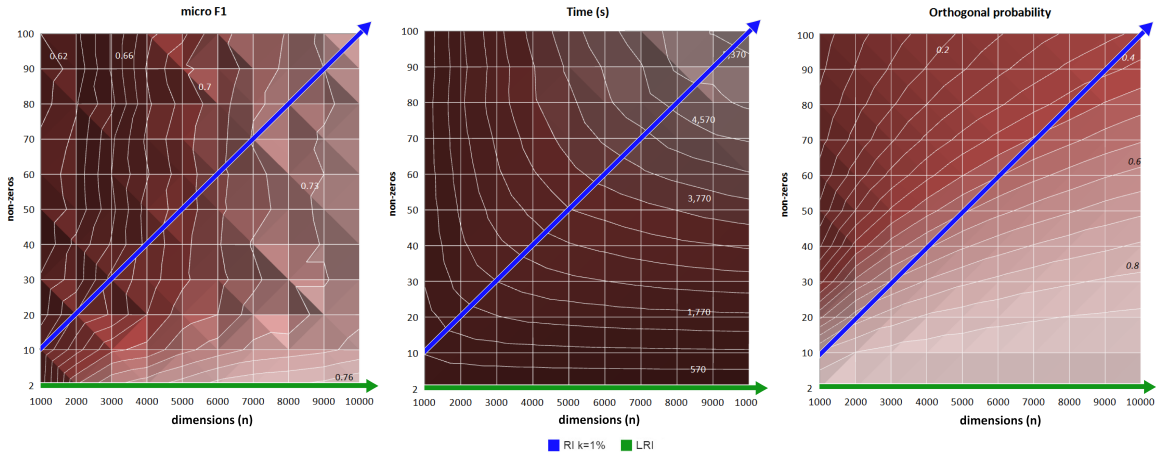


Figure 9: Impact of dimensionality n (on the x axis) and number k of non-zero values (on the y axis) on classification accuracy (left), execution time (center), and probability of finding an orthogonal pair of random indexes (right). Darker regions represent lower values.

In Figure 9, the behaviour of the LRI method we propose is described by the green horizontal line at the bottom of each plot, while RI's behaviour is described by the blue diagonal line from coordinates $(n = 1,000, k = 10)$ to $(n = 10,000, k = 100)$. The performance in RI improves at the cost of space and time efficiency, and by gradually disrupting the orthogonality of the base. On the contrary, the following desirable features of LRI are evident: when dimensionality increases (i) accuracy improves without penalizing execution times, due to the preservation of sparsity, and (ii) the orthogonality of the base is improved.

6. Conclusions

We have compared several techniques for polylingual text classification, checking their suitability as dimensionality reduction techniques and as techniques for the generation of alternative representations for the co-occurrence matrix, on two PLTC benchmarks (one parallel and one comparable). Our investigation indicates that reducing the dimensionality of the data is not sufficient if reasonable efficiency (in terms of both time and space) is required. Based on this observation we have proposed a variant of Random Indexing, a method originated within the IR community that, to the best of our knowledge, was never tested in PLTC up to date. Our proposal, Lightweight Random Indexing, yielded the best results not only in terms of (both time and space) efficiency, but also in terms of classification accuracy, for which Lightweight Random Indexing obtained the best results both in terms of macro- and micro-averaged F_1. Lightweight Random Indexing preserves matrix sparsity, which means that both memory footprint and training time are not penalized. For example, from Figures 6 and 7 we may see that Lightweight Random Indexing (in the "full" configuration – that is, where the random vectors have the same dimensionality of the original space) improved over Latent Semantic Analysis (in the $n = 1,000$ configuration – that is, where

the dimensionality of the reduced space is 1,000) by a margin of +4.37% in terms of F_1^μ with an 89.69% reduction in execution time and an 82.60% reduction in memory footprint.

Even though Lightweight Random Indexing works very well as a dimensionality reduction method, it achieves its best performance when the projection does not reduce the original dimensionality. Apparently, the BoW representation might be expected to be preferable in such a case, because it is truly orthogonal. However, in the polylingual BoW representation most of the features are only informative for a restricted set of the data; e.g., a German term has an entire dimension reserved for it in the vector space model, and this dimension is useful only for documents written in German. Random projections instead map the feature space into a space that is shared among all languages at once. The effect is that any dimension of the space becomes informative to represent documents regardless of the language they were originally written in. This configuration, in which the projection space is larger than the actual number of different features for a single language, is reminiscent of the "kernel-trick" effect, because the informative space for each language is enlarged and thus becomes more easily separable.

In the light of our experiments, Lightweight Random Indexing has important advantages with respect to previous PLTC approaches. First, the method is machine translation-free, dictionary-free, and does not require any sort of additional resources apart from the labelled collection. The projected matrix preserves sparsity, which has a direct effect in reducing both running time and total memory usage. With respect to the original random indexing technique, Lightweight Random Indexing presents the following advantages: (i) the probability of finding a pair of truly orthogonal indexes is higher; (ii) it requires less memory to allocate the index dictionary; and (iii) it avoids the need for tuning the k parameter.

LRI has proven to be very effective in PLTC, and we conjecture it could bring similar benefits in other related tasks, such as CLTC, cross-lingual information retrieval, as well as when tackling problems dealing with sparse and heterogeneous sources of data in general. As discussed above, one of the reasons why $k = 2$ is a safe configuration is that it still preserves the representation capacity. However, this might not hold under all circumstances; e.g., when processing huge streams of very dynamic data (e.g., streams of tweets), at a certain point the representation capacity might saturate if the dimensionality of the space has not been chosen carefully. In these cases, opting for configurations with $k > 2$ might mitigate the problem.

Another fact that emerges from our experiments is that dimensionality reduction is not necessarily a synonym of computational efficiency. The reason is that modern secondary storage data structures are optimized to operate on sparse data, and when the dimensionality is drastically reduced, matrix density may increase, and the net effect may be a decrease in efficiency. A true benefit is thus achieved to the extent that the trade-off between sparsity and separability is preserved; on this dimension, LRI proved extremely effective.

Although results are encouraging, further investigations are still needed to shed some light on the foundations of random projection methods. A first question is whether there is any criterion to better choose the random index vectors; given that the current criterion is random, it seems there might be room for better motivated strategies, possibly by leveraging class labels or by taking into account the document language labels. Considering that Random Indexing was originally proposed in the context of the IR community, we wonder whether the proposed approach could produce similar improvements on IR tasks such as

query expansion or bilingual lexicon acquisition. Finally, it could be interesting to combine Lightweight Random Indexing with Reflexive Random Indexing (Cohen, Schvaneveldt, & Widdows, 2010; Rangan, 2011), a more recent formulation of the model that iteratively alternates between row indexing and column indexing in the original co-occurrence matrix.

Acknowledgements

Fabrizio Sebastiani is on leave from Consiglio Nazionale delle Ricerche, Italy.

References

Achlioptas, D. (2001). Database-friendly random projections. In *Proceedings of the 20th ACM Symposium on Principles of Database Systems (PODS 2001)*, pp. 274–281, Santa Barbara, US.

Al-Rfou', R., Perozzi, B., & Skiena, S. (2013). Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013)*, pp. 183–192, Sofia, BL.

Amini, M.-R., & Goutte, C. (2010). A co-classification approach to learning from multilingual corpora. *Machine Learning*, *79*(1/2), 105–121.

Amini, M.-R., Usunier, N., & Goutte, C. (2009). Learning from multiple partially observed views; An application to multilingual text categorization. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS 2009)*, pp. 28–36, Vancouver, CA.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pp. 238–247, Baltomore, US.

Bel, N., Koster, C. H., & Villegas, M. (2003). Cross-lingual text categorization. In *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003)*, pp. 126–139, Trondheim, NO.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1–127.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). Neural probabilistic language models. In *Innovations in Machine Learning*, pp. 137–186. Springer, Heidelberg, DE.

Bingham, E., & Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2001)*, pp. 245–250, San Francisco, US.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bullinaria, J. A., & Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, *39*(3), 510–526.

Chandar, S., Lauly, S., Larochelle, H., Khapra, M. M., Ravindran, B., Raykar, V. C., & Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS 2014)*, pp. 1853–1861, Montreal, CA.

Cohen, T., Schvaneveldt, R., & Widdows, D. (2010). Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, *43*(2), 240–256.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

de Melo, G., & Siersdorfer, S. (2007). Multilingual text classification using ontologies. In *Proceedings of the 29th European Conference on Information Retrieval (ECIR 2007)*, pp. 541–548, Roma, IT.

Deerwester, S. C., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Dumais, S. T., Letsche, T. A., Littman, M. L., & Landauer, T. K. (1997). Automatic cross-language retrieval using latent semantic indexing. In *Working Notes of the AAAI Spring Symposium on Cross-language Text and Speech Retrieval*, pp. 18–24, Stanford, US.

Ehrmann, M., Cecconi, F., Vannella, D., McCrae, J. P., Cimiano, P., & Navigli, R. (2014). Representing multilingual data as linked data: The case of BabelNet 2.0. In *Proceedings of the 9th Conference on Language Resources and Evaluation (LREC 2014)*, pp. 401–408, Reykjavik, IS.

Esuli, A., Fagni, T., & Moreo, A. (2016). JaTeCS (Java Text Categorization System). In *Github*. Retrieved September 11, 2016, from `https://github.com/jatecs/jatecs`.

Faruqui, M., & Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pp. 462–471, Gothenburg, SE.

Forman, G. (2004). A pitfall and solution in multi-class feature selection for text classification. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, pp. 38–45, Banff, CA.

Fradkin, D., & Madigan, D. (2003). Experiments with random projections for machine learning. In *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2003)*, pp. 517–522, Washington, US.

García Adeva, J. J., Calvo, R. A., & López de Ipiña, D. (2005). Multilingual approaches to text categorisation. *European Journal for the Informatics Professional*, *6*(3), 43–51.

Gliozzo, A., & Strapparava, C. (2005). Cross-language text categorization by acquiring multilingual domain models from comparable corpora. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pp. 9–16, Ann Arbor, US.

Gliozzo, A., & Strapparava, C. (2006). Exploiting comparable corpora and bilingual dictionaries for cross-language text categorization. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, pp. 553–560, Sydney, AU.

Gorman, J., & Curran, J. R. (2006). Random indexing using statistical weight functions. In *Proceedings of the 4th Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pp. 457–464, Sydney, AU.

Gouws, S., & Søgaard, A. (2015). Simple task-specific bilingual word embeddings. In *Proceedings of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies Conference (NAACL-HLT 2015)*, pp. 1386–1390.

Haddow, B., Hoang, H., Bertoldi, N., Bojar, O., & Heafield, K. (2016). MOSES statistical machine translation system. In *Moses website*. Retrieved September 11, 2016, from `http://www.statmt.org/moses/`.

Harris, Z. S. (1968). *Mathematical structures of language*. Wiley, New York, US.

Hecht-Nielsen, R. (1994). Context vectors: General-purpose approximate meaning representations self-organized from raw data. In *Computational Intelligence: Imitating Life*, pp. 43–56. IEEE Press.

Hermann, K. M., & Blunsom, P. (2014). Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pp. 58–68, Baltimore, US.

Joachims, T. (2009). SVMperf: Support Vector Machine for multivariate performance measures. In *Cornell University website*. Retrieved September 11, 2016, from `http://www.cs.cornell.edu/people/tj/svm_light/svm_perf.html`.

Joachims, T. (2005). A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pp. 377–384, Bonn, DE.

Johnson, W. B., Lindenstrauss, J., & Schechtman, G. (1986). Extensions of Lipschitz maps into Banach spaces. *Israel Journal of Mathematics*, *54*(2), 129–138.

Jurgens, D., & Stevens, K. (2009). Event detection in blogs using temporal random indexing. In *Proceedings of the Workshop on Events in Emerging Text Types*, pp. 9–16, Borovets, BG.

Kanerva, P., Kristofersson, J., & Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society (CogSci 2000)*, p. 1036, Philadelphia, US.

Karlgren, J., Holst, A., & Sahlgren, M. (2008). Filaments of meaning in word space. In *Proceedings of the 30th European Conference on Information Retrieval (ECIR 2008)*, pp. 531–538, Glasgow, UK.

Kaski, S. (1998). Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 1998)*, pp. 413–418, Anchorage, US.

Klementiev, A., Titov, I., & Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pp. 1459–1474, Mumbai, IN.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, Vol. 5, pp. 79–86. Publicly available in `http://www.statmt.org/europarl/`.

Lauly, S., Boulanger, A., & Larochelle, H. (2014). Learning Multilingual Word Representations using a Bag-of-Words Autoencoder. *ArXiv e-prints*, *arXiv:1401.1803 [cs.CL]*.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, *5*(Apr), 361–397. Publicly available in `http://www.jmlr.org/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm`.

Li, P., Hastie, T. J., & Church, K. W. (2006). Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pp. 287–296, Philadelphia, US.

Li, Y., & Shawe-Taylor, J. (2007). Advanced learning algorithms for cross-language patent retrieval and classification. *Information Processing and Management*, *43*(5), 1183–1199.

Mikolov, T., Le, Q. V., & Sutskever, I. (2013a). Exploiting Similarities among Languages for Machine Translation. *ArXiv e-prints*, *arXiv:1309.4168 [cs.CL]*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS 2013)*, pp. 3111–3119, Lake Tahoe, US.

Mimno, D., Wallach, H. M., Naradowsky, J., Smith, D. A., & McCallum, A. (2009). Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pp. 880–889, Singapore, SN.

Moreo, A. (2016). Data resources for reproducing experiments in polylingual text classification. In *Human Language Technologies (HLT) group website*. Retrieved September 11, 2016, from `http://hlt.isti.cnr.it/pltc`.

Nastase, V., & Strapparava, C. (2013). Bridging languages through etymology: The case of cross-language text categorization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 651–659, Sofia, BL.

Österlund, A., Ödling, D., & Sahlgren, M. (2015). Factorization of latent variables in distributional semantic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 227–231, Lisbon, PT.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Papadimitriou, C. H., Raghavan, P., Tamaki, H., & Vempala, S. (1998). Latent semantic indexing: A probabilistic analysis. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems (PODS 1998)*, pp. 159–168, Seattle, US.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP 2014)*, pp. 1532–1543, Doha, QA.

Prettenhofer, P., & Stein, B. (2010). Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pp. 1118–1127, Uppsala, SE.

Rangan, V. (2011). Discovery of related terms in a corpus using reflective random indexing. In *Proceedings of the ICAIL 2011 Workshop on Setting Standards for Searching Electronically Stored Information*, Pittsburgh, US.

Richardson, J. (2008). PolyLDA++. In *Atlassian Bitbucket*. Retrieved September 11, 2016, from `https://bitbucket.org/trickytoforget/polylda`.

Rigutini, L., Maggini, M., & Liu, B. (2005). An EM-based training algorithm for cross-language text categorization. In *Proceedings of the 3rd IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005)*, pp. 529–535, Compiègne, FR.

Rohde, D. (2011). A C library for computing singular value decompositions. In *SVDLIBC*. Retrieved September 11, 2016, from `http://tedlab.mit.edu/~dr/SVDLIBC/`.

Sahlgren, M. (2001). Vector-based semantic analysis: Representing word meanings based on random labels. In *Proceedings of the ESSLLI Workshop on Semantic Knowledge Acquistion and Categorization*, Helsinki, FI.

Sahlgren, M. (2005). An introduction to random indexing. In *Proceedings of the Workshop on Methods and Applications of Semantic Indexing*, Copenhagen, DK.

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Swedish Institute for Computer Science, University of Stockholm, Stockholm, SE.

Sahlgren, M., & Cöster, R. (2004). Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, CH.

Sahlgren, M., & Karlgren, J. (2005). Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural Language Engineering*, *11*(3), 327–341.

Sahlgren, M., Karlgren, J., Cöster, R., & Järvinen, T. (2002). SICS at CLEF 2002: Automatic query expansion using random indexing. In *Working Notes of the Cross-Language Evaluation Forum Workshop (CLEF 2002)*, pp. 311–320, Roma, IT.

Steinberger, R., Pouliquen, B., & Ignat, C. (2004). Exploiting multilingual nomenclatures and language-independent text features as an interlingua for cross-lingual text analysis applications. In *Proceedings of the 4th Slovenian Language Technology Conference*, Ljubljana, SL.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., & Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pp. 2142–2147, Genova, IT. Publicly available in `https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis`.

Vinokourov, A., Shawe-Taylor, J., & Cristianini, N. (2002). Inferring a semantic representation of text via cross-language correlation analysis. In *Proceedings of the 16th Annual Conference on Neural Information Processing Systems (NIPS 2002)*, pp. 1473–1480, Vancouver, CA.

Vulić, I., & Moens, M.-F. (2015). Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2015)*, pp. 363–372, Santiago, CL.

Wei, C.-P., Lin, Y.-T., & Yang, C. C. (2011). Cross-lingual text categorization: Conquering language boundaries in globalized environments. *Information Processing and Management*, *47*(5), 786–804.

Wei, C.-P., Yang, C.-S., Lee, C.-H., Shi, H., & Yang, C. C. (2014). Exploiting poly-lingual documents for improving text categorization effectiveness. *Decision Support Systems*, *57*, 64–76.

Xiao, M., & Guo, Y. (2013). A novel two-step method for cross-language representation learning. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS 2013)*, pp. 1259–1267, Lake Tahoe, US.

Xu, C., Tao, D., & Xu, C. (2013). A survey on multi-view learning. *ArXiv e-prints*, *arXiv:1304.5634 [cs.LG]*.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML 1997)*, pp. 412–420, Nashville, US.

Zou, W. Y., Socher, R., Cer, D. M., & Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP 2013)*, pp. 1393–1398, Melbourne, AU.